

# Methoden in der Formoptimierung

Andreas Schmitt 296194

16.04.2014

## 1 Einleitung

Bei einer Optimierung wird versucht ein Modell bestmöglich bezüglich einer bestimmten Eigenschaft auszulegen. Diese Eigenschaft ist durch einen funktionalen Zusammenhang von dem Modell abhängig. In der Formoptimierung ist die Eigenschaft von der Geometrie des betrachteten Bauteiles abhängig. Die Optimierung wird mit Optimierungsalgorithmen durchgeführt, die auf verschiedene Arten das Modell anhand von Parametern, die auch Designvariablen genannt werden, verändern, um den Wert der Eigenschaft zu minimieren, oder maximieren. Diese Parameter sind im Fall der Formoptimierung Punkte im Raum und deren Verhältnis zueinander, die die Geometrie des zu optimierenden Bauteiles beschreiben.

In dieser Ausarbeitung werden Methoden zur Parametrisierung von Geometrien und zur Optimierung einer parametrisierten Geometrie vorgestellt, die in der numerischen Formoptimierung genutzt werden. Hierbei wird ein besonderer Fokus auf genetische Optimierungsalgorithmen gelegt.

Formoptimierung ist ein Teil der Strukturoptimierung. Die Strukturoptimierung ist der Prozess zur optimalen Auslegung technischer Bauteile und Konstruktionen bezüglich ihrer Anwendung. Anhand der Abbildung 1 ist gut zu erkennen, welchen Teil die Formoptimierung im Prozess der Strukturoptimierung einnimmt. Das anschauliche Beispiel des Tragwerkes zeigt, dass für eine Optimierung Querschnitts-, Form- und Topologieoptimierung benötigt werden. Diese unterscheiden sich durch die zur Optimierung genutzten Parameter. Bei der Querschnittsoptimierung liegt der Fokus auf Parametern wie der Dicke, Höhe, Querschnittsfläche etc., die den Querschnitt der einzelnen Bauteile in der gesamten Konstruktion beschreiben. Die Formoptimierung nutzt vor allem geometrische Größen der gesamten Konstruktion als Parameter. Die Topologieoptimierung wählt die beste grundsätzliche Modellierung der Struktur aus verschiedenen Varianten aus.

Ein weiteres Beispiel, welches im Folgenden immer wieder angeführt werden wird, ist die Optimierung eines Tragflügelprofils (Abb. 2). Das Profil soll so optimiert werden, dass es zum Beispiel einen maximalen Auftrieb gewährleistet.

Um eine solche Optimierung durchführen zu können, muss zunächst das Profil parametrisiert werden, um dem Optimierer einen Parametersatz zur Verfügung stellen zu können, der im Optimierungsprozess variiert werden kann. Verschiedene Methoden der Parametrisierung werden im folgenden Kapitel behandelt. Anschließend wird auf Optimierungsalgorithmen eingegangen, die im Zuge der Formoptimierung genutzt werden können.

