



Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Fachbereich Informatik
Department of Computer Science

Bachelorarbeit

im Bachelor-Studiengang Computer Science

Einbindung von Constraints in eine divergente Optimierungsmethode am Beispiel der Surrogat-assistierten Illumination

von

Jan Kruska

Betreuer: Prof. Dr. Alexander Asteroth
Zweitbetreuer: Alexander Hagg, M.Sc.
Eingereicht am: 26. Februar 2020

Erklärung

Jan Kruska
Grantham Allee 19
53757 Sankt Augustin

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbst angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

.....
Ort, Datum Unterschrift

Inhaltsverzeichnis

Abbildungsverzeichnis	iv
1 Einleitung	2
2 Grundlagen	4
2.1 Genetische Algorithmen	4
2.1.1 Divergente genetische Algorithmen	5
2.1.2 MAP-Elites	6
2.2 Surrogat-Modellierung	7
2.2.1 Gaußprozesse	8
2.2.2 Exploration Exploitation Problem	10
2.2.3 SAIL	12
2.3 Constraints in Optimierungsprozessen	13
2.3.1 Ausschluss durch Definition der Repräsentation	14
2.3.2 Hard Constraints	14
2.3.3 Soft Constraints	15
2.3.4 Vergleich	15
3 Methode	17
3.1 Repräsentation der Domänen	17
3.1.1 Freiformdeformation	17
3.1.2 OpenFOAM	18
3.1.3 Parametrisierung des Gaußprozesses	19
3.2 Radkästen des Velomobils	20
3.2.1 OpenFOAM	21
3.2.2 Constraint	21
3.2.3 Wahl der Features	24
3.3 E-Roller	24
3.3.1 OpenFOAM	25
3.3.2 Constraint	26
3.3.3 Wahl der Features	26
3.3.4 Deformationspunkte	26
4 Auswertung	28
4.1 Radkasten	28
4.1.1 Initiales Validierungsexperiment	28
4.1.2 Erhöhung der Freiheitsgrade	32
4.1.3 Erhöhung der Laufzeit	34
4.2 E-Roller	46
5 Diskussion	53
5.1 Radkasten	53
5.2 E-Roller	54
6 Ausblick	55
6.1 Radkasten	55
6.2 E-Roller	56
7 Literaturverzeichnis	58

Abbildungsverzeichnis

1	Gaußprozess im Ausgangszustand (a) und nach dem Lernen von fünf Datenpunkten (b) (aus Rasmussen und Williams (2008))	8
2	Mittelwert und 2σ -Intervall von Gaußprozessen mit unterschiedlichen Längenmaßen, Beobachtungen sind als "+" dargestellt (aus Rasmussen und Williams (2008))	10
3	Gaußprozess im Ausgangszustand (a) und nach dem Lernen von fünf Datenpunkten (b).	12
4	Teilschritte von SAIL	13
5	3D-Mesh einer Kugel in 4x4x4 FFD-Box	17
6	Deformiertes 3D-Mesh erzeugt durch Bewegung der Deformationspunkte . .	17
7	Radkasten des Velomobils	20
8	Luftwiderstandswerte über Simulationen von vier zufällig verformten Radkästen	22
9	Das modellierte Radausschlagsvolumen $FV_{radausschlag}$	22
10	Radausschlagsvolumen $FV_{radausschlag}$ mit Velomobil	22
11	Das vereinfachte, geschlossene Mesh des Radkastens $FV_{radkasten}$	23
12	Das Differenzvolumen FV_{diff} für den unverformten Radkasten. Aus diesem wird der Strafwert berechnet	23
13	E-Roller (weiß) und das zu deformierende Bauteil (rot)	24
14	Berechneter c_D -Wert über den Lauf von Simulationen von fünf zufällig deformierten Bauteilen	25
15	Die FFD-Box und Deformationspunkte des E-Roller-Bauteils. Pfeil zeigt in Fahrtrichtung	26
16	Deformationspunkte des ersten Experiments	29
17	Vergleich der Luftwiderstände der produzierten Lösungen	30
18	Vergleich der Constrainterfüllung der produzierten Lösungen	30
19	Karte der Luftwiderstandswerte der Lösungen	30
20	Karte der Constraintwerte der Lösungen	31
21	Verbesserungen des Constraints zwischen Version ohne und mit Constraint	32
22	Deformationspunkte des 2. Experiments	33
23	Vergleich der Luftwiderstände der produzierten Lösungen	34
24	Vergleich der Constrainterfüllung der produzierten Lösungen	34
25	Karte der Luftwiderstandswerte der Lösungen	35
26	Karte der Constraintwerte der Lösungen	35
27	Verbesserungen des Constraints zwischen Version ohne und mit Constraint	36
28	Vergleich der Luftwiderstände der produzierten Lösungen	37
29	Vergleich der Constrainterfüllung der produzierten Lösungen	37
30	Karte der Luftwiderstandswerte der Lösungen	37
31	Karte der Constraintwerte der Lösungen	38
32	Regression von Spaltenindex und Luftwiderstand	38
33	Quadratische Regression von Zeilenindex und Luftwiderstand	39
34	Regression von Spaltenindex und Constraint	39
35	Regression von Zeilenindex und Constraint	40
36	Prozentuale Verbesserung der Samples in Akquisegenerationen in Version ohne Constraint	40
37	Prozentuale Verbesserung der Samples in Akquisegenerationen in Version mit Constraint	40
38	Genotypen, die für präzise Evaluation ausgewählten Individuen (ohne Constraint)	41

39	Genotypen, der für präzise Evaluation ausgewählten Individuen (mit Constraint)	42
40	Kategorisierung der Samples, die für präzise Evaluationen ausgewählt wurden (ohne Constraint)	43
41	Kategorisierung der Samples, die für präzise Evaluationen ausgewählt wurden (mit Constraint)	43
42	Die Phänotypen des Individuums (4-3) mit und ohne Constraint	44
43	Die Phänotypen des Individuums (4-24) mit und ohne Constraint	45
44	Die Phänotypen des Individuums (25-22) mit und ohne Constraint	46
45	Die Phänotypen des Individuums (12-12) mit und ohne Constraint	47
46	Karte der Luftwiderstandswerte der E-Roller Domäne	48
47	Regression für c_D gegen Spaltenindex und Zeilenindex	49
48	Prozentuale Verbesserung der Kinder pro Akquisegeneration	50
49	Die Phänotypen des Individuums (1-1), $c_D=0,2255$	50
50	Die Phänotypen des Individuums (23-22), $c_D=0,1979$	51
51	Die Phänotypen des Individuums (9-16), $c_D=0,2091$	51
52	Die Phänotypen des Individuums (20-5), $c_D=0,1970$	52
53	Laufzeiten zur parallelen Evaluation von 10 präzisen Evaluationen	52

Abstract

Evolutionary Algorithms have proved to be a powerful optimization heuristic, to both examine and solve complex problems. Through approaches such as surrogate-modelling, they are applicable even in computationally expensive domains. A common problem that arises however, is the problem of constraints. Constraints often occur as a result of real world restrictions placed upon solutions generated by an evolutionary algorithm. Such restrictions need to be either fulfilled or optimized alongside the primary goal, depending on the type of constraint. Additionally constraints need to be integrated into the evolutionary algorithm, without interfering with its core functionality. This thesis tries to answer the question how such a constraint can be integrated into a divergent evolutionary algorithm and if the integration of this constraint influences the diversity of such an algorithm, one of the main advantages of divergent evolutionary algorithms. The algorithm at hand is the surrogate-assisted Illumination (SAIL) algorithm, which extends the existing divergent evolutionary algorithm MAP-Elites with a surrogate modelling approach, to make it applicable in computationally expensive domains.

Zusammenfassung

Evolutionäre Algorithmen sind ein bewährtes Optimierungsverfahren, zur Untersuchung komplexer Probleme. Durch Verfahren wie der Surrogatassistenz können diese auch in Domänen genutzt werden, in denen die Berechnung der Fitness rechenintensiv ist, weil für diese beispielsweise eine Simulation nötig ist. Häufig tritt bei der praktischen Anwendung von evolutionären Algorithmen das Problem auf, dass neben der Zielfunktion weitere Constraints existieren, welche optimiert beziehungsweise erfüllt sein sollten. Bei der Modellierung von Constraints stellen sich einige Fragen: Wie sollen diese mathematisch formuliert werden? Wie sollen diese in den evolutionären Algorithmus integriert werden? Und wird der Hauptvorteile von divergenten evolutionären Verfahren, die Lösungsdiversität, maßgeblich durch die Einführung des Constraints beeinflusst. In dieser Arbeit soll untersucht werden, ob und wie ein Constraint in einen divergenten evolutionären Algorithmus integriert werden kann und welchen Effekt der Constraint auf die Eigenschaften des evolutionären Algorithmus hat. Dies soll am Beispiel der Surrogat-assistierten Illumination (SAIL) geschehen, einem auf MAP-Elites basierten divergenten evolutionären Algorithmus mit Surrogatassistenz.

1 Einleitung

Evolutionäre Algorithmen haben sich in der Vergangenheit als mächtige Heuristik zur Untersuchung komplexer Problemräume bewiesen (Yu und Gen (2010)). Evolutionäre Algorithmen sind dabei in der Lage kreative Lösungen, in komplexesten Problemräumen zu ermitteln. Genetische Algorithmen sind eine solche Taktik, die an der biologischen Evolution angelehnt entwickelt wurde. Diese entwickeln erst eine Möglichkeit eine funktionale Lösung, auch Phänotyp genannt, durch einen Genotypen, typischerweise ein Vektor, ausdrücken zu können. Mit einer Mapping-Funktion kann aus jedem Genotypen eine konkrete Lösung, ein Phänotyp, erstellt werden. Außerdem wird eine Zielfunktion oder Fitnessfunktion benötigt, mit der die Lösungsqualität eines Individuums, d. h. dessen Phänotyps, ermittelt werden kann. Sind diese Dinge definiert, können die drei Mechanismen der Evolution, Mutation, Crossover und Selektion angewandt werden. Bei der praktischen Anwendung evolutionärer Algorithmen stellen sich allerdings einige Probleme.

Ein sehr häufig auftretendes Problem ist das Problem der zu starken Konvergenz des Algorithmus. Da genetische Algorithmen ohne Diversitätsmanagement typischerweise sehr stark konvergieren können diese dadurch oft in lokalen Optima hängen bleiben. Um dieser Tendenz entgegenzuwirken und diverse Populationen aufrechtzuerhalten wurden verschiedene Taktiken vorgeschlagen, die sich grob in genotypische und phänotypische Diversitätsmanagementverfahren unterscheiden lassen. Verfahren zur Erhaltung genotypischer Diversität messen Diversität, die nichts anderes als das Inverse der Ähnlichkeit einer Population untereinander ist, durch die Ähnlichkeit zweier Genotypen zueinander. Da Genotypen Vektoren sind, kann hier auf eines der vielen Ähnlichkeitsmaße, die für Vektoren existieren, zurückgegriffen werden. Daneben existieren allerdings auch solche Verfahren, die Diversität anhand der Ähnlichkeit der Phänotypen messen. Das hat den Vorteil, dass der Phänotyp die Funktion eines Individuums beschreibt, es wird also funktionale, statt genetischer Diversität hergestellt. Diese Möglichkeit bringt allerdings das weitere Problem, der Definition der phänotypische Ähnlichkeit zweier Individuen, mit sich. Da der Phänotyp stark von der Problemdomäne abhängt, muss diese Ähnlichkeitsdefinition für jede Domäne spezifisch erfolgen. Im Folgenden soll sich mit einer Möglichkeit zum phänotypischen Diversitätsmanagement auseinandergesetzt werden.

Daneben ergeben sich häufig domänenspezifische Probleme. Eines dieser Probleme ist, dass genetische Algorithmen eine große Zahl von Auswertungen der Fitnessfunktion benötigen. Das kann dazu führen, dass genetische Algorithmen in Problemdomänen, in denen die Aus-

wertung der Fitnessfunktion rechenaufwändig ist, in einer unveränderten Form, aufgrund der astronomisch hohen Rechenzeiten, die benötigt würden, schlicht nicht anwendbar sind. Um genetische Algorithmen für solche Probleme nutzen zu können werden somit Möglichkeiten benötigt, durch die die Anzahl an rechenaufwändigen Evaluationen stark reduziert werden kann. Eine solche Möglichkeit, die Erfolg gezeigt hat, sind Surrogat-Modelle. Ein Surrogat-Modell ist eine Machine-Learning Modell, welches mit den Ergebnissen tatsächlicher Evaluationen trainiert wird, um deren Ergebnisse hervorzusagen. Das führt dazu, dass sie statt der tatsächlichen Evaluation im genetischen Algorithmus genutzt werden können. Die Auswertung eines Surrogat-Modells benötigt dabei nur einen Bruchteil der Zeit, die eine tatsächliche Evaluation benötigen würde.

Das letzte große Problem, was sich stellt, ist die Einbindung von Constraints. Häufig existieren neben der primären Zielfunktion die optimiert wird, weitere sekundäre Ziele oder Einschränkungen. Wenn die Ressourcen für eine multivariate Optimierung fehlen, können solche sekundären Optimierungsparameter nicht einfach zu Zielfunktionen erklärt werden. Stattdessen müssen diese Constraints anderweitig in die Algorithmik eingebunden werden. Es gibt verschiedene Ansätze zu Integration von Constraints in Optimierungsprobleme, welcher dieser Ansätze aber am besten funktioniert hängt stark von der Problemdomäne und den spezifischen Constraints ab.

In dieser Arbeit soll untersucht werden wie sich die Einbindung eines Constraints in die Algorithmik eines divergenten genetischen Verfahren auf die Lösungen, die von diesem Verfahren erzeugten werden, auswirkt. Dies soll anhand der Lösung eines Problems der Aerodynamik-Domäne durch Anwendung der Surrogat-assistierten Illumination geschehen, einem Verfahren welches den divergenten genetischen Algorithmus MAP-Elites um eine Surrogat-Assistenz erweitert. Insbesondere die Frage, ob eine Einbindung von Constraints in einen solchen divergenten evolutionären Algorithmus so möglich ist, dass sowohl die eigentliche Zielfunktion, in diesem Falle die aerodynamischen Eigenschaften, als auch der Constraint optimiert werden. Es ist zu erwarten, dass durch die Einbindung eines Constraints die Optimierung der Zielfunktion schlechter ausfallen wird. Es gilt diesen Qualitätsverlust zu quantifizieren und in Relation zum Gewinn bezüglich der Erfüllung des Constraints zu setzen. Auch die Frage ob und wenn ja wie stark, ein solcher Constraint sich auf die Diversität der Lösungen auswirkt, gilt es zu untersuchen.

2 Grundlagen

2.1 Genetische Algorithmen

Genetische Algorithmen sind eine Klasse von Optimierungsalgorithmen, die sich an den Mechanismen der natürlichen Evolution orientieren (Simon (2013)). Namentlich Mutation, Crossover und Selektion. Die Begrifflichkeiten orientieren sich entsprechend an der natürlichen Evolution. Genetische Algorithmen arbeiten typischerweise auf Gruppen von Individuen, auch Population genannt. Diese Populationen entwickeln sich über die Iterationen des Algorithmus, typischerweise Generationen genannt. Jedes Individuum besitzt einen Genotypen sowie einen Phänotypen. Der Genotyp eines Individuums ist meist ein Vektor, das Genom des Individuums genannt, der aus einzelnen Zahlen, den Genen des Individuums, besteht. Dieser stellt die genetischen Informationen eines Individuums dar. Daneben existiert in genetischen Algorithmen eine Mapping-Funktion mit der ein Genotyp in einen Phänotyp, sprich eine konkrete Lösung des Optimierungsproblems, übersetzt werden kann.

Algorithm 1 Genetischer Algorithmus

```
1: procedure GENETICALGORITHM(fitnessFunction,hyperparameters)
2:   population  $\leftarrow$  initializePopulation
3:   populationFitness  $\leftarrow$  fitnessFunction(population)
4:   for numberIterations do
5:     children  $\leftarrow$  crossover(population,hyperparameters)
6:     children  $\leftarrow$  mutate(children,hyperparameters)
7:     childFitness = fitnessFunction(children)
8:     population  $\leftarrow$  selection(population,populationFitness,children,childFitness)
9:     populationFitness  $\leftarrow$  updateFitness(population)
10:   end for
11: end procedure
```

Die Mechanismen der Mutation und des Crossovers agieren auf der Ebene des Genotyps. Mutation beschreibt die Situation, dass jedes Gen zufällig mutieren kann. Mutieren bedeutet, dass eine Mutationswahrscheinlichkeit existiert, mit der sich der Wert eines Gens ändern kann. Abhängig vom Typen des Gens, ist eine Änderung unterschiedlich formuliert, eine reelle Zahl mag sich anhand einer Normalverteilten Zufallsvariable ändern, eine natürliche Zahl mit ± 1 und ein Index mag einen zufälligen anderen möglichen Index annehmen. Mutation ist für einen genetischen Algorithmus wichtig, da sich durch diese neuen Eigenschaften entwickeln können und neues genetisches Material in den Algorithmus Einzug finden kann.

Crossover beschreibt die Situation, dass zwei oder mehr Elternindividuen zu einem Kind-individuum kombiniert werden. Auch hier ist eine konkrete Implementierung nicht vorgeschrieben, wichtig ist nur, dass ein Kind als Kreuzung der Eltern erzeugt werden kann. Das Ziel des Crossovers ist es, dass sich positive Eigenschaften in Populationen verteilen und mehrere unabhängig entstandenen positiven Mutationen in einem Individuum vereint werden können. Es ist allerdings anzumerken, dass Crossover einen optionalen Teil von genetischen Algorithmen darstellt, einige Algorithmen verzichten aus verschiedenen Gründen auf Crossover und mutieren ihre Populationen nur. Diese beiden Methoden stellen sicher, dass erstens neues genetisches Material Einzug in den Algorithmus finden kann, und dass sich genetische Informationen über Generationen in Populationen verteilen können. Dadurch können systematisch neue Genotypen und damit neue Lösungen generiert werden, sie stellen allerdings keinen Mechanismus zur Verfügung, der Lösungen optimieren kann.

Dieser Mechanismus, der der Optimierung eine Richtung gibt, ist die Selektion. Anders als die ersten beiden agiert diese auf Phänotypen, sprich ausgedrückten Lösungen. Jeder genetische Algorithmus benötigt eine Funktion mit der Lösungen bewertet werden können. Dies kann entweder in Form einer Fitnessfunktion, die die Güte eines Individuums berechnet und bei der höhere Werte besser sind, oder in der Form einer Kostenfunktion, die die Kosten eines Individuums berechnet und bei der entsprechend niedrigere Werte besser sind, geschehen. Ob eine Fitness- oder Kostenfunktion genutzt wird hängt von der Problemformulierung ab und ist letztendlich nur die Frage, ob der Algorithmus minimiert oder maximiert. Mit diesen Funktionen kann für jedes Individuum eine Qualität berechnet werden und Individuen können bezüglich dieser verglichen werden. Durch Selektion werden in jeder Iteration dann schlechte Lösungen eliminiert, wodurch sich die Population zu Optima hin entwickelt. Häufig ist die Selektion aus verschiedenen Gründen komplexer, als die einfache Auswahl der besten Individuen. Besonders die Diversität ist eine erwünschte Eigenschaft, die eine komplexere Formulierung der Selektion benötigt. Die in Abschnitt 2.1.1 besprochenen Ansätze zum Diversitätsmanagement, berücksichtigen Diversität von Populationen, bevorzugen dabei solche Lösungen, die die Diversität aufrechterhalten und produzieren damit am Ende eine diversere Ergebnispopulation.

Insgesamt lässt sich ein evolutionärer Algorithmus also wie folgt zusammenfassen. Zuerst wird eine zufällige Startpopulation erzeugt. Daraufhin werden aus den Individuen dieser Population Kinder per Crossover erzeugt und mutiert. Für diese Kinder wird die Fitness berechnet und Selektion durchgeführt, wodurch eine neue Population entsteht, die daraufhin zur neuen Elterngeneration wird. Die Crossover→Mutation→Selektion Schleife wird so oft wiederholt werden bis entweder ein Abbruchkriterium, wie eine zu niedrige Verbesserungsrate, oder die maximale Anzahl an Generationen erreicht ist. Die letzte erzeugte Population stellt dann die Lösungspopulation des Algorithmus dar.

2.1.1 Divergente genetische Algorithmen

Eine sehr häufige Anforderung an genetische Algorithmen ist die Einbindung von Divergenz, bzw. Diversität. Dies hat verschiedene Gründe. Einerseits führt eine niedrige Diversität der Individuen dazu, dass nur ein kleiner Bereich des Suchraums, nämlich der in dem die Individuen liegen abgesucht wird. Sehr homogene Populationen können dadurch einfach in lokalen Optima stagnieren, da andere Optima zu weit im Suchraum entfernt sind. In einer diversen Population ist das Springen aus lokalen Optima hingegen einfacher und selbst wenn Populationen stagnieren, dann geschieht dies in mehreren Optima gleichzeitig, wodurch die Wahrscheinlichkeit ein im globalen Vergleich gutes lokales Optimum zu finden steigt. Auch ist es nicht unbedingt wünschenswert eine sehr homogene Population als Ergebnis zu erhalten, da die Individuen mit hoher Wahrscheinlichkeit alle der gleichen Lösungsklasse angehören. Wenn man eine diverse Population aus Individuen erhält, können Aussagen über unterschiedliche Lösungsklassen getroffen werden, Lösungsklassen und ihre Qualität können miteinander verglichen werden, und je nachdem kann größere Erkenntnis erlangt werden, was eine qualitativ hochwertige Lösung ausmacht, wodurch wiederum das gestellte Problem besser verstanden werden kann.

Es existieren verschiedenste Ansätze, um Diversität zu gewährleisten, welche sind grundsätzlich in genotypische und phänotypische Ansätze aufteilen lassen, abhängig davon auf welcher Abstraktionsebene Diversitätsmanagement stattfindet. Ansätze wie Niching (Mahfoud (1996), Shir (2012)), gewährleisten genotypische Diversität durch die Ermittlung der genetischen Ähnlichkeit zweier Individuen. Das hat den Vorteil, dass der Genotyp eines Individuums typischerweise ein Vektor ist, und eine Vielzahl von Ähnlichkeits- und Distanzmetriken für Vektoren existiert. Dadurch fällt die Definition, was Diversität bedeutet sehr leicht. Auch ist die Berechnung einer Distanz zwischen Vektoren sehr effizient, was vorteilhaft ist, da die Berechnung der Ähnlichkeit zweier Individuen sehr häufig stattfinden muss.

Allerdings hat es den Nachteil, dass, besonders wenn die Genotyp-zu-Phänotyp-Mapping Funktion komplex ist, genetische Distanz nicht der funktionalen Distanz zwischen Lösungen entspricht. D. h. zwei genetisch unähnliche Lösungen, können trotzdem der gleichen Lösungsklasse angehören, sind also funktional ähnlich. Oft ist eher funktionale Diversität, sprich Distanz der Phänotypen, gewünscht, nicht Diversität der Enkodierung. Deshalb wurden Ansätze wie Novelty-Search (Lehman und Stanley (2011)) oder MAP-Elites (Mouret und Clune) zum Diversitätsmanagement vorgeschlagen, die Diversität auf der Ebene, des Phänotyps bestimmen und somit die funktionale Distanz messen. Da der Phänotyp domänenspezifisch ist, muss die Diversitätsmetrik für jede Domäne maßgeschneidert definiert werden, und es kann nicht auf Allgemeinlösungen zurückgegriffen werden. Auch besteht eine erhebliche Limitierung bezüglich der Komplexität einer solche Diversitätsmetrik, da die Ähnlichkeit zweier Phänotypen sehr häufig berechnet werden muss.

2.1.2 MAP-Elites

Algorithm 2 MAP-Elites

```

1: procedure MAPELITES(fitnessFunction,categorizationFunction,hyperparameters)
2:   map  $\leftarrow$  ▷ Initialize empty n-dimensional map
3:   fitness  $\leftarrow$  ▷ Initialize empty n-dimensional fitnessMap
4:   population  $\leftarrow$  initializeRandomPopulation ▷ Start of with randomly generated intitial samples
5:   for individualin population do
6:     f = fitnessFunction(individual)
7:     c = categorizationFunction(individual) ▷ Categorization c is an Index to a cell in the map
8:     if map(c) =  $\emptyset$   $\vee f < fitness(c)$  then ▷ If cell is unoccupied or this solution is better put it in cell
9:       map(c)  $\leftarrow$  individual
10:      fitness(c)  $\leftarrow$  f
11:    end if
12:   end for
13:   for numberGenerations do
14:     children  $\leftarrow$  selectRandom(population,childrenPerGeneration) ▷ Select samples randomly from existing solutions
15:     children  $\leftarrow$  variate(children,hyperparameters) ▷ Variate these children with crossover and/or mutation
16:     for individualin children do
17:       f = fitnessFunction(individual)
18:       c = categorizationFunction(individual)
19:       if map(c) =  $\emptyset$   $\vee f < fitness(c)$  then
20:         map(c)  $\leftarrow$  individual
21:         fitness(c)  $\leftarrow$  f
22:       end if
23:     end for
24:   end for return map,fitness
25: end procedure

```

MAP-Elites (Multi-dimensional Archive of Phenotypic Elites)(Mouret und Clune) ist ein genetischer Algorithmus mit phänotypischem Diversitätsmanagement. Das bedeutet, dass die Morphologie bzw. Funktionalität einer konkreten ausgedrückten Lösung, wie beispielsweise Volumen oder Krümmung eines Bauteils, betrachtet wird. Dazu wird der Lösungsraum entlang anhand beliebig vieler Features aufgeteilt, deren Untersuchung als zielfüh-

rend zum Verständnis des Problems befunden wird. Diese Feature-Dimensionen sind direkte Merkmale der Phänotypen von Lösungen, wie beispielsweise das Volumen eines Phänotyps in einer dreidimensionalen Domäne. Als Features sollten solche Merkmale gewählt werden, die unabhängig von der Zielfunktion sind, d. h. deren Einfluss auf die Fitness von Lösungen unklar ist. Gerade die Untersuchung wie Merkmale mit der Fitness interagieren ist ein typisches Untersuchungsziel bei der Nutzung von MAP-Elites. Diesen Feature-Dimensionen wird ein Minimum, Maximums sowie eine Schrittweite zugewiesen um den Lösungsraum in Zellen zu diskretisieren. Diese Diskretisierung zu Zellen wird typischerweise Karte oder Archiv genannt. Essentiell für MAP-Elites ist, dass jede Zelle dieser Karte nur maximal eine Lösung enthalten kann.

Zu Beginn des Algorithmus werden alle Zellen zuerst einmal leer initialisiert. Im Laufe des Algorithmus werden diese nach und nach gefüllt. Dazu werden zuerst zufällige Individuen generiert um die Karte mit einer Initialpopulation zu befüllen. Ist dies erfolgt werden daraufhin in jeder Generation aus den momentan in der Karte enthaltenen Lösung zufällig Individuen ausgewählt. Aus dieses ausgewählten Individuen wird eine Kindpopulation per Mutation und/oder Crossover erzeugt. Jedes dieser Kinder wird bezüglich der Fitness evaluiert und dessen zugehöriger Phänotyp wird nach den gewählten Features kategorisiert. Diese Kategorisierung weist jedem Kind eine Zelle zu, die es befüllen könnte. Ist diese Zelle noch leer befüllt das Kind diese, befindet sich bereits ein Individuum in der Zelle, findet zwischen diesem und dem Kind lokal Konkurrenz statt.

Die lokale Konkurrenz innerhalb der Zellen ist es, was MAP-Elites zu einem divergenten Algorithmus macht. Da Lösungen nur durch Lösungen verdrängt werden können, denen die gleiche Zelle zugewiesen ist, können Lösungen nur durch phänotypisch ähnliche Lösungen ersetzt werden. Dadurch wird die phänotypische Diversität des Algorithmus gewährleistet. Außerdem kann die Größe des Archivs während des Algorithmus nur zunehmen, da Individuen nur verdrängt werden können, wenn sie durch ein anderes ersetzt werden, aber im Laufe des Algorithmus immer mehr leere Zellen befüllt werden können. Dass bedeutet, dass die Anzahl an lokalen Optima, die der Algorithmus ermittelt hat, über dessen Laufzeit wächst. Ein Vorteil von MAP-Elites ist es, dass nicht nur wird eine Vielzahl von Lösungen generiert wird, die unterschiedliche Lösungsklassen abdecken, sondern die regelmäßig aufgeteilte Karte kann auch dabei helfen die Effekte, die die Features auf die Lösungsqualität haben, zu verstehen.

2.2 Surrogat-Modellierung

Für die Selektion von Individuen innerhalb eines genetischen Algorithmus wird die Lösungsqualität dieser Individuen, typischerweise Fitness genannt, benötigt. Durch die große Anzahl an Individuen, die für den evolutionären Algorithmus benötigt werden, findet die Auswertung der Fitnessfunktion sehr häufig statt. Dies stellt bei relativ einfachen Fitnessfunktionen keine große Einschränkung dar, auf Problemdomänen in denen die Auswertung der Fitness eines Individuums allerdings komplexer und dadurch zeitaufwändiger ist, kann dies die Anwendbarkeit einfacher genetischer Algorithmen einschränken.

Aerodynamische Probleme, für deren Fitnessauswertung zeitaufwändige Simulationen nötig sind, gehören ohne Zweifel zu der Klasse von Problemen, für die die Anzahl der benötigten Funktionsauswertungen zu groß sind, als dass der Algorithmus in vertretbarer Laufzeit abschließen kann. Um evolutionäre Algorithmen auf eine solche Problemdomäne anzuwenden, wird eine Möglichkeit benötigt, um die Anzahl der benötigten Funktionsauswertungen erheblich zu reduzieren. Eine solche Möglichkeit ist die Nutzung eines Surrogatmodells (Jin (2011), Preen und Bull (2016)), eine Machine-Learning Modell, welches aufgrund echter Simulationsauswertungen trainiert wird, um deren Ergebnis annähernd vorherzusagen. Eine Auswertung des Modells erfordert dabei nur einen winzigen Bruchteil des Aufwands, der für eine Simulation nötig wäre.

Die Einführung eines Surrogatmodells bringt allerdings einige Probleme mit sich. Das wichtigste zu lösende Problem ist, wie bestimmt wird anhand welcher realen Funktionsauswertungen das Surrogatmodell trainiert wird. Dabei müssen einige Ziele beachtet werden. Erstens sollte das Surrogatmodell möglichst präzise sein, da es seinen Zweck für reale Funktionsauswertungen einzustehen nur erfüllen kann, wenn die Vorhersagen des Surrogatmodells ausreichend gut reale Funktionsauswertungen abbilden. Da das Surrogatmodell aber genau für den Zweck genutzt wird die Anzahl an benötigten teuren Funktionsauswertungen zu reduzieren, wäre es sinnvoll, wenn Funktionsauswertungen so gewählt werden, dass sie maximalen Nutzen bringen.

2.2.1 Gaußprozesse

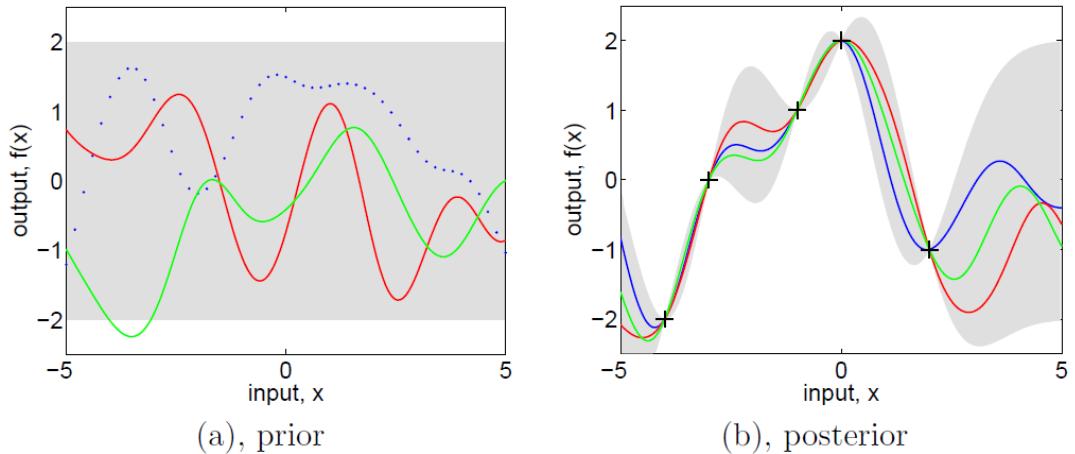


Abb. 1: Gaußprozess im Ausgangszustand (a) und nach dem Lernen von fünf Datenpunkten (b) (aus Rasmussen und Williams (2008))

Gaußprozesse sind ein in Rasmussen und Williams (2008) entwickeltes Machine-Learning Verfahren, mit welchem beliebige mathematische Funktionen approximiert werden können. Ein großer Vorteil von Gaußprozessen ist, dass sie durch ihre Herkunft aus der Statistik die zu approximierende Funktion als Wahrscheinlichkeitsverteilung modellieren. Wie jedes überwachte Machine-Learning Verfahren zur Regression erhalten Gaußprozesse dabei eine Reihe an Beobachtungen auf deren Basis das Modell konstruiert wird. Vor dem Lernen sind Gaußprozesse als Basisvermutung initialisiert. Diese Basisvermutung besteht aus einer Mittelwerts- und einer Varianzfunktion, die entsprechend den Mittelwert und Varianz für alle Eingabewerte vorgeben. Die Funktionen sind frei wählbar, wenn allerdings nichts über das Problem bekannt ist, so wird meist auf konstante Funktionen zurückgegriffen. In Abb. 1 (a) ist eine solche Basisannahme, mit konstanter Mittelwertfunktion $m(x) = 0$ und konstanter Varianzfunktion $v(x) = 2$ dargestellt (graues Band). Außerdem sind 3, der unendlich vielen möglichen Funktionen, die die Daten beschrieben eingezzeichnet.