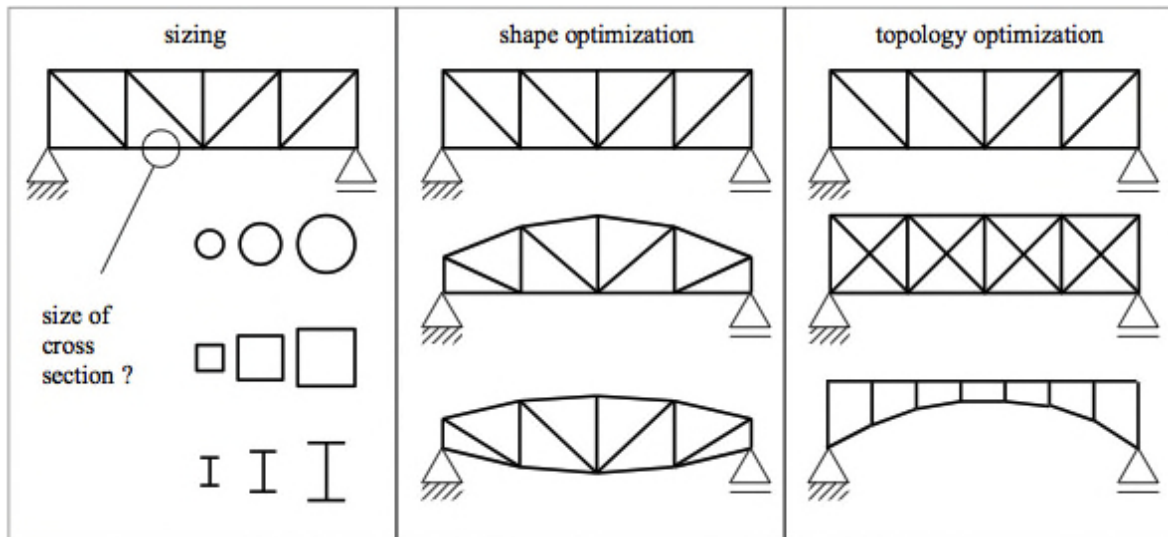


Abbildung 1: Einteilung der Strukturoptimierung in die drei Optimierungsbereiche am Beispiel eines Tragwerkes. Grafik entnommen aus [2]

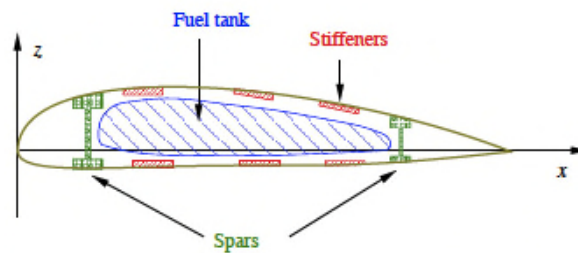


Abbildung 2: Geometrie eines Tragflügels entnommen aus [5]

## 2 Parametrisierung

Bei einer numerische Optimierung muss die Form in geeigneter Weise mathematisch dargestellt werden. Die Parameter dieser Darstellung sind gleichzeitig die Stellschrauben für den Optimierungsprozess. Für diese Parametrisierung gibt es verschiedene Verfahren, die im Folgenden auf Basis der Ausarbeitung von Samareh [5] diskutiert werden.

## 2.1 Diskrete Koordinatendarstellung

Die einfachste Form der Parametrisierung ist die diskrete Koordinatendarstellung. Hier bilden die Koordinaten diskreter Punkte der Oberfläche den Parametersatz, wie es in Abbildung 3 dargestellt ist. Diese Parameter stimmen im Falle eines schon vorhandenen Gitters mit den Gitterpunkten des Profils überein. Die Verbindungen zwischen den einzelnen Punkten sind durch eine lineare Interpolation gegeben.

Der Vorteil dieser Methode ist, dass ein schon vorhandenes, bzw. für die Simulation zu erstellendes Gitter für die Parametrisierung genutzt werden kann. Hierdurch entsteht kein weiterer Rechenaufwand für die Parametrisierung der Geometrie.

Diese Methode wird jedoch nur sehr selten genutzt, da sie gravierende Nachteile besitzt. Diese sind zum einen, dass sich mit der diskreten Koordinatendarstellung eine sehr große Anzahl an Parametern ergibt, die zu sehr langen Rechenzeiten in der Optimierung führen. Zum anderen gibt es bei dieser Methode keine einschränkenden Bedingungen zwischen den einzelnen Punkten, so dass diese durch den Optimierer frei verschoben werden können. Dies kann zum Verlust der glatten Geometrie und auch zu unphysikalischen Formen führen. Dieses Problem kann zwar durch das Einführen eben solcher Bedingungen eingeschränkt werden, allerdings bleibt die lange Rechenzeit auch dann ein großer Nachteil der Methode.

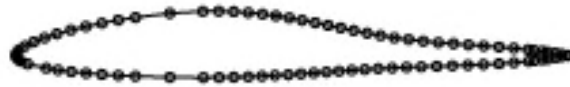


Abbildung 3: Parametrisierung einer Flügelgeometrie mit der diskreten Koordinatendarstellung. Entnommen aus [5]

## 2.2 Parametrisierung mit Polynomen und Splines

Um der großen Anzahl an Parametern entgegen zu wirken, werden Polynom- und Splinedarstellungen der Geometrie  $G$  genutzt. In Abbildung 4 ist sehr gut zu erkennen, dass die Geometrie des Flügelprofils mit in diesem Beispiel neun Parametern dargestellt werden kann. Der neunte Punkt liegt auf dem ersten um das Profil zu schließen und eine nicht gerundete Ablaufkante zu erzeugen.

Einfache Kurven können sehr gut mit Hilfe von Polynomen in der Basisform

$$G(u) = \sum_{i=0}^{n-1} c_i u^i \quad (1)$$

dargestellt werden. Hierbei ist  $n$  die Anzahl der Designvariablen,  $c_i$  sind eine Menge von Koeffizientenvektoren, deren Komponenten ebenfalls als Designvariablen genutzt werden können. Die Variable  $u$  ist die Parameterkoordinate entlang der beschriebenen Kurve.



Abbildung 4: Parametrisierung eines Flügelprofils mit neun Kontrollpunkten entnommen aus [5]

Eine weitere Möglichkeit der Parametrisierung auf der Basis von Polynomen sind Bézierkurven, welche mit der Vorschrift

$$G(u) = \sum_{i=1}^n P_i B_{i,p}(u) \quad (2)$$

aufgestellt werden. Hierbei handelt es sich um die Summe von Produkten aus einem Bernsteinpolynom  $B_{i,p}(u)$  der  $p$ -ten Ordnung und Kontrollpunktkoordinaten  $P_i$ . Letztere sind die Parameter dieser Darstellung. Die beiden genannten Darstellungen in Glg. (1) und (2) sind mathematisch äquivalent. Allerdings ist der Zusammenhang zwischen den Kontrollpunkten und der sich ergebenden Form bei den Bézierkurven deutlich anschaulicher. Die beschriebene Kurve liegt immer in der konvexen Hülle der Kontrollpunkte. Ein numerischer Vorteil ist, dass die Bernsteinpolynome rekursiv berechnet werden und somit weniger Rundungsfehler auftreten als bei der Darstellung mit Basispolynomen. Bei steigender Komplexität der beschriebenen Kurve steigen jedoch sowohl der Rundungsfehler, als auch die Berechnungskosten für die Bézierkurven, da Bernsteinpolynome höherer Ordnung benötigt werden.

Komplexe Kurven werden aus dem genannten Grund nicht mit einer Bézierkurve beschrieben. Hierfür werden zum Beispiel B-Splines genutzt. Diese sind Funktionen, die stückweise aus Polynomen höchstens  $p$ -ten Grades zusammengesetzt sind und komplexe Kurve beschreiben. Ein solches B-Spline kann mit

$$G(u) = \sum_{i=1}^n P_i N_{i,p}(u) \quad (3)$$

beschrieben werden, wobei  $P_i$  die B-Spline Kontrollpunkte sind und  $N_{i,p}$  die  $i$ -te B-Spline Basisfunktion der Ordnung  $p$  ist. Mit diesen Splines ist es möglich komplexe Kurven effizient und genau darzustellen. Bei impliziten konischen Geometrien treten jedoch Probleme auf.

Zur Beschreibung solcher Formen gibt es die sogenannten Non-Uniform Rational B-Splines (NURBS). Mit diesen NURBS können auch Formen wie Zylinder oder Kegel dargestellt werden. Eine Ausnahme sind weiterhin einige wenige implizite Formen. Die NURBS Kurve wird durch

$$G(u) = \frac{\sum_{i=1}^n N_{i,p}(u) W_i P_i}{\sum_{i=1}^n N_{i,p}(u) W_i}$$

beschrieben. Hier wurden im Vergleich zu den B-Splines in Glg. (3) noch Gewichte  $W_i$  eingeführt, die ebenfalls als Designvariablen genutzt werden können.

Alle in diesem Abschnitt genannten Methoden sind sehr gut auf zweidimensionale und einfache dreidimensionale Problemstellungen anwendbar. Bei der Parametrisierung komplexer dreidimensionaler Formen werden jedoch sehr viele Kontrollpunkte benötigt und bei ihrer Optimierung können irreguläre und wellige Geometrien entstehen.