Diese Basisannahme wird daraufhin mit einer Reihe von Beobachtungen trainiert, wodurch sich das in Abb. 1 (b) dargestellte Ergebnis ergibt. Vom Zustand Abb. 1 (a) zu Abb. 1 (b) wurde der Gaußprozess auf Basis von fünf Datenpunkten trainiert (als +-Symbole dargestellt). An jeder Beobachtung wird der Mittelwert auf den beobachteten Wert gesetzt und die Varianz an dieser Stelle kollabiert zu null, da dort absolute Sicherheit bezüglich des Wertes besteht. Auch die Varianzen um die Datenpunkte nehmen stark ab, werden allerdings nicht null. Je näher ein Punkt an tatsächlichen Beobachtungen liegt, desto kleiner ist die Varianz. Dies ist durch die Grundannahme zu begründen, dass die zu approximie-

rende Funktion als stetig behandelt werden kann, d. h. es wird angenommen, dass kleine Änderungen der Eingabewerte in kleinen Änderungen der Funktionswerte resultieren.

Gaußprozesse werden primär durch die Wahl der drei innerhalb des Gaußprozesses benötigten Funktionen und deren Hyperparameter gesteuert. Die Mittelwertsfunktion ist die anfängliche Annahme des Mittelwerts des Gaußprozesses. Durch diese wird die Grundannahme des Gaußprozesses in Bereichen, in denen keine Datenpunkte bekannt sind, definiert. Wie diese Funktion aussehen kann, ist nicht weiter beschränkt, die Daten sollten allerdings durch diese Funktion beschreibbar sein. Ein linearer Mittelwert kann beispielsweise niemals korrekt durch eine konstante Mittelwertfunktion beschrieben werden. In Abb. 1 ist ein Beispiel einer konstanten Mittelwertfunktion abgebildet. Die Hyperparameter der Mittelwertsfunktion hängen stark von der Funktion ab, die hier gewählt wird, im Falle von polynomiellen Funktionen wären dies die Koeffizienten.

Typischerweise wird davon ausgegangen, dass die Eingabedaten, mit denen der Gaußprozess trainiert, nicht rauschfrei sind. Zur Beschreibung des Rauschens des Signals existiert die Wahrscheinlichkeits-Funktion. Diese Funktion ist eine Wahrscheinlichkeitsverteilung, die das Rauschen beschreibt. Die typische Annahme ist, dass das Rauschen normalverteilt ist. Wenn die Wahrscheinlichkeitsfunktion eine Normalverteilung ist, wird diese nur durch σ parametrisiert. In Szenarien in denen eine Normalverteilung zur Beschreibung des Rauschens nicht korrekt ist, müssen stattdessen andere Verteilungen herangezogen werden. Hyperparameter der Wahrscheinlichkeits-Funktion sind die Hyperparameter der gewählten Verteilung.

Die letzte und wichtigste der Funktionen ist die Kovarianzfunktion. Die Aufgabe der Kovarianzfunktion ist es die Kovarianz zwischen Punkten zu berechnen. Damit wird durch die Kovarianzfunktion gesteuert welche Punkte für wie ähnlich befunden werden und damit, wie Beobachtungen sich auf ihre Umgebung auswirken. Der wichtigste Hyperparameter der Kovarianzfunktion ist das Längenmaß ℓ . Das Längenmaß bestimmt wie weit der Einfluss einer Beobachtung reicht. Bei kleinen Längenmaßen steigt die Varianz schnell an je weiter man sich von einer Beobachtung weg bewegt, bei größeren Längenmaßen haben Beobachtungen einen weiteren Einfluss und die Funktionen werden glatter. In Abb. 2 ist ein Gaußprozess mit unterschiedlichen Längenmaßen dargestellt. Es ist zu erkennen das Abb. 2 (b) wesentlich steiler ist als Abb. 2 (a & c). Auch ist zu erkennen, dass für $\ell = 0.3$ die Varianz schnell ansteigt, wenn man sich von Beobachtungen entfernt. Im Gegensatz dazu ist in Abb. 2 (c) der Gaußprozess mit $\ell = 3$ dargestellt. Dessen Mittelwert ist wesentlich glatter, beschreibt die Punkte aber nicht gut. Stattdessen wird für diese höheren Längenmaße ein größeres σ_n der Wahrscheinlichkeits-Funktion benötigt. Kurz gesagt beschreibt der Gaußprozess mit $\ell = 3$ eine glatte Funktion mit starkem Rauschen. Es sollte klar erkennbar sein, dass Gaußprozess (a) die Daten am besten beschreiben kann, (b) führt sehr viele Bereiche mit hoher Varianz ein, die für diese Daten nicht angemessen sind, und (c) kann die Punkte nicht ordentlich approximieren und benötigt zur Beschreibung der Daten stattdessen ein starkes Rauschen. Die Frage wie die "beste" Parametrisierung eines Gaußprozesses systematischer gefunden werden kann ist enorm wichtig.

Aufgrund der erheblichen Schwierigkeit, die mit der Wahl der korrekten Hyperparameter verbunden ist, wird in Rasmussen und Williams (2008) ein Verfahren zur automatischen Bestimmung von passenden Hyperparametern beschrieben. Auf die Eigenheiten der Modellbestimmung durch Log-Marginal-Likelihood-Verfahren wird hier nicht weiter eingegangen werden, es soll nur erwähnt werden, dass ein solches Verfahren existiert, mit dem die Hyperparameter an die Daten angepasst werden können.

Der Hauptnachteil von Gaußprozessen ist die relativ hohe Speicher- und Rechenkomplexität. Da die Anzahl an Simulationen, die durchgeführt werden können aber ohnehin durch den Rechenaufwand, der mit diesen verbunden ist, stark limitiert ist, wird die Anzahl an Trainingsdaten so gering bleiben, dass Komplexitätsüberlegungen hinfällig sind.

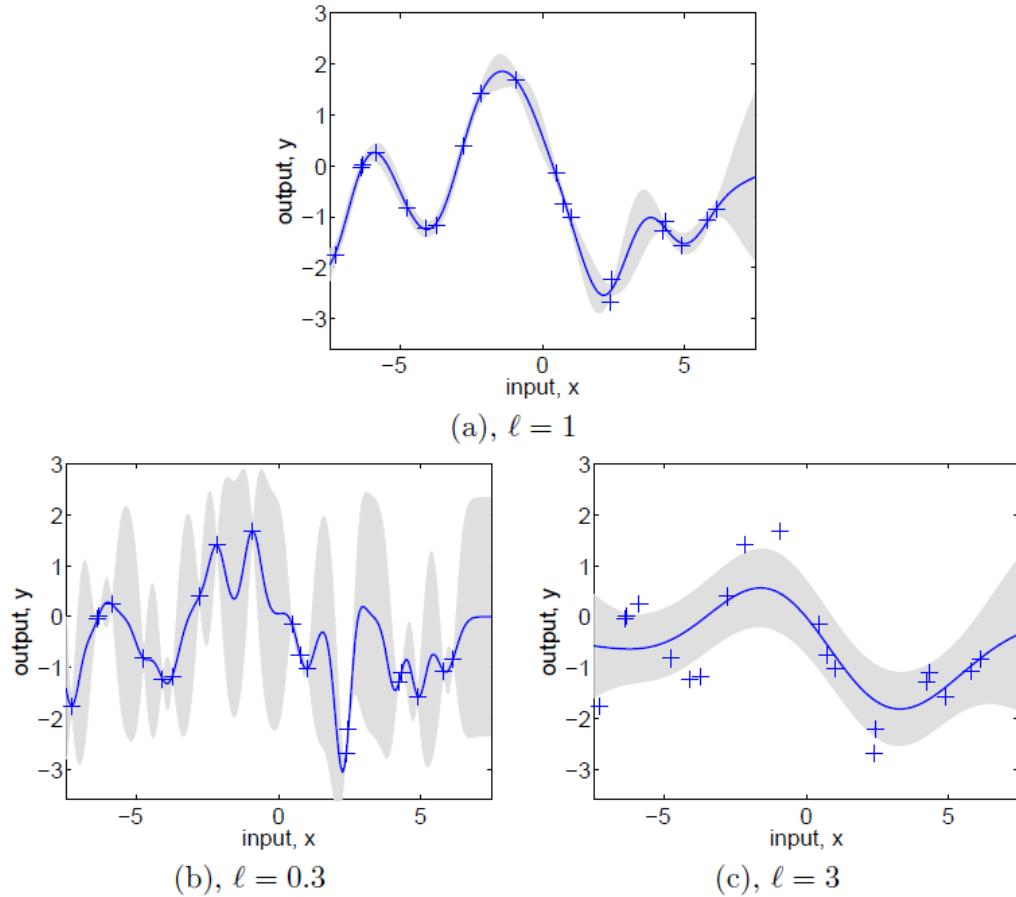


Abb. 2: Mittelwert und 2σ -Intervall von Gaußprozessen mit unterschiedlichen Längenmaßen, Beobachtungen sind als "+" dargestellt (aus Rasmussen und Williams (2008))

Der Vorteil von Gaußprozessen liegt in der Art, wie diese die zu approximierende Funktion beschreiben. Ein Gaußprozess liefert im Gegensatz zu anderen Machine-Learning Verfahren nicht nur eine Vorhersage, sondern den Mittelwert und die Varianz der Vorhersage. Hauptvorteil dieser Tatsache ist, dass der Gaußprozess zwar Aussagen in Bereichen, zu denen er keine Trainingsdaten erhalten hat, trifft, aber die eigene Unsicherheit bezüglich dieser Ergebnisse Teil dieser ist. Dadurch wird automatisch eine Aussage über die eigene Sicherheit bezüglich der Vorhersage getroffen, wodurch fehlerhafte Aussagen, die andere Machine Learning Verfahren weit entfernt von Trainingsdaten treffen vermieden werden können.

Dies ist allerdings nicht der Grund, weshalb Gaußprozesse für SAIL genutzt werden. Stattdessen tritt auch in SAIL ein typisches Problem von Reinforcement Learning, die Spannung zwischen Exploration und Exploitation, auf. Gaußprozesse bieten mit ihrer Vorhersage von Mittelwert und Varianz eine perfekte Grundlage zur Anwendung des Upper-Confidence Bound Verfahrens, einer Methode zum Umgang mit diesem Dilemma. Auf das Dilemma und Upper-Confidence-Bound als Lösungsmöglichkeit wird im Folgenden in Abschnitt 2.2.2 weiter eingegangen. Da sich Gaußprozesse hervorragend zur Anwendung von Upper-Confidence Bound eignen, werden diese in SAIL als Surrogatmodell genutzt.

2.2.2 Exploration Exploitation Problem

Ein häufig auftretendes Dilemma, wenn der Gewinn eines unbekannten Problems maximiert werden soll, ist die Spannung zwischen Exploration und Exploitation. Das einfachste Beispiel, in dem eine solche Spannung existiert, ist das Problem des mehrarmigen Bandits.

ten (Robbins (1952)). In diesem Problem existiert eine Reihe von Glücksspielautomaten, deren Gewinn durch statistische Verteilungen beschrieben wird. Der Gewinn jedes einzelnen Automaten wird durch eine andere Verteilung definiert. Diese Verteilungen sind einem Spieler, mit dem Ziel den eigenen Gewinn zu maximieren, allerdings nicht bekannt. Mit jedem Spiel an einem Automaten erhält der Spieler sowohl den Gewinn des Spiels, als auch Wissen über die Gewinnverteilung des Automaten und damit über das gesamte Problem. Das Problem ist die Frage, mit welcher Strategie der Spieler seinen Gewinn maximieren kann. Einerseits könnte der Spieler nur den Automaten spielen, der den besten bekannten erwarteten Gewinn hat, allerdings besteht die Möglichkeit, dass ein anderer unbekannter Automat einen höheren erwarteten Gewinn besitzt, von dem der Spieler allerdings nicht weiß. Die kurzfristig beste Strategie kann also langfristig nicht optimal sein. Stattdessen kann der Spieler versuchen Wissen über das Problem zu sammeln, indem er an Automaten spielt über deren Verteilung er wenig weiß. Dies führt aber dazu, dass der Spieler an vielen schlechten Automaten spielen wird, um mehr Wissen über diese zu erlangen. Das Dilemma zwischen diesen Entscheidungen ist ein Beispiel für das Dilemma zwischen Exploration und Exploitation. An dem besten bekannten Automaten zu spielen ist ein Beispiel der Exploitation, der Ausnutzung des Wissens, an unbekannten Automaten zu spielen ein Beispiel der Exploration, der Sammlung von neuem Wissen.

Allgemeiner formuliert beschreibt Exploration dabei die Untersuchung von unbekannten, bisher unerforschten Gebieten, um das zugrundeliegende Problem besser zu verstehen und damit Strategien besser optimieren zu können, während Exploitation die Ausnutzung des gesammelten Wissens beschreibt, um den Nutzen zu maximieren. Die beiden Herangehensweisen stellen gegensätzliche Konzepte dar. Je mehr Exploration durchgeführt wird, desto weniger Exploitation kann durchgeführt werden und umgekehrt.

Ein solches Problem stellt sich auch bei der Surrogatmodellierung. So ist einerseits die Exploration unerforschter Gebiete wichtig um die globale Präzision des Modells zu maximieren und Optima in unerforschten Bereichen zu finden. Andererseits ist die Exploitation wichtig um Optimalregionen möglichst präzise abzubilden, da in diesen Regionen der Hauptteil der Optimierung stattfindet. Ist das Surrogatmodell in Regionen hoher Fitness nicht präzise, beschränkt das die maximale Qualität der Lösungen, die der Optimierungsalgorithmus, der auf das Surrogatmodell zurückgreift, erzeugen kann. Außerdem ist die Anzahl der erlaubten realen Auswertungen stark begrenzt, die Reduzierung dieser ist gerade der Grund für die Nutzung eines Surrogatmodells. D. h. dass die Auswertungen, die durchgeführt werden sollten maximalen Nutzen für das Surrogatmodell haben.

Eine Möglichkeit mit dem Dilemma zwischen Exploration und Exploitation umzugehen ist das in Auer (2002) beschriebene Upper-Confidence-Bound Verfahren. Dabei werden Exploration-Exploitation Probleme als statistischen Prozesse bestehend aus Mittelwert und Varianz aufgefasst. Der Mittelwert an einem Punkt spiegelt dabei den erwarteten Nutzen wider, eine dem Mittelwert folgende Auswahl bevorzugt also Exploitation von Bereichen, in denen der erwartete Nutzen hoch ist. Die Varianz spiegelt die Unsicherheit an einem Punkt wider, eine Auswahl nach Varianz bevorzugt die Exploration von Bereichen in denen eine hohe Varianz auftritt, also eben solche Bereiche in denen noch keine Datenpunkte existieren. Upper-Confidence-Bound summiert Mittelwert und Varianz zur Bestimmung der Auswahl der nächsten Auswertung und erreicht damit eine Balance in der sowohl Exploration als auch Exploitation stattfindet.

Gaußprozesse eignen sich durch ihre Eigenschaft an jedem Punkt Mittelwert und Varianz des Gaußprozesses zu berechnen hervorragend zur Anwendung von Upper Confidence Bound. In Abb. 3 sind die drei nächsten Punkte, die durch Upper Confidence Bound zur Auswertung bestimmt würden dargestellt. Die beiden Punkte nahe null stellen dabei Beispiele von Exploitation dar. Beide Punkte liegen nahe an einer tatsächlichen Auswertung, durch den hohen Mittelwert in diesem Bereich sollte dieser aber genauer untersucht werden. Der Punkt am rechten Rand bei $x = 5$ stellt ein Beispiel von Exploration dar.

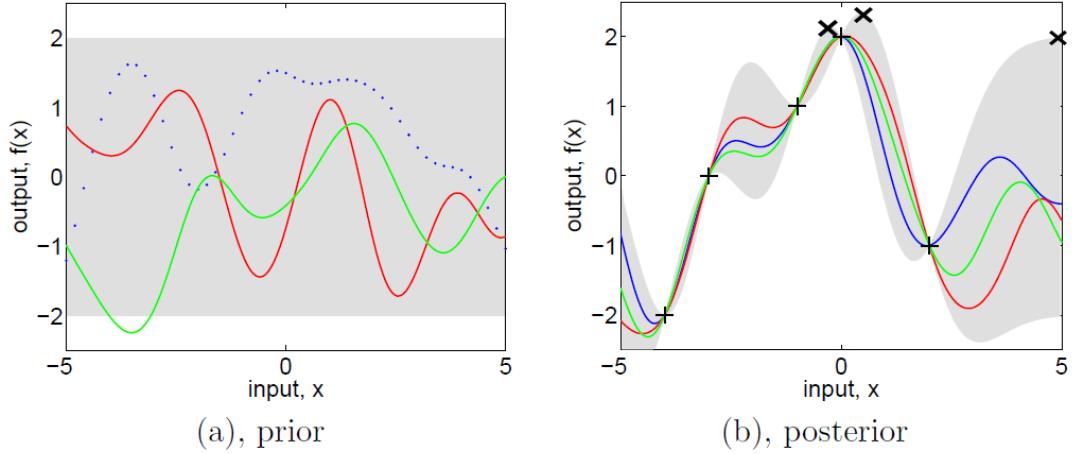


Abb. 3: Gaußprozess im Ausgangszustand (a) und nach dem Lernen von fünf Datenpunkten (b). Die drei nächsten Punkte, die per UCB ausgewählt werden sind mit "x" markiert (aus Rasmussen und Williams (2008))

Der Gaußprozess weiß über diesen Bereich wenig, folglich besteht dort eine hohe Varianz. Durch die große Varianz wird dieser Punkt trotz des niedrigen Mittelwerts ausgewählt.

2.2.3 SAIL

Algorithm 3 MAP-Elites

```

1: procedure MAPELITES(evaluate,categorizationFunction,hyperparameters)
2:   while numberEvaluatedSamples < totalSamples do
3:     if notInitialized then
4:       evaluatedSamples  $\leftarrow$  evaluate(randomSampling())  $\triangleright$  Evaluate is the real
          costly Fitness-Function.
5:     else
6:       surrogate  $\leftarrow$  trainSurrogate(evaluatedSamples)
7:       acquisitionMap  $\leftarrow$  MapElites(acqFunc,categorizationFunc,hyperparameters)
8:       nextSamples  $\leftarrow$  sampleRandomFromMap(acquisitionMap)  $\triangleright$  SAIL
          utilizes Sobol-sequences to achieve optimal spread of the Sampling
9:       evaluatedSamples  $\leftarrow$  evaluatedSamples  $\cup$  evaluate(nextSamples)
10:    end if
11:   end while
12: end procedure

```

SAIL (Surrogate-Assisted Illumination) ist ein in Gaier et al. beschriebenes Verfahren, welches die in Abschnitt 2.1.2 und Abschnitt 2.2 beschriebenen Ansätze vereint, mit dem Ziel MAP-Elites in rechenintensiven Domänen anwendbar zu machen.

Zur Initialisierung SAILS werden zuerst zufällig Samples ausgewählt und evaluiert. Um eine gleichmäßige Verteilung der Samples über den Problemraum zu garantieren werden diese Samples nicht aus einer Gleichverteilung, sondern aus einer Sobol Sequenz (Sobol' (1967)) ausgewählt. Der eigentliche Kern von SAIL ist die graduelle Verfeinerung des Surrogatmodells. In jeder Iteration der Verfeinerungsschleife wird ein Gaußprozess auf Basis aller bisher ausgewerteten Samples trainiert. Mit diesem Gaußprozess wird MAP-Elites durchgeführt, wobei als Fitnessfunktion für die Durchführung von MAP-Elites die Akquisefunktion genutzt wird. Die Akquisefunktion ist dabei Upper Confidence Bound. Dadurch

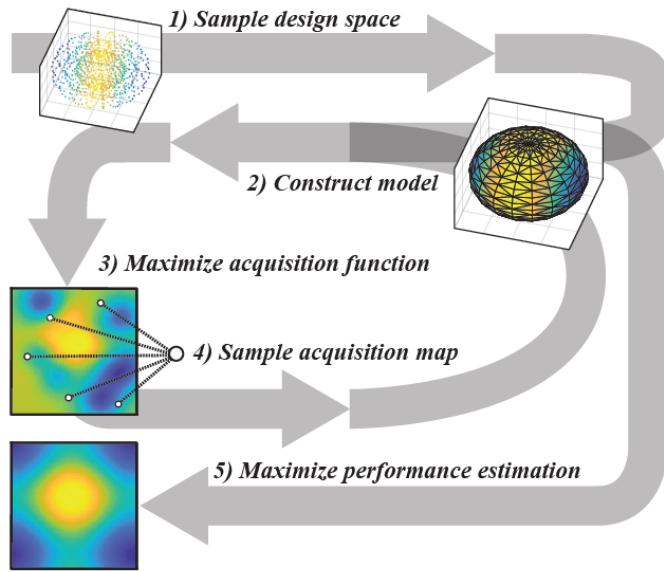


Abb. 4: Teilschritte von SAIL, 1) zufälliges Sampling des Problemraums, 2) Trainieren eines Surrogatmodells, 3) Erstellung der Akquisekarte mit MAP-Elites mit Upper-Confidence-Bound als Fitness, 4) Auswahl der Individuen mit höchstem UCB-Wert und reale Auswertung, 5) Finale Auswertung mit MAP-Elites

wird die sogenannte Akquisekarte erzeugt, welche diverse Lösungen enthält, die die Akquisefunktion maximieren. Aus dieser Akquisekarte werden dann mehrere Lösungen für reale Fitnessevaluationen ausgewählt. Durch die Nutzung von Upper-Confidence-Bound als Akquisefunktion stellen diese Lösungen Optima bezüglich Exploration und Exploitation dar. Die ausgewählten Samples werden daraufhin ausgewertet und den bereits ausgewerteten hinzugefügt, um den Gaußprozess in der nächsten Iteration besser zu trainieren. Dies kann abhängig von den verfügbaren Rechenkapazitäten beliebig oft wiederholt werden. Am Ende produziert diese Schleife eine Menge an echten ausgewerteten Samples auf deren Basis der Gaußprozess trainiert werden kann.

In der finalen Auswertung wird dann ein Gaußprozess auf Basis aller ausgewerteten Samples trainiert. Mit diesem Gaußprozess wird dann ein letztes Mal MAP-Elites durchgeführt, statt der Akquisefunktion wird allerdings die Fitnessfunktion optimiert, die durch den Mittelwert des Gaußprozesses beschrieben wird. Der Mittelwert des Gaußprozesses an einem Punkt entspricht gerade der mittleren Vorhersage zur Fitness einer realen Auswertung an diesem Punkt. Damit wird dann eine diverse Vorhersagekarte erzeugt, die die Optima des Gaußprozesses zeigt. Bei korrekter Parametrisierung von SAIL, des Gaußprozesses, und einer ausreichend großen Anzahl an realen Auswertungen korrespondieren Optima im Gaußprozess mit Optima in der realen Fitnessfunktion.

In Gaier et al. wurde gezeigt, dass dieser Ansatz erfolgreich auf zweidimensionale und dreidimensionale aerodynamische Domänen angewandt werden kann. Auch dort wurden per Freiform-Deformation dreidimensionale Objekte deformiert.

2.3 Constraints in Optimierungsprozessen

Constraints sind eine Möglichkeit anderweitige Beschränkungen, welche eingehalten werden sollen oder müssen, in einen Optimierungsprozess einzuarbeiten, ohne auf multivariate Optimierung zurückgreifen zu müssen. Dass Beschränkungen existieren, wie Lösungen aussehen dürfen, ist sehr häufig der Fall. Die einfachste Möglichkeit ist es die Genotypenkodierung und die Mapping-Funktion so zu wählen, dass nur Lösungen generiert werden können, die die Beschränkung erfüllen. Ist dies nicht möglich, muss eine explizite Behand-

lung des Constraints stattfinden. Constraints lassen sich grundsätzlich in Soft Constraints, bei denen die Nichteinhaltung des Constraints zur Addition von Strafwerten auf die Fitnessfunktion führt, und Hard Constraints, die eine binäre erfüllt/nicht erfüllt Auswahl treffen, unterscheiden. Welche der drei Arten genutzt wird, hängt stark von der Problemstellung ab.

Auch stellt die Formulierung von Constraints häufig eine Herausforderung dar. So müssen diese offensichtlich berechenbar sein, Constraints sind allerdings häufig nicht in mathematischer Form formuliert, sondern eher in einer Art die an Requirements erinnert. Es muss folglich zuerst eine mathematische Formulierung für den Constraint entwickelt werden, die dem Constraint so weit wie möglich entspricht. Zweitens werden Constraints in einem typischen evolutionären Optimierungsalgorithmus sehr häufig evaluiert werden. Das bedeutet, dass die Berechnung des Constraints entweder nicht besonders rechenintensiv sein darf, oder Hilfsmechanismen integriert werden wie beispielsweise die Modellierung des Constraint durch ein Surrogatmodell, wie es in SAIL für die Fitness bereits geschieht.

2.3.1 Ausschluss durch Definition der Repräsentation

Die einfachste Art mit äußeren Einschränkungen umzugehen, ist es die Repräsentation des Problems so zu formulieren, dass es nicht möglich ist Phänotypen zu generieren, die den Constraint verletzen. Im Gegensatz zum Phänotyp, dessen Definition von der Problemdomäne abhängt, kann der Genotyp frei definiert werden. Die einzigen Kriterien, die der Genotyp erfüllen muss, sind, dass eine Abbildung existiert mit der aus einem Genotyp ein zugehöriger Phänotyp generiert werden kann, dass die Methoden der evolutionären Optimierung, sprich Mutation und Crossover, mit ihm durchgeführt werden und dass neue zufällige Genotypen generiert werden können. Die Wahl einer guten Repräsentation des Problems ist einer der wichtigsten Teile der evolutionären Optimierung. So existieren Beispiele, in denen das gleiche Problem mit einer Repräsentation nicht gelöst werden kann, während dies mit einer anderen Repräsentation möglich ist. Constraints können in der Wahl der Problemrepräsentation bereits behandelt werden. Ist es möglich, durch eine kluge Wahl der Repräsentation zu garantieren, dass nur solche Genotypen generiert werden können deren zugehöriger Phänotyp den Constraint erfüllt, ist sichergestellt, dass jede mögliche Lösung valide ist, und der Constraint nicht weiter behandelt werden muss.

Dies hat den Vorteil, dass ungültige Lösungen niemals in den Algorithmus einfließen, und damit weder Rechenkapazitäten für nicht nutzbare Lösungen genutzt werden, noch solche Eigenschaften, durch die eine Lösung die Constraints verletzt, vom Algorithmus überhaupt in Erwägung gezogen werden. Dies kann eine sehr einfache und effektive Art sein um mit Constraints umzugehen und sollte, sofern es möglich ist, die erste Wahl zu Behandlung von Constraints sein. Häufig ist es allerdings so, dass die Definition einer Repräsentation, die den Constraint erfüllt, eine große Herausforderung darstellt. Besonders in solchen Problemdomänen, in denen die Genotyp-Phänotyp Abbildung komplex ist, ist es häufig sehr schwer die Repräsentation so zu wählen, dass sie die Verletzung eines beliebigen Constraint ausschließt.

2.3.2 Hard Constraints

Ist es nicht möglich durch die Wahl der Repräsentation Lösungen auszuschließen, die den Constraint verletzen, dann ist ein Hard Constraint die nächste Wahl. Im Gegensatz zur Wahl der Repräsentation, findet die Auswahl durch einen Hard Constraint auf der Phänotypebene statt. Dazu existiert eine Funktion die für jeden Phänotyp testet, ob dieser den Constraint erfüllt. Solche Lösungen, die den Constraint nicht erfüllen werden disqualifiziert, während solche die ihn erfüllen, im Algorithmus genutzt werden dürfen. Da Constraints immer Constraints des Phänotyps sind, ist die Behandlung dieser auf Phä-

notyp ebene einfacher, als der Ausschluss invalider Phänotypen durch Änderung des Genotyps. Typischerweise werden bei der Nutzung von Hard Constraints an allen Stellen im Algorithmus an denen neue Lösungen generiert werden, sprich Zufallsinitialisierung, Mutation und Crossover, ungültige Lösungen herausgefiltert.

Hard Constraints haben denselben Vorteil wie der Ausschluss durch die Wahl der Repräsentation, dass ungültige Lösungen niemals in den Algorithmus einfließen, und damit weder Rechenkapazitäten für nicht nutzbare Lösungen genutzt werden, noch solche Eigenschaften, durch die eine Lösung die Constraints verletzt, vom Algorithmus überhaupt in Erwägung gezogen werden. Im Gegensatz zu dieser Variante muss bei der Nutzung von Hard Constraints, für jedes generierte Individuum allerdings getestet werden, ob es den Constraint erfüllt. Dies nimmt Rechenzeit in Anspruch und schränkt die Formulierung der Constraint-Funktion durch die hohe Zahl an Individuen insoweit ein, als dass diese angemessen schnell berechenbar sein muss.

2.3.3 Soft Constraints

Soft Constraints disqualifizieren Lösungen, die die Constraints verletzen, nicht. Stattdessen wird beim Nichterfüllen von Constraints ein Strafwert auf die Kostenfunktion addiert, bzw. von einer Fitnessfunktion subtrahiert. Dass Lösungen, die die Constraints nicht erfüllen, nicht disqualifiziert werden, hat den Vorteil, dass viele Optimierungsmethoden iterativ zu (lokalen) Optima konvergieren¹, und diese auch Lösungen, die die Constraints nicht erfüllen, als Trittbretter zu Lösungen, die die Constraints erfüllen, nutzen können.

Soft Constraint eignen sich in Fällen, in denen zu Beginn keine Lösungen bekannt sind, die die Constraints erfüllen, und in denen explorativ nach Lösungen gesucht werden soll, die die Constraints erfüllen. Auch eignen sie sich für solche Probleme, in denen qualitative Unterschiede bezüglich der Stärke der Verletzung der Constraints zwischen unterschiedlichen Lösungen existieren können. So kann argumentiert werden, dass im Falle, dass ein Constraint die Einhaltung eines Schwellenwerts ist, eine Lösung, die diesen um 1 überschreitet, qualitativ besser ist als eine, die diesen um 10 überschreitet.

Eine der größten Gefahren bei Soft Constraints ist, dass diese typischerweise als Kostenfunktionen formuliert werden, deren Wert mit der eigentlichen Zielfunktion der Optimierung kombiniert wird. Eine solche Kombination enthält immer eine Gewichtung für alle Teilfunktionen, aus denen sie besteht. Diese Gewichtung kontrolliert wie stark die einzelnen Funktionen in die Berechnung der Fitness einfließen. Oft entsteht die Einführung eines Soft Constraints ein weiteres Problem: Welche Gewichtung führt zu den besten beziehungsweise gewollten Ergebnissen? Unter welcher Gewichtung der Algorithmus die qualitativ besten Ergebnisse liefert ist typischerweise nicht klar, bevor dieser mit dieser Gewichtung ausgeführt wird. Wird eine suboptimale Gewichtung gewählt kann dies dazu führen, dass Constraints über oder unter priorisiert werden, was wiederum zur Folge haben kann, dass entweder Constraints oder das Primärziel ignoriert werden.

2.3.4 Vergleich

Welcher der drei Ansätze am besten für ein Problem ist, hängt von dessen Formulierung ab. Grundsätzlich sollte der Ausschluss durch Modellierung Hard Constraints und diese wiederum Soft Constraints vorgezogen werden. Wenn der Ausschluss durch die Modellierung allerdings mit hohem Aufwand verbunden ist, wie es in komplexen Domänen häufig der Fall ist, sollten stattdessen Hard Constraints in Erwägung gezogen werden. Wenn die Vermutung besteht, dass zum Erreichen ausreichend guter Lösungen Trittbrettlösungen,

¹Auch ein divergentes Optimierungsverfahren wie MAP-Elites konvergiert zu Optima, es wird nur sicher gestellt, das zu einer Vielzahl lokaler Optima konvergiert wird

Variante	Vorteile	Nachteile
Repräsentation		
	<ul style="list-style-type: none"> • Behandlung des Constraints im Algorithmus nicht mehr nötig • Keine zusätzlichen Berechnungen notwendig 	<ul style="list-style-type: none"> • Keine Trittbretter • Formulierung einer passenden Repräsentation in komplexen Domänen häufig aufwendig
Hard Constraint		
	<ul style="list-style-type: none"> • Ergebnis kann keine invaliden Lösungen enthalten • Formulierung des Constraints auf Phänotypeebene leichter 	<ul style="list-style-type: none"> • Keine Trittbretter
Soft Constraint		
	<ul style="list-style-type: none"> • Nutzung von Trittbrettern wird ermöglicht 	<ul style="list-style-type: none"> • Ergebnis kann invalide Lösungen enthalten

Tabelle 1: Vor-/Nachteile der verschiedenen Möglichkeiten einen Constraint zu modellieren

die die Constraints nicht erfüllen, benötigt werden sollten stattdessen Soft Constraints genutzt werden.