## 2.3 Free-Form Deformation

Ein weiterer viel genutzter Ansatz ist die so genannte Free-Form Deformation. Im Unterschied zu den bisher erwähnten Verfahren wird dabei während der Optimierung nicht die Form selbst, sondern das zur Beschreibung genutzte Gitter verformt. Damit sind die Gitterpunkte die Parameter des Optimierungsvorganges.

Der erste Schritt der Free-Form Deformation ist die Parametrisierung der Geometrie, indem die Geometrie mit einem einfachen Objekt, wie einem Rechteck im zweidimensionalen Fall, eingehüllt wird. Dieses Rechteck wird durch ein Spline dargestellt, dessen Basispunkte ein einfaches Gitter sind. In dieser Hülle befindet sich die Geometrie als Punktwolke. Dies ist in Abbildung 5(a) anhand eines Flügelprofils dargestellt.

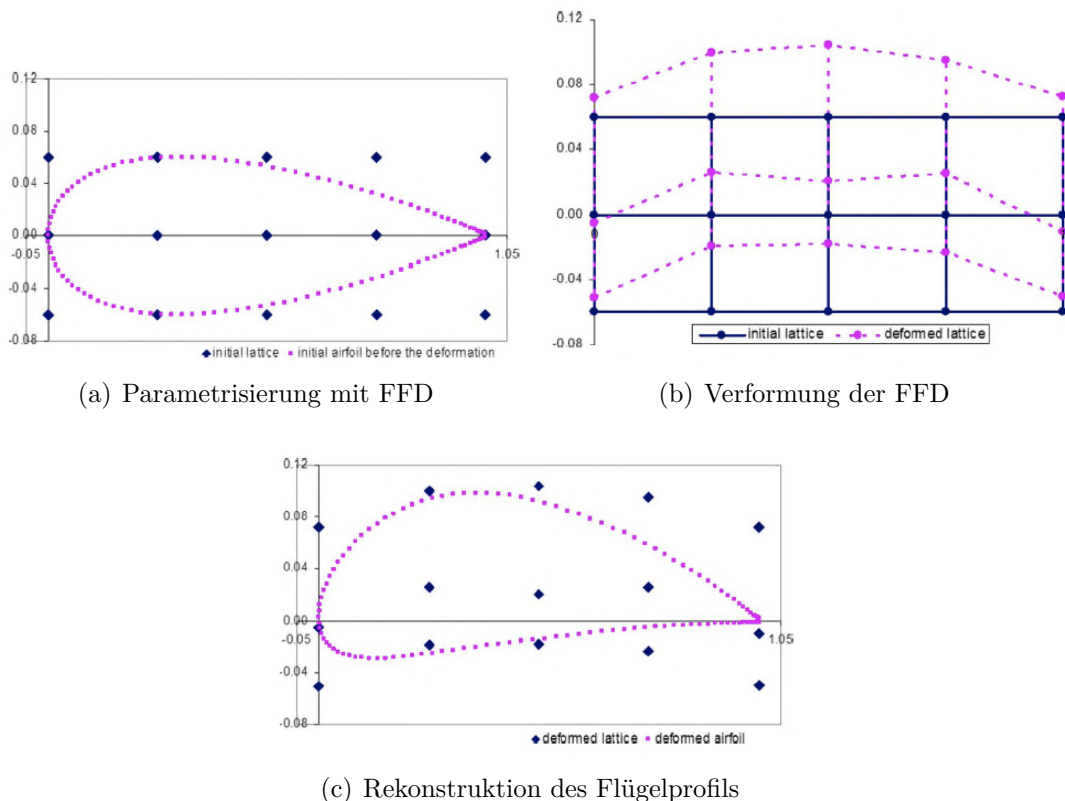


Abbildung 5: Schritte der Free-Form Deformation (FFD). Bilder entnommen aus [1]

Während der Optimierung im zweiten Schritt der Free-Form Deformation (Abb. 5(b)) werden die Kontrollpunkte des Splines, welches die Hülle beschreibt, verschoben. Der letzte Schritt der Free-Form Deformation ist das Berechnen der Position der Punkte aus der Punktwolke (Abb. 5(c)). Dies ist einfach möglich, da die Punkte im System der verformten Hülle weiterhin die selben Koordinaten wie vor der Verformung besitzen. Diese können auf Grund der bekannten Deformation auf die entsprechenden kartesischen Koordinaten zurückgerechnet werden.