3 Methode

In diesem Kapitel werden die in Abschnitt 2 besprochenen Techniken auf zwei Domänen angewandt. Die erste dieser Domänen ist die Generierung von verformten Radkästen eines Velomobils, unter Beachtung eines Constraints. Die zweite Domäne ist die explorative Untersuchung verschiedener Bauteilformen an der Unterseite eines E-Rollers. In beiden Domänen ist die Verformung dreidimensionaler Bauteile und die darauffolgende Bewertung des Effektes auf die Aerodynamik, den diese Verformung hat, Untersuchungsziel. Auf die domänenübergreifende Teile der Repräsentation wird in Abschnitt 3.1 eingegangen. Darauffolgend wird in Abschnitt 3.2 und Abschnitt 3.3 auf die Spezifika, wie Constraints und Featurewahl, der Radkasten- beziehungsweise E-Roller-Domäne eingegangen.

3.1 Repräsentation der Domänen

In beiden Domänen geht es um die Deformation von dreidimensionalen Bauteilen und die Bewertung der aerodynamischen Effekte solcher Deformationen. D. h. es wird eine robuste Methode benötigt, um dreidimensionale Bauteile zu deformieren und solche Deformationen auf wenige Parameter reduzieren zu können. Eine Methode, die diese Rahmenbedingungen erfüllt ist die Freiformdeformation (Sederberg und Parry (1986)) auf die in Abschnitt 3.1.1 genauer eingegangen wird, die Deformationen von komplexen geometrischen Objekten anhand Verschiebungen von Rahmenpunkten ausdrücken kann. Für aerodynamische Simulationen wird in beiden Domänen OpenFoam verwendet, ein Open-Source-Programm welches unter anderem fähig ist Windkanalsimulationen durchzuführen. Auf die genaueren Eigenschaften von OpenFoam und den Aufbau einer passenden Simulationsumgebung wird in Abschnitt 3.1.2 eingegangen.

3.1.1 Freiformdeformation

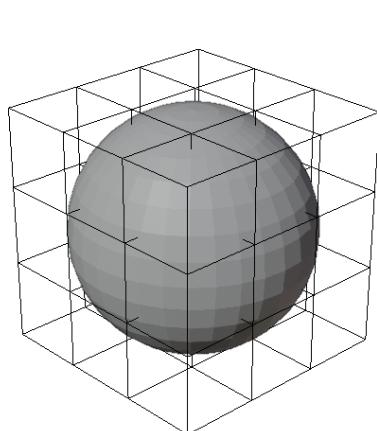


Abb. 5: 3D-Mesh einer Kugel in 4x4x4 FFD-Box

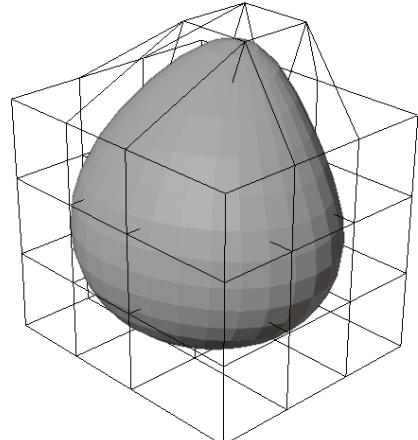


Abb. 6: Deformiertes 3D-Mesh erzeugt durch Bewegung der Deformationspunkte

Freiformdeformation ist ein in Sederberg und Parry (1986) beschriebenes Verfahren zu Deformation von geometrischen Objekten. Mithilfe der Freiformdeformation können beliebige dreidimensionale Meshes robust deformiert werden. Dazu wird eine Box aus Kontrollpunkten erzeugt, die manipuliert werden können, um das Mesh zu deformieren. In Abb. 6 ist das 3D-Mesh einer Kugel umgeben von einer FFD-Box aus vier Punkten in jeder Dimen-

sion. Damit ergeben sich 64 Deformationspunkte. Jeder dieser Kontrollpunkte kann dann anhand von Verformungsparametern verschoben werden, wodurch für alle Punkte innerhalb der FFD-Box eine Verschiebung berechnet wird, die von der relativen Position des Punktes zum Kontrollpunkt abhängt. Eine deformierte Variante der Kugel ist in Abb. 6 abgebildet. Dort wurden alle Punkte der rechten Ebene der FFD-Box nach außen bewegt und die vier inneren Punkte auf der Oberseite der FFD-Box nach oben und hinten bewegt. Die Knoten der Kugel werden durch die Freiformdeformation entsprechend bewegt, wodurch das deformiert Mesh entsteht. So können beliebige dreidimensionale Meshes robust anhand einiger Punkte deformiert werden. Auch können Deformationen durch relativ wenige Deformationsparameter definiert werden. Im Beispiel läge die Anzahl an Freiheitsgraden bei $64 * 3 = 192$. Durch eine weitere Auswahl an welchen Punkten Deformationen in welche Richtungen² stattfinden dürfen, kann diese Zahl noch weiter reduziert werden. Dadurch eignet sich das Verfahren zur Anwendung in evolutionären Algorithmen, in denen die Anzahl der möglichen Freiheitsgrade, was der Länge des Genoms entspricht, nach oben weich begrenzt ist. Dadurch das nur die Punkte des Meshes ihre Position ändern, hat das Verfahren, den Vorteil das die Dreiecke erhalten bleiben und somit beispielsweise das Entstehen von Löchern bei korrekter Anwendung praktisch ausgeschlossen ist.

3.1.2 OpenFOAM

OpenFoam (OpenCFD) ist ein Open-Source-Programm zur Durchführung von Fluideodynamiksimulationen. Berechnungen in OpenFOAM sind in sogenannten "Cases" organisiert, auf die nacheinander OpenFoam-Funktionen angewandt werden. Diese Cases bestehen aus einer Basisstruktur, die Initiale Startparameter, sowie Dateien zur Steuerung von OpenFoam enthalten. Diese Basis enthält typischerweise Skripts um die gesamte Kette aus verschiedenen OpenFoam Funktionen, die zur Durchführung einer Simulation notwendig sind, auszuführen, sowie ein Skript, welches den Case wieder in den Startzustand versetzt. Außerdem enthält diese Basis das Verzeichnis system, das C++-Dictionaries enthält, mit denen die einzelnen OpenFoam Funktionen parametrisiert und gesteuert werden. Für jede der genutzte OpenFoam Funktion existiert dort eine Datei, in der beschrieben wird was die entsprechende Funktion wie tun soll. Zuletzt befindet sich in diesem Ordner auch noch der Initiale Startzustand, mit dem die Simulation beginnt.

OpenFoam bietet native Unterstützung von Parallelität über MPI (OpenMPI). Neben den Funktionen um die korrekte Aufteilung eines Cases auf mehrere Prozessoren aufzuteilen und am Ende wieder zu rekonstruieren sind zwei Funktionen und deren dazugehörige Dateien hervorzuheben.

Diese beiden wichtigsten genutzten OpenFoam-Funktionen sind snappyHexMesh und simpleFoam. Mit snappyHexMesh, dessen Parametrisierung in **snappyHexMeshDict** zu finden ist, wird aus dem STL das für die Fluideodynamiksimulation benötigte Mesh generiert. Über snappyHexMesh kann die Auflösung des in der Fluideodynamiksimulation genutzten Meshes kontrolliert werden. Die Parametrisierung von snappyHexMesh ist deshalb besonders wichtig, da die Auflösung des Meshes die Simulation maßgeblich beeinflusst. SnappyHexMesh ist ein auf Octrees (Meagher (1982)) basierender Verfeinerungsalgorithmus. Die Parametrisierung von snappyHexMesh kontrolliert welche Bereiche stärker verfeinert werden. In diesen Bereichen wird der Baum tiefer und es werden mehr Zellen generiert. In der daraufliegenden Simulation bilden Zellen die atomare Einheit, wenn in bestimmten Bereichen nicht weit genug verfeinert wurde, besteht in der Simulation überhaupt nicht die Möglichkeit der Realität entsprechende Messwerte in dieser Zelle zu berechnen. Mit wachsender Anzahl der Zellen wächst allerdings auch die Zeit die für Verfeinerung und Simulation benötigt wird. Die korrekte Auswahl in welchen Bereichen es nötig ist für eine korrekte

²Deformation in x, y, z finden jeweils separat statt

Simulation stärker zu verfeinern und in welchen Bereichen eine weitere Verfeinerung nur weitere Berechnungskomplexität ohne Informationsgewinn hinzufügt ist sehr wichtig.

Die zweite dieser Funktionen ist die eigentliche Flüssigkeitsdynamiksimulation. Abhängig von den benötigten Simulationsparametern stellt OpenFoam einige unterschiedliche Solver bereit. Da die Strömungsgeschwindigkeiten der Simulation weit unter $0,3Ma$ bleiben kann die Strömung als inkompressibel behandelt werden (Anderson (2017)). Es existieren mehrere OpenFoam Solver für inkompressible Strömungen aus denen simpleFoam ausgewählt wurde. simpleFoam implementiert den SIMPLE (Semi-Implicit Method for Pressure Linked Equations) Algorithmus, der in Caretto et al. (1973) beschrieben wurde. Für die hier benötigten Werte reicht simpleFoam vollkommen aus. Zudem werden die Koeffizienten der auf das Objekt wirkenden Kräfte berechnet, welche für die Berechnung der Fitness benötigt werden.

3.1.3 Parametrisierung des Gaußprozesses

Die Parametrisierung des Gaußprozesses besteht aus drei Schritten. Zuerst werden die Funktionen für Kovarianz, Mittelwert und Wahrscheinlichkeit ausgewählt. Daraufhin müssen die Hyperparameter dieser drei Funktionen gesetzt werden. Da das Setzen der richtigen Hyperparameter schwer ist, können diese daraufhin, nachdem Datenpunkte vorhanden sind, so optimiert werden, dass die Daten am besten beschrieben werden.

Die erste Funktion, die ausgewählt wurde, ist die Kovarianzfunktion. Es sind einige Eigenschaften der zugrundeliegenden Funktion anzunehmen. Durch die Nutzung der Freiformdeformation zur Repräsentation der Domäne ist anzunehmen, dass kleine Änderungen des Genotyps mit kleinen Änderungen des Phänotyps korrespondieren. Auch ist anzunehmen, dass ähnliche Phänotypen ähnliche aerodynamische Eigenschaften haben. Es ist anzunehmen, dass die Kovarianz zwischen einem Punkt und seiner Umgebung in alle Richtungen abnimmt und es ist nicht anzunehmen, dass die Luftwiderstandswerte beispielsweise periodisch sind. Die isotropische Squared exponential Konvarianz Funktion

$$k(x, x') = \sigma_f * \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$

ist eine solche Funktion, die gaußsche Abnahme der Kovarianz in alle Richtungen annimmt. Diese besitzt die Hyperparameter σ_f , welcher die Standardabweichung des Signals bestimmt, und ℓ , wodurch das Längenmaß bestimmt wird. Die Squared exponential Funktion wird manchmal auch als Radial basis function (RBF) bezeichnet und stellt eine der meist genutzten Kernelfunktionen dar. Da sie unendlich differenzierbar ist, werden sehr glatte Funktionen generiert (Duvenaud (2014)).

Isotropisch bedeutet in diesem Fall, dass das Längenmaß ℓ in allen Dimensionen gleich ist. Da alle Dimensionen auf das Intervall $[0; 1]$ normiert sind, kann das gleiche Längenmaß für alle Dimensionen genutzt werden. Die Nutzung eines Längenmaßes vereinfacht die Parametrisierung, häufig kann es trotzdem der Fall sein, dass bestimmte Dimensionen wichtiger für die Korrelation zwischen zwei Punkten sind, eine Möglichkeit damit umzugehen ist das sogenannte Automatic relevance determination (ARD), wodurch jede Dimension ein unabhängiges Längenmaß erhält.

Die zweite Funktion die ausgewählt werden muss ist die Mittelwertsfunktion. Da unser Wissen über den Problemraum sich hauptsächlich auf Hypothese und Schätzungen beschränkt wurde hier die einfachste der Mittelwertsfunktionen gewählt. Der konstante Mittelwert

$$m(x) = c$$

besitzt nur einen Hyperparameter c , der festlegt auf welcher Konstanten der Mittelwert liegt.

Als letztes muss noch eine Entscheidung bezüglich der Wahrscheinlichkeitsfunktion getroffen werden. Diese steuert die Varianz des Gaußprozesses. Da es keinen Grund gibt davon auszugehen, dass diese nicht normalverteilt ist, wurde hier die Normalverteilung mit

$$\mathcal{N}(y_i | f_i, \sigma_n^2) = \frac{1}{\sqrt{2\pi * \sigma_n^2}} * \exp\left(-\frac{(y_i - f_i)^2}{2\sigma_n^2}\right)$$

gewählt. Auch diese Funktion hat nur einen Hyperparameter σ_n , der steuert wie stark das Rauschen auf den Daten ist. Bei einem stärkeren Rauschen der Daten wird ein größeres σ_n benötigt um nahe Datenpunkte mit sehr verschiedenen Werten beschreiben zu können. Die initiale Parametrisierung der Hyperparameter erfolgte gemäß Tabelle 2.

ℓ	0.5
σ_f	1
c	0
σ_n	1

Tabelle 2: Initiale Hyperparameter des Gaußprozesses

Da die Hyperparameter im Laufe des Algorithmus auf Basis realer Funktionsevaluierungen optimiert werden, sind die initialen Parameter weniger wichtig. Ein Längenmaß von 0,5 bedeutet hier, dass Punkte eine Auswirkung bis zu einer Entfernung von 0,5 im Genotypraum haben der in jeder Dimension auf das Intervall [0; 1] beschränkt ist. Da die Ermittlung des tatsächlichen Mittelwerts durch Openfoam-Simulationen möglich ist, wird ein Mittelwert von 0 gewählt, der nach der Auswertung der Initialsamples auf den Mittelwert dieser angepasst wird. Da die Spanne an Luftwiderstandswerten im vor herein nicht bekannt ist, wird $\sigma_n = 1$ gewählt. Wie bereits erwähnt werden diese Parameter durch die log-marginal-Likelihood-Methode während des Algorithmus angepasst.

3.2 Radkästen des Velomobils

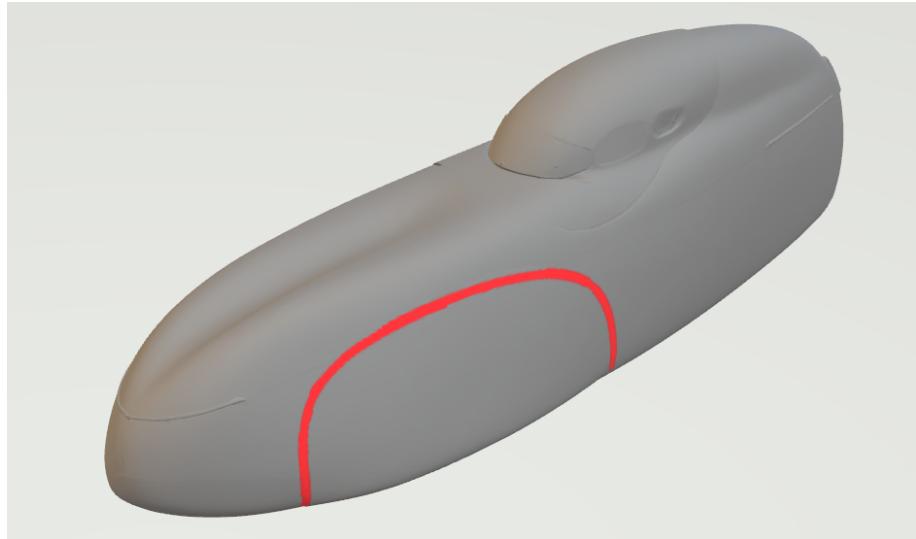


Abb. 7: Radkasten des Velomobils

Die erste Problemdomäne stellt die Optimierung der Radkästen eines Velomobils dar. Da die momentanen Radkästen den Lenkausschlag des Velomobils erheblich einschränken, besteht das Ziel hierbei Radkästen zu generieren, die den maximalen, oder zumindest einen weiteren Lenkausschlag als den momentan möglichen, ermöglichen und dabei trotzdem

gute aerodynamische Eigenschaften aufweisen. Es ist zu erwarten, dass eine zusätzliche Fokussierung auf ein zweites Ziel, mit einer Qualitätsabnahme des ersten Ziels verbunden ist. Das Ziel besteht also darin Lösungen zu entwickeln, welche durch minimalen Fitnessverlust einen maximalen Constraintgewinn möglich machen.

Da das Velomobil symmetrisch gebaut ist, wurde angenommen, dass die Verformungen des linken und rechten Radkastens symmetrisch sind. Im Folgenden wird bei allen Beschreibungen von Deformationen vom rechten Radkasten ausgegangen. Die Deformationspunkte zur Deformation des linken Radkastens ergeben sich durch eine Invertierung der y-Koordinate. Außerdem finden alle Deformationen in y-Richtung für den linken Radkasten negativ statt. Dies führt dazu, dass der linke Radkasten symmetrisch zum rechten verformt wird. Dadurch kann die Anzahl an benötigten Freiheitsgraden halbiert werden. Auch die Berechnungen des Constraints und der Kategorisierung können dadurch nur für den rechten Radkasten durchgeführt werden, was auch dafür den benötigten Aufwand halbiert.