Ein Vorteil dieser Methode liegt darin, dass in dem verformten Gitter verschiedene Kurven gleichmäßig mitverformt werden. Somit kann ein Flügel, wie er in Abbildung 2 dargestellt ist auch multidisziplinär optimiert werden, da auch die Elemente im Flügel mitbetrachtet werden können. So ließe sich zum Beispiel mit dem Auftrieb gleichzeitig ein Tankinhalt optimieren. Von Nachteil ist allerdings, dass die Parameter der Optimierung keinen direkten physikalischen Zusammenhang mit dem betrachteten Objekt haben.

### 3 Optimierungsproblem

Im Folgenden wird mit besonderem Fokus auf die genetischen Algorithmen auf Optimierungsmethoden eingegangen, die in der Formoptimierung genutzt werden können, nachdem das Objekt parametrisiert ist. Für die Optimierung wird ein mathematisches Modell benötigt, welches aus einer Kostenfunktion  $f(x)$  und Nebenbedingungen  $n_i(x)$  besteht. Hierbei sind in der Variable  $x$  alle Parameter der Optimierung zusammengefasst. Das Optimierungsproblem kann dann mit

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. } & n_i(x) = 0 \quad \forall i \in E \\ & n_i(x) \geq 0 \quad \forall i \in I \end{aligned}$$

beschrieben werden mit den Indexmengen der Gleichungs-  $E$  und Ungleichungsbedingungen  $I$ . Wenn, wie zum Beispiel beim Auftrieb, die Kostenfunktion maximiert werden soll, dann wird das Problem

$$\begin{aligned} & - \min_{x \in \mathbb{R}^n} (-f(x)) \\ \text{s.t. } & n_i(x) = 0 \quad \forall i \in E \\ & n_i(x) \geq 0 \quad \forall i \in I \end{aligned}$$

betrachtet, in welchem die Maximierung in eine Minimierung überführt wurde. Für diese Problemstellungen gibt es verschiedene Optimierungsalgorithmen, mit denen sich der folgende Abschnitt befasst.

## 3.1 Optimierungsalgorithmen

Die Optimierung der oben genannten Probleme wird im Allgemeinen numerisch mit Hilfe von Optimierungsalgorithmen durchgeführt. Diese Algorithmen haben alle gemeinsam, dass sie iterative Verfahren sind. Somit benötigen sie einen Startwert  $x_0$  und ein Abbruchkriterium. Die verschiedenen Algorithmen unterscheiden sich im Wesentlichen durch die verschiedenen Arten der Iteration. Eine grobe Aufteilung in gradientenbasierte und stochastische Algorithmen ist möglich. Ein Einblick in beide Arten wird im Folgenden gegeben.

### 3.1.1 Gradientenbasierte Algorithmen

Die bekanntere Art der Optimierungsalgorithmen ist die der gradientenbasierten Algorithmen. Bei diesen Algorithmen wird als Startwert ein Punkt  $x_0$  im Parameterraum gewählt. Die Wahl ist beliebig, sie wird aber meist anhand von Erfahrungswerten mit dem betrachteten physikalischen Problem getroffen. Ausgehend von diesem Startpunkt folgt im Fall eines Line-Search basierten Algorithmus ein Iterationsschritt mit der Vorschrift

$$x_{n+1} = x_n + \alpha_n D_n \nabla f(x_n),$$

wobei  $\alpha_n$  eine optimale Schrittweite vorgibt und  $D_n$  je nach Algorithmus bestimmt wird um die Suchrichtung mit Hilfe des Gradienten  $\nabla f(x_n)$  festzulegen. Die optimale Schrittweite wird mit dem Optimierungsproblem

$$\min_{\alpha > 0} f(x_n + \alpha p_n)$$

bestimmt. Hierbei ist  $p_n$  die normierte Suchrichtung

$$p_n = D_n \frac{\nabla f(x_n)}{\|\nabla f(x_n)\|}.$$

Diesem Schema folgend wird anschließend bis zur Konvergenz zu einem Minimum iteriert. Im allgemeinen Fall findet ein solcher Algorithmus nur ein lokales Minimum, welches stark von dem gewählten Startwert abhängt. Diese und weitere Algorithmen, die auf anderen Verfahren zur Schrittweitenoptimierung basieren, finden sich in Nocedal und Wright [3].