Um möglichst glatte Übergänge zwischen Radkästen und Velomobil herstellen zu können wurden die Radkästen nicht separat verformt und daraufhin mit dem Rest des Velomobils kombiniert, sondern es wurden mit FFD-Boxen um die Radkästen das komplette Modell verformt. Dies ist zwar eine leichte Abweichung von der Aufgabenstellung, allerdings sind scharfe Ränder aerodynamisch suboptimal und daher zu vermeiden, und genau das wird dadurch erreicht.

3.2.1 OpenFOAM

Es wird wie bereits in Abschnitt 3.1.2 beschrieben snappyHexMesh zur Konstruktion der Volumenmeshes genutzt. Zur Verfeinerung wurde der "distance" Modus genutzt, bei dem Zellen abhängig von ihrer Distanz zu einer Oberfläche stärker verfeinert werden. Die Oberfläche zur Messung dieser Distanz ist in diesem Fall das Velomobil. Somit kann erreicht werden, dass Zellen nahe des Velomobils stark verfeinert werden, und weiter entfernte Zellen nicht zu stark verfeinert werden. Da aerodynamische Effekte primär am Velomobil stattfinden bot diese Einstellung eine Möglichkeit dafür angemessene Verfeinerungen herzustellen. Als Parametrisierung wurde (0.2 5) (0.5 4) (1.0 2) gewählt, d. h. alles mit einer Entfernung kleiner 0.2m zum Velomobil wird zu Level 5 verfeinert, alles mit einem Abstand bis 0.5m mit Level 4 und alles mit einem Abstand bis zu 1m mit Level 2.

Es wurden mehrere OpenFoam-Simulationen des Velomobils durchgeführt um zu testen, ob und wann die berechneten c_D -Werte ausreichend konvergierten. In Abb. 8 sind die berechneten c_D -Werte über den Simulationslauf dargestellt. Es ist erkennbar, dass diese zum Start der Simulation stark schwanken, sich aber relativ schnell auf den finalen Wert einpendeln. Ungefähr bei 50 Sekunden sind die c_D -Werte ausreichen konvergiert. Da die Konvergenz für unterschiedliche Formen unterschiedlich erfolgt wurden die c_D -Werte von 100 Sekunden bis 200 Sekunden gemittelt, um die Fitness zu berechnen. Offensichtlich reichen 200 simulierte Sekunden aber aus, um aussagekräftige Ergebnisse zu erlangen.

3.2.2 Constraint

Zur Erfüllung des Constraints wurden alle möglichen Radausschläge als Mesh modelliert. Dafür wurden aus einer Kugel entsprechend des maximalen Radausschlags nach rechts und links ein dreieckiges Stück ausgeschnitten. Zusätzlich wurde die innere Hälfte der Kugel entfernt, da dort keine Überschneidungen stattfinden, und durch deren Entfernen, Berechnungen effizienter sind. Das Mesh $FV_{radausschlag}$ ist in Abb. 9 alleine, in Abb. 10 zusammen mit dem restlichen Velomobil dargestellt. Es ist klar zu erkennen, dass dieses Volumen die unverformten Radkästen schneidet und diese den maximalen Lenkausschlag nicht ermöglichen.

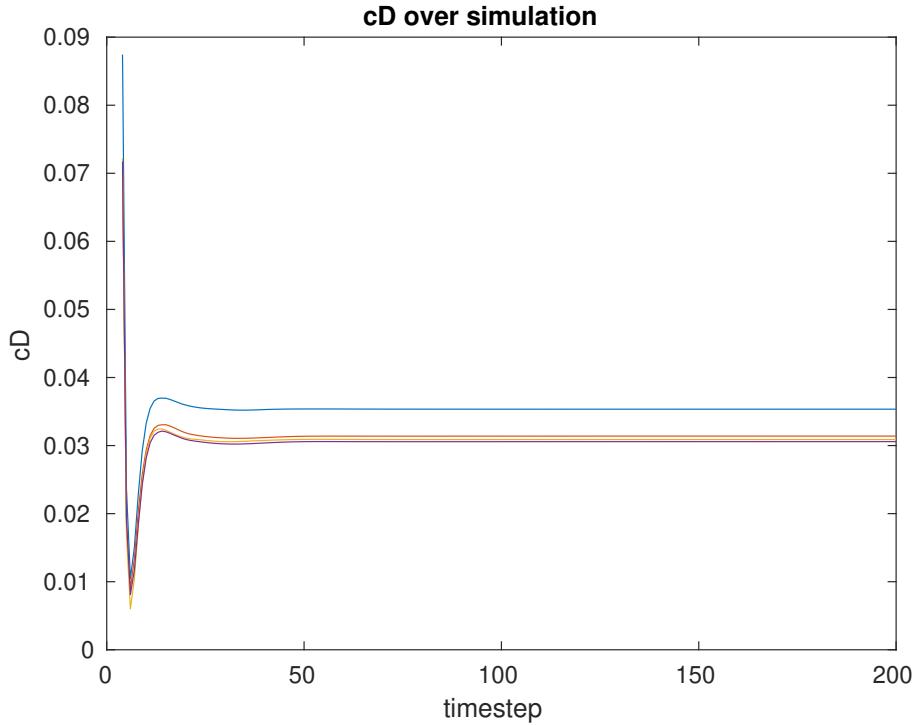


Abb. 8: Luftwiderstandswerte über Simulationen von vier zufällig verformten Radkästen



Abb. 9: Das modellierte Radausschlagsvolumen $FV_{radausschlag}$

Abb. 10: Radausschlagsvolumen $FV_{radausschlag}$ mit Velomobil

Da ein Ziel darin bestand, den Constraint nicht als binäres, erfüllt/nicht erfüllt Problem zu definieren, da zwischen zwei Lösungen, die den Constraint nicht erfüllen trotzdem qualitative Unterschiede bestehen können wie stark der Constraint verletzt wird, wird ein Soft Constraint zur Modellierung gewählt. Dazu wird die Meshdifferenz des Radausschlags und des verformten Radkastens berechnet um $FV_{diff} = FV_{radausschlag} - FV_{radkasten}$ zu ermitteln. Das Ergebnis einer solchen Meshdifferenz für den unverformten Radkasten ist in Abb. 12 abgebildet. Die Größe des Volumens $V(FV_{diff})$ wird zur Berechnung des Constraints herangezogen. Zwar besteht keine direkte Kausalität zwischen diesem Volumen und dem maximalen möglichen Radausschlag aber die Vermutung, dass ein geringeres Volumen dieser Differenz mit größerem möglichen Radausschlag korreliert ist, liegt nahe.

Da alle generierten Verformungen der Radkästen symmetrisch sind, wird der Constraint jeweils nur für den rechten Radkasten berechnet. Zur Berechnung der Differenz zwischen Radausschlagsvolumen und Radkasten wird die Bibliothek *gptoolbox* (Alec Jacobson and others (2018)) genutzt, zur Generierung von Tetraedermeshes zur Volumenberechnung *TetGen* (Si (2015)). Um die Meshdifferenz berechnen zu können wurde der rechte Radkasten zudem geschlossen. Da die Constraintberechnung in jeder Generation für jedes Kind erfolgen muss wurde der Radkasten außerdem vereinfacht um die Berechnung des Constraints ausreichend schnell durchführen zu können. Um den Strafwert p für ein Individuum

Position		Rotation	
x	926mm	ϕ	16,6°
y	343mm	θ	0°
z	205mm	ψ	0°
Radius des Rads	230mm		
Radausschlag	±24,37°		

Tabelle 3: Parameter Radausschlag. Ursprung des Koordinatensystems für Position auf Boden an der Spitze des Velomobils. Linkshändiges Koordinatensystem mit z Höhenachse und Velomobil in -x-Richtung ausgerichtet.



Abb. 11: Das vereinfachte, geschlossene Mesh des Radkastens $FV_{radkasten}$



Abb. 12: Das Differenzvolumen FV_{diff} für den unverformten Radkasten. Aus diesem wird der Straf- wert berechnet

zu berechnen wird $V(FV_{diff})$ des Individuums berechnet und in Relation zum Außenvolumen des unverformten Radkastens gesetzt ³. Dadurch kann der Constraint normiert werden und es kann eine einfache Einordnung in Bezug zum unverformten Radkasten geschehen. Erfüllt ein Individuum den Constraint gleich gut wie der unverformte Radkasten gilt $p = 1$. Ist ein Individuum besser gilt $0 \leq p < 1$, ist es schlechter gilt $p > 1$.

$$p = \frac{V(FV_{diff})}{V(FV_{diffUnverformt})}$$

Auch wenn die genutzten Bibliotheken zwar recht robust sind ist es trotzdem so, dass sie in ihrer Kombination aber immer noch für ca. 1 aus 10000 Lösungen abstürzen. Die Häufigkeit dieser Abstürze wurde als gering genug eingestuft, als dass eine Untersuchung des Grunds nicht nötig ist. Stattdessen werden solche Lösungen im Algorithmus als unendlich schlecht bewertet und damit ausgeschlossen.

Da sich der Constraint durch die Vereinfachung des Radkastens ausreichend schnell berechnen ließ, dass eine direkte Berechnung dessen für jedes Individuum zeitlich möglich war, wurde dies getan. Das Trainieren eines zweiten Surrogatmodells zur Modellierung des Constraints hätte einen nicht unerheblichen Komplexitätszuwachs bedeutet, der an dieser Stelle nicht nötig war. Also wurde der Strafwert mit Gewicht $w_{constraint} \geq 0$ gewichtet in die Fitnessfunktion integriert und als $p * w_{constraint}$ auf die Fitness jedes Individuums addiert ⁴. Allerdings wurde der Strafwert zusätzlich noch in die Akquisefunktion, mit der innerhalb von SAIL die Akquisekarte zur Auswahl neuer Individuen zur Auswertung stattfindet auf die gleiche Weise integriert. Das hat den Grund, dass das Gaußprozessmodell in solchen Bereichen bevorzugt Exploitation durchführen soll, die sowohl bezüglich der Aerodynamik, als auch bezüglich des Constraints eine gewisse Optimalität aufweisen, da genau in diesen Bereichen, in der finalen Auswertung von MAP-Elites solche Lösungen zu erwarten sind, die bezüglich der Kombination dieser beiden Kriterien optimal sind.

³Für den unverformten Radkasten beträgt das Differenzvolumen ca. $1,6L$

⁴Die vorliegende SAIL-Implementation minimiert, Addition positiver Werte ist folglich Strafe

3.2.3 Wahl der Features

Als erste Kategorie wurde die Breite des Velomobils gewählt. Diese Kategorie wurde deshalb gewählt, da hier sehr klare Hypothesen aufgestellt werden können, wie die Breite des Velomobils mit der Aerodynamik und dem Constraint interagiert. Die erste Hypothese, die aufgestellt werden kann ist, dass breiteren Velomobilen die Erfüllung des Constraints leichter fällt. Es wäre also zu erwarten, dass breitere Velomobile den Constraint tendenziell besser erfüllen, als schmalere.

H1: Breitere Velomobile sind positiv mit Erfüllung des Constraints korreliert.

Auch kann die Hypothese aufgestellt werden, dass die Breite negativ mit dem Luftwiderstandsbeiwert korreliert ist.

H2: Breitere Velomobile sind negativ mit Luftwiderstandsbeiwert korreliert.

Insgesamt wäre also zu vermuten, dass entlang dieser Achse der Luftwiderstandsbeiwert zunimmt, während der Strafwert des Constraints abnimmt.

Als zweite Kategorie wurde die x-Koordinate des breitesten Punkts gewählt. Da das Velomobil in negative x-Richtung ausgerichtet ist, bedeuten kleinere Werte hier, dass der Punkt weiter vorne liegt, größere, dass er weiter hinten liegt. Zu dieser Kategorie lassen sich keine so direkten Hypothesen aufstellen wie zur ersten. Die Frage, ob sich hier klare Tendenzen bezüglich der aerodynamischen Eigenschaften und/oder des Constraints aufzeigen ist auch ein Ziel der Untersuchung.

Da auch die Kategorisierung so häufig stattfindet, wird auch diese mit dem für die Berechnung des Constraints vereinfachten Modell des Radkastens durchgeführt. Dies stellt keinen signifikanten Verlust an Genauigkeit in der Kategorisierung dar, da sich die am feinsten gemessenen Teile des Radkastens an dessen Rändern befinden. Gerade die sind aber die Teile des Radkastens, die am unwahrscheinlichsten den breitesten Punkt darstellen. Auch führen minimale Genauigkeitsfehler in der Kategorisierung nicht dazu, dass die von MAP-Elites geforderte Lokalität verletzt wird. Der Geschwindigkeitsgewinn bei der Berechnung hingegen macht es möglich solche Kategorien, für die eine Verformung des Radkastens notwendig ist, zu wählen, was in Relation einen wesentlich größeren Gewinn darstellt.

3.3 E-Roller

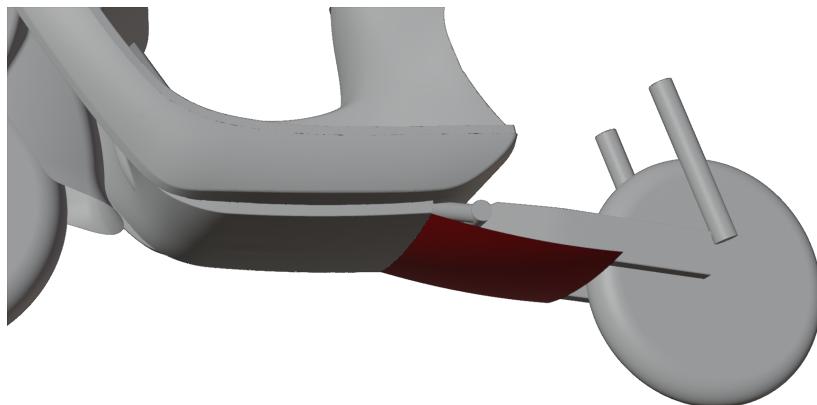


Abb. 13: E-Roller (weiß) und das zu deformierende Bauteil (rot)

Die zweite Problemdomäne ist die explorative Untersuchung eines Bauteils an der Unterseite eines E-Rollers. Hier besteht die Frage ob nicht-triviale Bauteile aerodynamische

Vorteile bieten und welche Eigenschaften solche Bauteile aufweisen. In Abb. 13 ist der E-Roller von unten mit dem Bauteil rot markiert dargestellt. Es ist ersichtlich, dass ein vereinfachtes Modell des E-Rollers genutzt wird. Um den Rechenaufwand in der Fluideodynamiksimulation zu reduzieren wurden Teile des E-Rollers die mit großer Wahrscheinlichkeit keinen Einfluss auf die Strömung um das Bauteil haben, wie beispielsweise der Lenkerbereich entfernt. Auch der Detailgrad am sonstigen E-Roller wurde reduziert und mit dem vereinfachten Modell gerechnet.

3.3.1 OpenFOAM

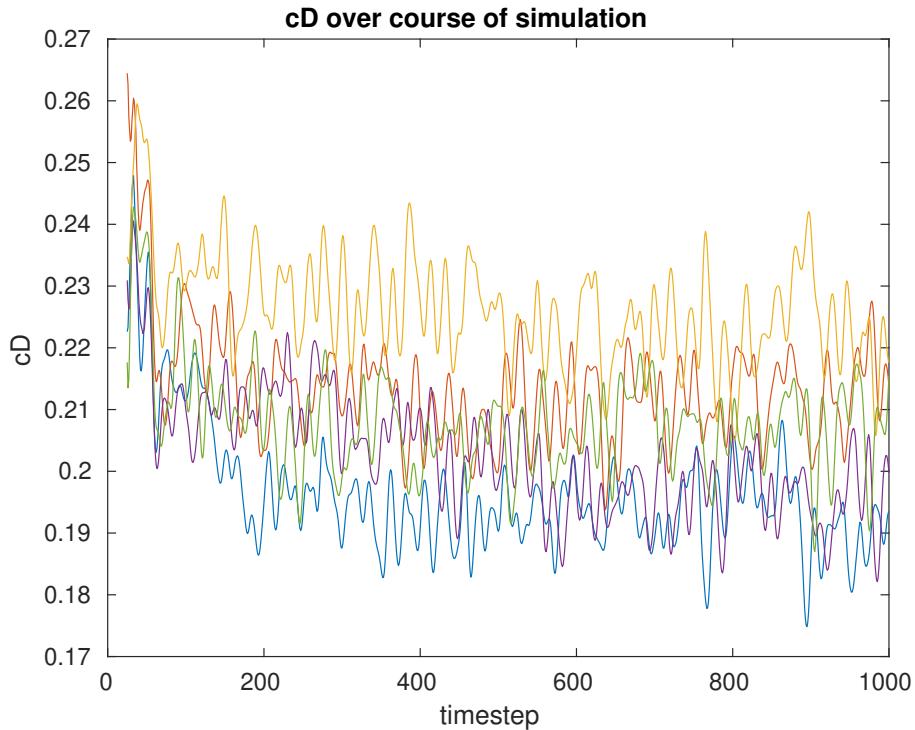


Abb. 14: Berechneter c_D -Wert über den Lauf von Simulationen von fünf zufällig deformierten Bauteilen

Auch der E-Roller wird zuerst mit snappyHexMesh gemesht, worauf die Fluideodynamiksimulation mit simpleFoam folgt. Es wurde wieder der "distance"Modus von snappyHexMesh genutzt um abhängig von der Distanz zum E-Roller Zellen mehr oder weniger zu verfeinern. In den Openfoam Simulationen stellte sich allerdings heraus, dass die c_D -Werte, wie in Abb. 14 zu sehen ist, nicht ausreichend konvergieren. Selbst nach 1000 simulierten Sekunden ist eine Konvergenz nicht absehbar, im Gegenteil ist vom hundertsten bis zum tausendsten Zeitschritt keine Verbesserung bezüglich der Konvergenz festzustellen. Auch entsteht keine regelmäßige Oszillation aus welcher der Mittelwert einfacher ermittelt werden kann. Stattdessen schwingt der c_D -Wert eher unregelmäßig, was die Berechnung eines Fitnesswerts erheblich erschwert. Dass die Simulation nicht konvergiert kann verschiedene Gründe, wie beispielsweise die falsche Parametrisierung der Simulation oder die falsche Auswahl der Simulationsfunktion, haben. Die Untersuchung und Anpassung von Openfoam zur Behebung dieses Problems sprengt allerdings den Rahmen dieser Arbeit. Zur Fitnessberechnung wurden die Simulationen daher mit 1000 Zeitschritten durchgeführt und der c_D -Wert über die letzten 900 Zeitschritte gemittelt. Aussagen über die Ergebnisse in der E-Rollerdomäne müssen daher unter Vorbehalt getroffen werden, da die Möglichkeit besteht, dass die Ergebnisse der Openfoam-Simulationen inkorrekt oder zumindest ungenau sind.

3.3.2 Constraint

In der E-Roller Domäne existierten einige Constraints. Anders als beim Velomobil konnten diese allerdings durch die Wahl der Repräsentation bereits ausgeschlossen werden. Zuerst muss das Bauteil einen gewissen Abstand zur Fahrbahn haben, um mit dieser nicht zu kollidieren. Bauteile, die dies nicht erfüllen, können durch die Wahl der maximalen Deformationstärke ausgeschlossen werden. Außerdem muss das Bauteil am restlichen E-Roller befestigt werden. Dafür darf die vordere Kante des Bauteils nicht deformiert werden. Dies kann vermieden werden, indem die Dimensionen der FFD-Box so gewählt werden, dass die vordere Kante nicht in dieser liegt. Dadurch wird ausgeschlossen, dass diese Kante deformiert wird und der korrekte Anschluss von Bauteil zum Rest des Rollers ist gewährleistet.

3.3.3 Wahl der Features

Die Wahl der Features in der E-Roller Domäne ist relativ ähnlich zu der Radkastendomäne. Statt der Breite ist hier die Höhe beziehungsweise Tiefe des Bauteils interessant. Als erstes Feature wurde daher die z-Koordinate des tiefsten Punkts des Bauteils gewählt. Im Gegensatz zum Velomobil kann keine klare Hypothese aufgestellt werden, ob die Höhe des Bauteils mit dessen Effekt auf den Luftwiderstand des E-Rollers korreliert ist. Zwar ist zu erwarten, dass der Luftwiderstand des Bauteils durch eine höhere Angriffsfläche steigt, es ist aber möglich, dass ein höheres Bauteil die Strömung im hinteren Bereich so weit verbessert, dass insgesamt eine Minderung des Luftwiderstands stattfindet.

Als zweites Feature wurde die x-Koordinate des tiefsten Punktes gewählt. Effektiv bedeutet das, dass unterschieden wird ob dieser weiter vorne oder hinten liegt. Auch hierzu ist keine klare Hypothese aufzustellen, ob und wenn ja welchen Effekt die Position des tiefsten Punkts auf die Aerodynamik hat.

3.3.4 Deformationspunkte

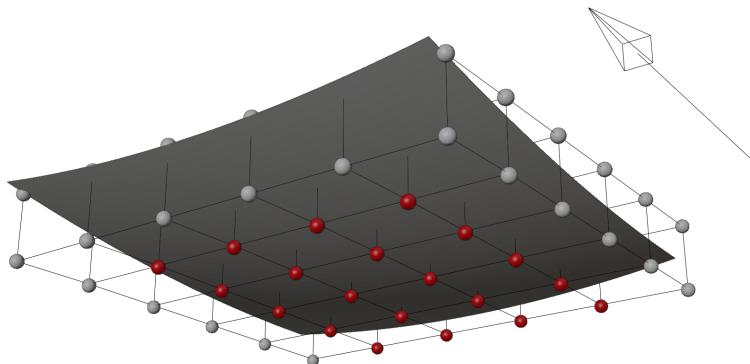


Abb. 15: Die FFD-Box und Deformationspunkte des E-Roller-Bauteils. Pfeil zeigt in Fahrtrichtung

Grundsätzlich interessant sind Deformationen des Bauteils nach unten. In Abb. 15 sind die Deformationspunkte dargestellt mit denen das Bauteil deformiert wird. Da die Vertikale die interessante Dimension ist, werden diese Punkte nur in y-Richtung deformiert. Dadurch kann eine höhere Anzahl an Deformationspunkten gewählt werden. Die Deformationspunkte wurden explizit so gewählt, dass asymmetrische Deformationen des Bauteils

möglich sind. Zwar ist anzunehmen, dass symmetrische Bauteile für den symmetrischen E-Roller optimaler sind, es ist allerdings ein Ziel diese Hypothese zu verifizieren.

H3: Symmetrische Bauteile werden vor asymmetrischen Bauteilen präferiert.

4 Auswertung

4.1 Radkasten

Im folgenden Kapitel werden auf Basis der in 3.2 eingeführten Methodik Experimente durchgeführt. Es wird so vorgegangen, dass zuerst ein eher einfaches Experiment durchgeführt wird, auf dessen Basis die grundsätzlichen Annahmen über das Problem der Radkastenoptimierung geprüft werden können. Darauf aufbauend werden aufwendigere und ausführlichere Experimente durchgeführt, um für die Realität nutzbare Ergebnisse zu erhalten.

4.1.1 Initiales Validierungsexperiment

Anzahl initialer Samples	100
Anzahl Samples	500
Anzahl neuer Samples pro Akquiseschleife	20
Anzahl Generationen Akquise-MAP-Elites	1024
Kinder pro Generation Akquise-MAP-Elites	32
Anzahl Generationen Ergebnis-MAP-Elites	2048
Kinder pro Generation Akquise-MAP-Elites	32
Auflösung der MAP-Elites Karte	25 * 25
Freiheitsgrade	6
Mittelwertgewichtung	1
Varianzgewichtung	2
Constraintgewichtung	1

Tabelle 4: Parametrisierung des ersten Experiments

Das Primärziel des ersten Experiments bestand darin zu verifizieren, dass der Algorithmus trotz Einbindung des Constraints korrekt funktioniert, d. h. auch mit Constraint vergleichbare Werte bezüglich Luftwiderstandskoeffizienten erreicht werden, und der Algorithmus trotz des Constraints korrekt lernt. Deshalb wurde die Parametrisierung vergleichsweise einfach gewählt, FFD-Deformationen fanden in zwei Punkten, in jeweils drei Richtungen statt. Auch wurde die Laufzeit so weit beschränkt, dass sinnvolle Aussagen darüber getroffen werden können, ob und wie gut der Algorithmus optimiert, für finale Ergebnisse hingegen wäre die Anzahl an Samples die ausgewertet werden sowie die Anzahl an Generationen für Akquise, und finaler Auswertung höher zu wählen. Um bewerten zu können, ob die Einbindung des Constraints einen Effekt auf den Algorithmus hat, und um diesen Effekt quantifizieren zu können wurde der Algorithmus unter den in Tabelle 4 beschriebenen Parametern sowohl mit Constraint als auch ohne Constraint durchgeführt.

Die Ergebnisse des ersten Experiments sehen vielversprechend aus in Abb. 17 ist ein Vergleich der Luftwiderstandsergebnisse ⁵ der erzeugten Lösungen abgebildet. Die erste Beobachtung die gemacht werden kann ist, dass die produzierten Werte aus der Variante mit Constraint in der gleichen Größenordnung liegen wie die aus der Variante ohne Constraint. Die Lösungen scheinen damit auf ersten Blick plausibel. Es ist zu erkennen, dass die Einbindung des Constraints zu einer Verschlechterung der Luftwiderstandswerte führt, eine Verschlechterung dieser durch eine Fokussierung auf zwei Ziele ist allerdings zu erwarten. Trotzdem muss betont werden, dass die Verschlechterung erstaunlich klein ist, der Median in der Version ohne Constraint liegt bei $-3,44$, während der Median in der

⁵Im Experiment wurde statt der direkten Luftwiderstandswerte, der natürliche Logarithmus dieser genutzt.

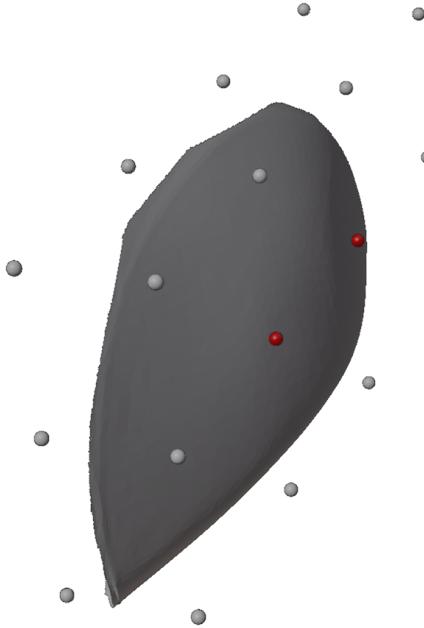


Abb. 16: Deformationspunkte des ersten Experiments

Version mit Constraint bei $-3,42$ liegt. Da die Fitness als $\ln(c_D)$ definiert ist, entspricht dies c_D -Werten von 0,0321 und 0,0327. Das entspricht einer relativen Verschlechterung des Luftwiderstands von 2%, was sehr klein ist.

Um eine solche Verschlechterung zu rechtfertigen, sollte die Qualität des Sekundärziels herangezogen werden. Dazu ist in Abb. 18 ein Vergleich bezüglich des Constraintvolumens dargestellt. Die erste Beobachtung hier ist, dass die Einbindung des Constraints zu einer stärkeren Beachtung dessen führt. Das ist offensichtlich das Ziel bei der Nutzung eines Constraints. Dass diese Erwartung erfüllt wird, zeigt, dass die Formulierung des Constraints in dieser Weise funktioniert. Auch ist hervorzuheben, dass die relative Verbesserung des Constraints wesentlich stärker ist als die relative Verschlechterung des Luftwiderstands. Eine Verschlechterung des Medians bezüglich des Luftwiderstands von 2% geht hier mit einer Verbesserung des Medians bezüglich des Constraints von 41,5% einher.

Interessant sind auch die Grenzen der Verteilungen, beide Verteilungen weisen eine untere Grenze von ca. 0,1, sprich 10% des Außenvolumens des nicht verformten Radkastens auf. Dies lässt sich dadurch erklären, dass das Radausschlagsvolumen einen Teil unterhalb des Radkastens hat, der durch eine seitliche Deformation des Radkastens nicht eliminiert werden kann. Die zweite interessante Grenze stellt die Obergrenze von 0,8 dar. Selbst die schlechteste Lösung in der Version ohne Constraint ist damit besser als der undeformierte Radkasten. Dies kann einerseits an zu eng gewählten Deformationsparametern liegen, die genutzte Konfiguration erlaubte allerdings Deformationen nach innen. Die plausiblere Hypothese ist, dass die Wahl der Breite des Radkastens als Feature bedeutet, dass eine Minimalbreite existiert, bei der nicht das Zentrum des Radkastens, sondern die Ränder den breitesten Punkt darstellen. Jede Deformation die das Zentrum stärker nach innen verformt, würde in die gleiche Zelle der Karte eingeordnet werden. Wenn weniger stark verformte Radkästen bezüglich deren Fitness optimaler sind, wird dies dazu führen, dass

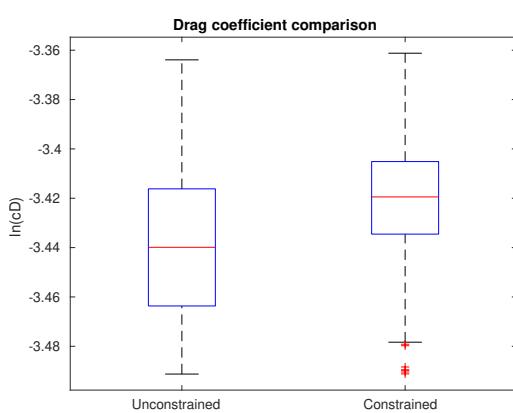


Abb. 17: Vergleich der Luftwiderstände der produzierten Lösungen

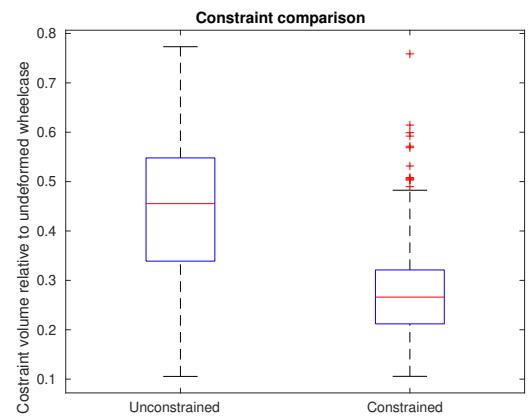


Abb. 18: Vergleich der Constrainterfüllung der produzierten Lösungen

Radkästen mit einer stärkeren Verformung nach innen keinen Einzug in die Karte finden können.

Aufgrund der Tatsache, dass SAIL ein divergenter Optimierungsalgorithmus ist haben die finalen Verteilungen nur eingeschränkte Aussagekraft. Besonders dadurch, dass das Feature der Breite des Velomobils mit der Schwierigkeit der Erfüllung des Constraints korreliert ist, müssen auch die produzierten Karten analysiert werden.

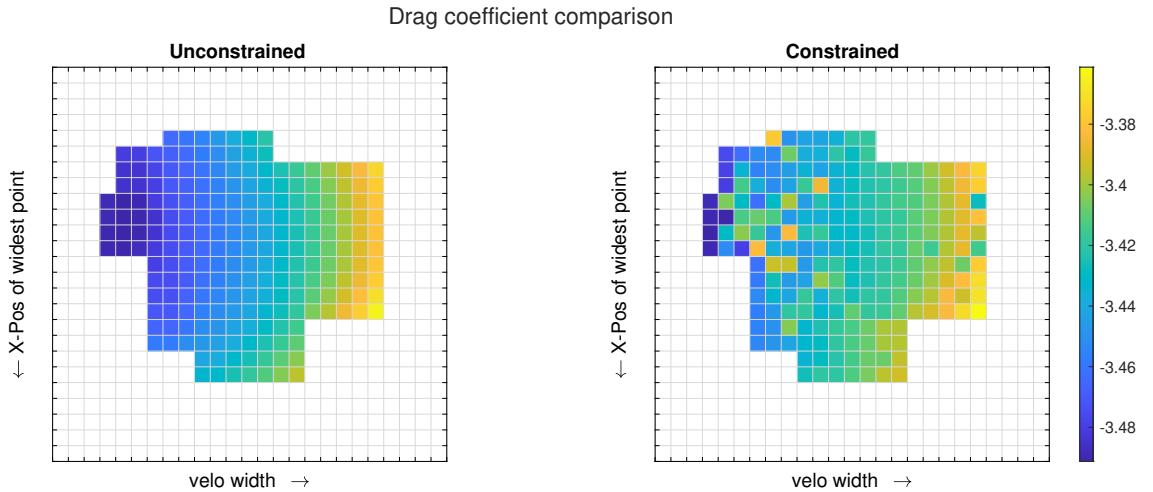


Abb. 19: Karte der Luftwiderstandswerte der Lösungen

In Abb. 19 sind die Karten von Luftwiderstandskoeffizienten der produzierten Ergebnisse in beiden Versionen aufgeführt. Die Karte ohne Constraint bestätigt die Hypothese, dass breitere Velomobile mit schlechterem Luftwiderstandskoeffizienten korreliert sind. Auch ist erkennbar, dass es ein mittleres Optimum bezüglich der x-Position dieses Punktes zu geben scheint, da Luftwiderstandswerte sowohl bei Abweichungen nach oben als auch bei Abweichungen nach unten schlechter werden. Dass die oben aufgestellte Hypothese bestätigt werden kann und die Karte einen graduellen Verlauf aufweist, sind Zeichen für die korrekte Funktionalität des Algorithmus. Im Gegensatz dazu ist die Karte der Variante mit Constraint wesentlich unregelmäßiger. Dies bedeutet allerdings nicht, dass der Algorithmus in dieser Variante nicht korrekt arbeitet, die Unregelmäßigkeit der Karte röhrt daher, dass die Fitness des Algorithmus aus einer Kombination von Luftwiderstandskoeffizienten und Constraint zusammensetzt. Auch bewegen sich die Luftwiderstandswerte, wie bereits anhand der Boxplots sichtbar war im Bereich. Besonders gute Lösungen bezüglich einer

dieser beiden Komponenten können fehlende Optimalität bezüglich der anderen damit ausgleichen. Trotzdem sind auch in dieser Karte die gleichen zugrundeliegenden Tendenzen zu erkennen. Breitere Velomobile sind auch hier schlechter bezüglich des Luftwiderstands und Optimalität sinkt nach unten und oben. Interessant ist, dass am rechten, sprich breiteren, Ende einige Lösungen gefunden wurden die sogar bessere Luftwiderstandswerte aufweisen als deren Gegenstücke in der Variante ohne Constraint. Und auch wenn auf der linken Seite einige Lösungen existieren, welche erheblich schlechtere Luftwiderstandswerte aufweisen wie ihre Gegenstücke, so existieren auch sehr viele mit vergleichbaren Werten. Um solche Verschlechterungen in Kontext setzen zu können müssen sich die Karten des Constraints angeschaut werden.