Algorithmen dieser Art sind zum Beispiel die Steepest-Descent-, Newtons- und Conjugate-Gradients-Algorithmen. Da der Fokus dieser Ausarbeitung auf den stochastischen Algorithmen liegt, wird im Folgenden nur kurz auf die Beschreibung des Steepest-Descent-Algorithmus eingegangen, der der einfachste gradientenbasierte Algorithmus ist.

Der Steepest-Descent-Algorithmus sucht in einem Iterationsschritt den nächsten Punkt, wie der Name sagt, in Richtung des steilsten Abstieges, also entlang der negativen Richtungsableitung mit größtem Absolutwert. Diese Richtungsableitung ist in

Richtung  $-\nabla f(x_n)$ . Damit wird in der oben genannten Iterationsvorschrift die Matrix auf  $D_n = -I$  gesetzt, wobei  $I$  die Identitätsmatrix ist.

Auf Grund der gewählten Suchrichtung iteriert der Steepest-Descent-Algorithmus abhängig vom Startwert  $x_0$  in Richtung des nächstgelegenen Minimums.

### 3.1.2 Stochastische Algorithmen

Neben der genannten Gruppe der gradientenbasierten Algorithmen gibt es auch die stochastischen Algorithmen. Bei diesen Algorithmen wird der Parametersatz des nächsten Iterationsschrittes durch einen Schritt in eine zufällige Richtung mit einer ebenfalls zufälligen Länge erreicht. Die Konvergenz dieser Algorithmen wird je nach Art des Algorithmus durch verschiedene Verfahren erreicht. Der offensichtliche Vorteil dieser Algorithmen ist, dass die Kostenfunktion nicht differenzierbar sein muss, da diese Algorithmen nicht auf Gradienten zurückgreifen. Des Weiteren ist es möglich einen sehr großen Teil des Parameterraumes abzusuchen. Es wird als Startwert nicht nur ein einzelner Punkt genutzt, von dem aus ein lokales Minimum gefunden wird. Bei diesen Algorithmen wird als Startwert ein ganzer Satz von Startpunkten gewählt. Diese sind in den meisten Fällen zufällig über den Parameterraum verteilt.

Auch bei den stochastischen Algorithmen existieren verschiedene Verfahren, wobei hier auf die genetischen Algorithmen eingegangen werden soll. Bei diesen Algorithmen wird versucht das Optimierungsproblem auf Basis der natürlichen Anpassung durch Mutation einer Spezies zu lösen. Aus diesem Grund wird ein Punkt im Parameterraum Individuum und die Menge der Individuen jedes Iterationsschrittes Generation genannt. Die Methodik der genetischen Algorithmen basiert nach Paszkowicz [4] zunächst auf dem Erstellen der Startgeneration durch zufällig verteilte Individuen. Die Nachfolgenerationen der Startgeneration werden in den folgenden Iterationsschritten durch Mutation von vorhandenen Individuen erreicht, die durch eine zufällige Variation eines oder mehrerer Parameter der Individuen simuliert wird. Diese mutierten Individuen werden anschließend auf Basis einer Kostenfunktion bewertet, so dass nur die Individuen mit einem verbesserten Wert der Kostenfunktion in die neue Generation übernommen werden. Die Iteration endet sobald der Wert der Kostenfunktion an den Individuen konvergiert, oder nachdem eine vorher festgelegte Anzahl von Iterationsschritten durchlaufen wurde, da eine Konvergenz je nach Problemstellung sehr lange dauern kann. Durch dieses Abbruchkriterium nach Iterationsschritten ist es nicht in jedem Fall möglich mit diesem Algorithmus ein Optimum zu finden. In dem Fall, dass ein Optimum gefunden wird ist es jedoch sehr wahrscheinlich, dass es sich um das globale Optimum handelt. Dies liegt an der großflächigen Verteilung der Startindividuen im Parameterraum, die durch die Suche in alle Richtungen in den folgenden Iterationen den Parameterraum vollständig durchlaufen kann.

Die Art der Iteration, dass in jedem Schritt eine ganze Generation neuer Individuen berechnet wird, macht die Genetischen Algorithmen sehr leicht parallelisierbar und führt so zu deutlich kürzeren Rechenzeiten bei Problemen mit vielen Parametern.



Ein genetischer Algorithmus, an dem man das genannte Vorgehen gut zeigen kann, ist der Differential-Evolution-Algorithmus, dessen Beschreibung sich im Folgenden an die Ausarbeitung von Amoiralis und Nikolos [1] anlehnt. Dieser ist einfach zu implementieren und erzielt bei Problemen, die durch einen Genetischen Algorithmus gelöst werden können, eine gute Konvergenz. Dieser Algorithmus kann auf Optimierungsprobleme ohne Nebenbedingungen angewendet werden. Somit wird das Problem

$$\min_{X \in \mathbb{R}^{n_v}} f(X)$$

betrachtet, in dem der Parametervektor  $X$  aus  $n_v$  Parametern besteht und  $f(X)$  die zu minimierende Kostenfunktion darstellt.

Bei diesem Algorithmus ist die Anzahl der Individuen pro Generation (Iterationsschritt) auf exakt  $N_p$  festgelegt. Es werden bis zu  $m_{gens}$  Generationen berechnet, wobei  $m_{gens}$  zu Beginn der Rechnung als Abbruchkriterium festgelegt wird. Somit lässt sich die Population der Individuen durch

$$X_{i,G_N}, \quad i = 1, \dots, N_p; \quad G_N = 1, \dots, m_{gens}$$

darstellen.

Der Iterationsprozess von Generation  $G_N$  zu Generation  $G_{N+1}$  vollzieht sich in drei Schritten. Der erste Schritt ist die Mutation zum Errechnen  $N_p$  neuer temporärer Individuen auf Basis von drei zufällig ausgewählten Individuen der aktuellen Generation. Dieser Schritt folgt der Vorschrift

$$Y_{i,G_{N+1}} = X_{r_3,G_N} + F(X_{r_1,G_N} - X_{r_2,G_N})$$

mit den zufällig gewählten Indizes

$$r_1, r_2, r_3 \in \{1, \dots, N_p\}, \quad r_1 \neq r_2 \neq r_3 \neq i,$$

wobei  $F \in [0, 1]$  ein Parameter des Algorithmus ist. Anschließend wird eine Zwischenpopulation  $Z$  erstellt, indem bei jedem Individuum per Zufall für jeden Parameter bestimmt wird, ob er aus dem ursprünglichen (Population  $X$ ), oder mutierten (Population  $Y$ ) Individuum übernommen wird. Die Zusammenstellung der Zwischenpopulation

$$z_{j,i,G_{N+1}} = \begin{cases} y_{j,i,G_{N+1}} & \text{wenn } rand_j \leq C_r \vee j = k_C \\ x_{j,i,G_N} & \text{sonst} \end{cases}$$

folgt der Vektorindizierung

$$\begin{aligned} X_{i,G_N} &= (x_{1,i,G_N}, \dots, x_{n_v,i,G_N})^T \\ Y_{i,G_{N+1}} &= (y_{1,i,G_{N+1}}, \dots, y_{n_v,i,G_{N+1}})^T \\ Z_{i,G_{N+1}} &= (z_{1,i,G_{N+1}}, \dots, z_{n_v,i,G_{N+1}})^T. \end{aligned}$$

Hierbei ist  $C_r \in [0, 1]$  genauso ein Parameter des Algorithmus, wie es  $F$  ist. Diese beiden Parameter werden vor der Anwendung des Algorithmus festgelegt und sind entscheidend für die Konvergenz und Stabilität der Iteration. Die zufällige Zahl  $rand_j \in [0, 1]$  wird für jedes  $z_{j,i,G_{N+1}}$ , genauso wie der zufällige Index  $k_C \in [1, n_v]$ , neu generiert.

Im letzten Schritt werden die Individuen anhand der Kostenfunktion bewertet, um die Individuen der neuen Generation  $X_{i,G_{N+1}}$  zu bestimmen. Hier wird nach folgendem Kriterium

$$X_{i,G_{N+1}} = \begin{cases} Z_{i,G_{N+1}} & \text{wenn } f(Z_{i,G_{N+1}}) \leq f(X_{i,G_N}) \\ X_{i,G_N} & \text{sonst} \end{cases}$$

ausgewählt, ob das Individuum aus der alten Generation übernommen wird, oder ob stattdessen das Individuum der Zwischengeneration aufgenommen wird.

Diese drei Schritte werden wiederholt, bis entweder eine Konvergenz der Individuen einer Generation, oder die maximale Anzahl an Generationen erreicht wird.

## 4 Zusammenfassung

In dieser Ausarbeitung wurde ein kleiner Überblick über verschiedene Methoden in der Formoptimierung gegeben. Bei den Methoden der Parametrisierung wurde zunächst die diskrete Koordinatendarstellung vorgestellt, um die grundsätzliche Idee der Parametrisierung darzulegen. Anschließend wurde auf die effektiveren und am meisten genutzten Methoden eingegangen, zu denen sowohl die Parametrisierung mit Polynomen und Splines, als auch die Parametrisierung mittels der Free-Form Deformation gehören. Beide Methoden haben gegenüber der diskreten Koordinatendarstellung den Vorteil, dass deutlich weniger Parameter zur Beschreibung der Geometrie benötigt werden. Somit reduziert sich die Rechenzeit. Im direkten Vergleich hat die Free-Form Deformation den Nachteil, dass die Optimierungsparameter keinen direkten Zusammenhang mit der physikalischen Geometrie haben. Dafür ist mit der Free-Form Deformation jedoch eine multidisziplinäre Optimierung einfacher möglich.

Der Überblick über die Optimierungsmethoden ist in zwei Teile aufgeteilt. Zum einen wurde in einem kurzen Abschnitt auf die gradientenbasierten Algorithmen eingegangen, die den Nachteil haben, dass sie auf Grund eines kleinen Suchraumes nur ein lokales Minimum auffinden. Zum anderen wurde auf die stochastischen Algorithmen eingegangen, wobei ein besonderes Augenmerk auf die genetischen Algorithmen am Beispiel des Differential-Evolution-Algorithmus gelegt wurde. Diese Algorithmen haben den Vorteil, dass sie den Parameterraum großflächig absuchen und somit in den meisten Fällen ein globales Minimum finden. Der Nachteil dieser Algorithmen hingegen ist, dass sie je nach Problem sehr lange zur Konvergenz benötigen und somit in der Praxis durch eine maximale Iterationszahl in der Laufzeit begrenzt sind. Bei einem Abbruch der Rechnung nach einer maximalen Iterationszahl kann keine Aussage über den bis dahin gefundenen besten Punkt getroffen werden, da nicht gesichert ist, dass es sich um ein Minimum handelt.

## 5 Quellen

- [1] Amoiralis, E. I., Nikolos, I.K., "Freeform Deformation Versus B-Spline Representation in Inverse Airfoil Design", Journal of Computing and Information Science in Engineering, Vol. 8, Juni 2008
- [2] Bletzinger, K.-U., Firl, M., Daoud, F., "Techniken der Formoptimierung", 2. Weimarer Optimierungs- und Statistiktage, 2005, [http://www.dynardo.de/fileadmin/Material\\_Dynardo/bibliothek/WOST\\_2.0/WOST\\_2\\_Formoptimierung\\_De.pdf](http://www.dynardo.de/fileadmin/Material_Dynardo/bibliothek/WOST_2.0/WOST_2_Formoptimierung_De.pdf)
- [3] Nocedal, J., Wright, S. J., "Numerical Optimization", Springer-Verlag New York Berlin Heidelberg, 1999
- [4] Paszkowicz, W., "Genetic Algorithms, a Nature-Inspired Tool: Survey of Applications in Material Science and Related Fields", Materials and Manufacturing Processes, 24: 174-197, 2009
- [5] Samareh, J. A., "A survey of shape parameterization techniques", NASA CONFERENCE PUBLICATION, 333-343, Juni 1999