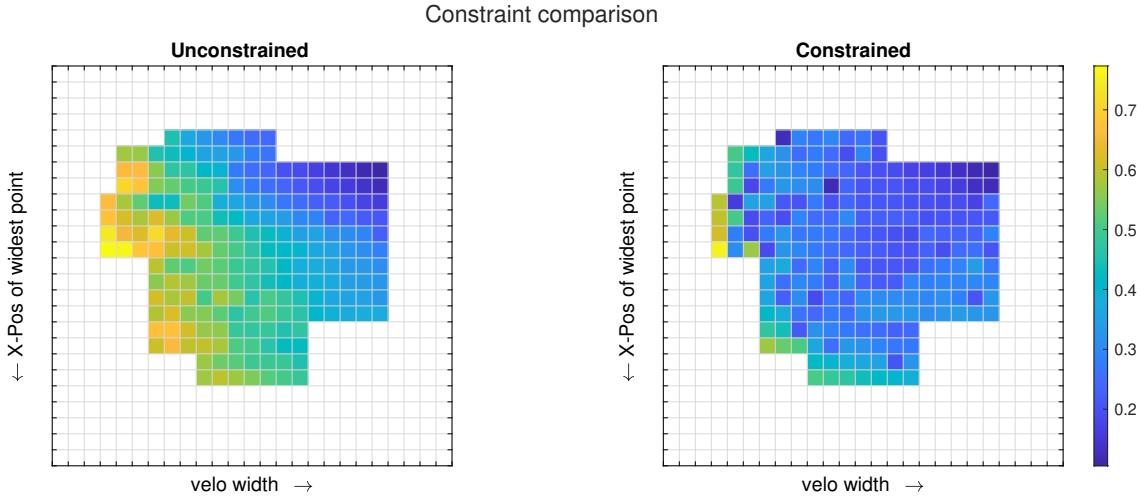


Abb. 20: Karte der Constraintwerte der Lösungen

In Abb. 19 sind die Karten der korrespondierenden Constraint-Werte für beide Varianten zu sehen. Erst einmal lässt sich auch hier die Hypothese bestätigen, dass breitere Velomobile den Constraint besser erfüllen. Interessant ist allerdings auch, dass ohne Einbindung des Constraints, Lösungen, deren breitesten Punkt weiter vorne ist den Constraint besser erfüllen.

Im Vergleich der beiden Karten ist noch viel stärker, als den Boxplots, ersichtlich welchen Effekt die Einbindung des Constraints hat. Es ist klar ersichtlich wie viel besser ein Großteil der Lösungen den Constraint erfüllen. Besonders stark ist dieser Effekt im Mittelfeld der Karte in denen sich Constraintwerte halbieren und teilweise noch stärker verringern. Zum rechten extremen Rand schwächt die Constraintverbesserung ab, da dortige Radkästen so breit sind, dass selbst die Variante ohne Constraint diesen leicht erfüllen kann. Auch ist hervorzuheben, dass sich Verbesserungen bis zum linken Rand der Individuen durchziehen. Selbst in der linken, sprich schmalsten Spalte, ist eine Verbesserung der Individuen erkennbar, und die nächste Spalte enthält bereits ein Individuum welches nahe der besten gefundenen Constraint-Werte angesiedelt ist.

In Abb. 21 ist die Verbesserung der Constraintwerte von der Version ohne Constraint zur Version mit Constraint dargestellt. Der bereits oben erwähnte Effekt, dass die größten Verbesserungen besonders in der linken Hälfte der Karte stattfinden ist hier noch klarer ersichtlich. Gerade diese Hälfte ist allerdings die, die bessere Luftwiderstandswerte aufweist und in der die Erfüllung des Constraints aufgrund der geringeren Breite, die Velomobile in diesen Zellen auszeichnet, wesentlich schwerer fällt. Zusammenfassend kann man sagen, dass die Einbindung des Constraints mit den im ersten Experiment vorliegenden Einschränkungen merkliche Auswirkungen auf das Ergebnis hat. Auch ist herauszustellen, dass eine starke Verbesserung des Constraints bei einer schwachen Verschlechterung des Luftwiderstands stattfindet.

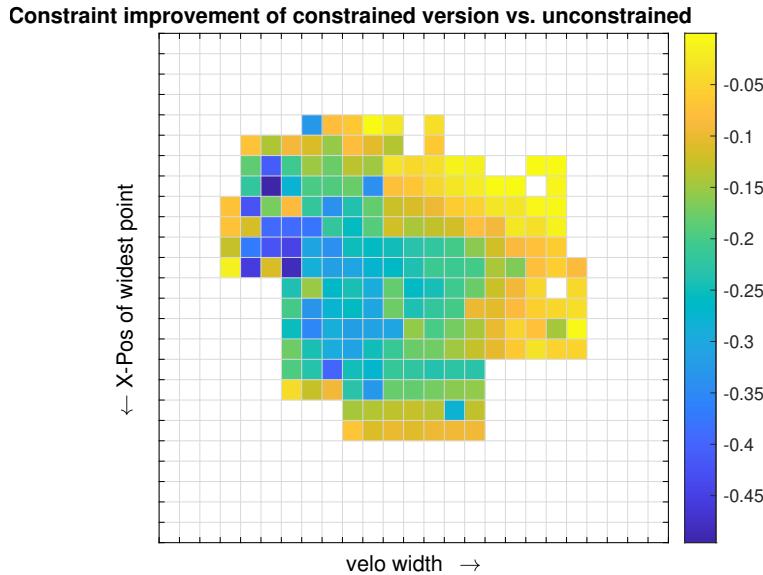


Abb. 21: Verbesserungen des Constraints zwischen Version ohne und mit Constraint

4.1.2 Erhöhung der Freiheitsgrade

Mit dem ersten Experiment konnte gezeigt werden, dass der Algorithmus sowohl Luftwiderstandskoeffizienten als auch Constraint optimiert, und dabei relativ geringe Kosten in Bezug zum erreichten Luftwiderstand entstehen. Das erste Experiment hatte allerdings einige Einschränkungen, bezüglich der Parametrisierung um es einfach zu halten. Die erste dieser Einschränkungen war die Reduzierung auf Deformation in 2 Punkten, sprich 6 Freiheitsgraden. Diese Reduzierung half dabei das Problem klein zu halten, hatte aber eine erhebliche Einschränkung des Lösungsraums zur Folge. Ziel des zweiten Experiments ist es die Anzahl an Freiheitsgraden erheblich zu erhöhen, damit die Zahl an möglichen Deformationen erheblich zu erhöhen, was den positiven Effekt der Einbindung eines Constraints hoffentlich verstärkt.

Die in Abb. 22 dargestellte Deformationspunkte wurden mit einigen Vorüberlegungen gewählt. Die Anzahl an horizontaler Reihen wurde von einer auf zwei erhöht um Deformationen , wie ein Auseinanderziehen oder Zusammendrücken in z-Richtung zu erlauben. Es liegt die Vermutung nahe, dass die im vorigen Experiment gewählten Deformationspunkte die möglichen Phänotypen einschränken, besonders in Anbetracht auf die x-Position des breitesten Punkts. Da diese eine der Features der MAP-Elites-Karte ist und die Karte im vorigen Experiment entlang dieser Dimension nicht vollständig ausgefüllt wurde, wurde die Anzahl der Deformationspunkte in x-Richtung von zwei auf drei erhöht, um mehr Deformationen zuzulassen, in denen der breiteste Punkt weiter vorne oder hinten liegt. Insgesamt ergeben sich dadurch 6 Deformationspunkte, welche in alle drei Richtungen bewegt werden können, wodurch sich 18 Freiheitsgrade ergeben.

Der erste Unterschied, der im Vergleich zum ersten Experiment auftritt, sind die stärkeren Unterschiede in Constraint und Luftwiderstand für.

In Abb. 25 und 26 ist klar zu erkennen, dass die Karte aus Lösungen weitaus besser gefüllt wird als beim ersten Experiment. Die bessere Füllung entlang der Vertikalen ist leicht zu erklären. Die Aufteilung in 6 Deformationspunkte führt dazu, dass die beiden vorderen weiter vorne liegen, als der vordere aus dem vorigen Experiment und die beiden hinteren entsprechend auch weiter hinten liegen. Deformationen mit diesen Deformationspunkten können also mehr Lösungen generieren, in denen der breiteste Punkt sehr weit vorne oder sehr weit hinten liegt. Interessant ist allerdings, dass die Karte auch entlang der Horizontalen besser gefüllt wird. An der Stärke der Deformationen in y-Richtung wurde

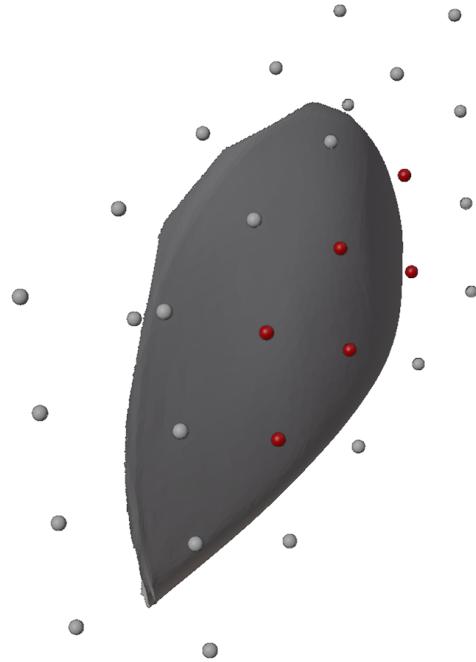


Abb. 22: Deformationspunkte des 2. Experiments

allerdings nichts geändert, es sollte also nicht möglich sein wesentlich breitere Velomobile zu generieren.

Die Karte der Constraints bestätigt auch hier, die im ersten Experiment bereits beobachtete Tendenz, dass der Constraint in breiteren Velomobilen und in solchen in denen der breiteste Punkt weiter vorne liegt einfacher zu erfüllen ist. Es ist allerdings herauszustellen, dass in der Variante ohne Constraint selbst die Lösungen in der oberen rechten Ecke eine weniger gute Constrainterfüllung aufweisen als die besten Lösungen in der Variante ohne Constraint des ersten Experiments, trotz der Tatsache, dass diese sowohl breiter sind und die breiteste Stelle weiter vorne liegt. Zwar ist dieser Effekt auch im Vergleich der Varianten ohne Constraint festzustellen, er fällt dort aber wesentlich schwächer aus. Dies deckt sich mit der Tatsache, dass eine Erhöhung der Freiheitsgrade und die damit verbundene Vergrößerung des Suchraumes schwerer macht ähnlich gute Lösungen bei gleicher Laufzeit zu finden. Dass der Effekt in der Variante ohne Constraint aber stärker ist bestätigt, dass die Schwierigkeit des zufälligen Erfüllens des Constraints ohne spezielle Behandlung stärker wächst, als die Schwierigkeit der Erfüllung unter Beachtung des Constraints.

Es fällt ein interessanter Unterschied zum ersten Experiment auf. Die Karten der Variante ohne Constraint und der mit Constraint sind im Gegensatz diesem nicht mehr gleich gefüllt. Stattdessen ist die Karte ohne Constraint besser gefüllt als die mit Constraint. Dies fällt vor allem an den Rändern der Karten auf. Das kann damit zusammenhängen, dass durch die Einführung des Constraints Exploration in Bereichen in denen der Constraint verletzt ist weniger stattfindet. Außerdem besteht die Vermutung, dass weite Teile des Problemraums den Constraint nicht erfüllen, sondern stark verletzen.

Zur definitiven Auswertung sollte allerdings sichergestellt sein, dass Akquise- und Ergebnis-MAP-Elites ausreichend konvergiert sind. Dass kann aus in diesem Experiment noch nicht definitiv festgestellt werden. Um eine definitive Aussage über Konvergenz treffen zu können sollte das Experiment unter ausführlicheren Bedingungen wiederholt werden.

Anzahl initialer Samples	100
Anzahl Samples	500
Anzahl neuer Samples pro Akquireschleife	20
Anzahl Generationen Akquise-MAP-Elites	1024
Kinder pro Generation Akquise-MAP-Elites	32
Anzahl Generationen Ergebnis-MAP-Elites	2048
Kinder pro Generation Akquise-MAP-Elites	32
Auflösung der MAP-Elites Karte	25 * 25
Freiheitsgrade	18
Mittelwertgewichtung	1
Varianzgewichtung	2
Constraintgewichtung	1

Tabelle 5: Parametrisierung des zweiten Experiments (Änderungen zum ersten Experiment hervorgehoben)

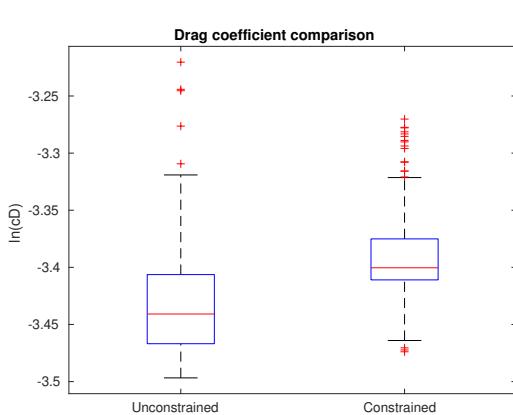


Abb. 23: Vergleich der Luftwiderstände der produzierten Lösungen

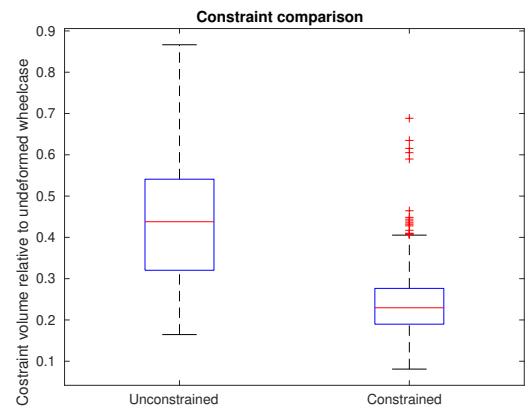


Abb. 24: Vergleich der Constrainterfüllung der produzierten Lösungen

4.1.3 Erhöhung der Laufzeit

Die zweite Einschränkung der beiden vorigen Experimente war die Beschränkung auf eine kleinere Anzahl an ausgewerteten Samples und Generationen in Akquise- und Ergebnis-MAP-Elites. Durch die Beschränkung auf eine kleinere Anzahl an Samples besteht die Möglichkeit, dass der Suchraum nicht präzise genug und/oder nicht weitläufig genug abgebildet wurde. Die Beschränkung in Generationen des Akquise- und Ergebnis-MAP-Elites kann dazu führen, dass diese noch nicht vollständig konvergiert sind. Da die Überlegung nahe liegt, dass die Variante mit Constraint länger benötigen könnte um ausreichend zu konvergieren, sollte untersucht werden, ob die Erhöhung dieser Parameter zu Änderung im Vergleich zur kürzeren Version führt, indem die Gewinne nach 500 Samples und 1024 bzw. 2048 Generationen quantifiziert werden.

Auf ersten Blick sehen die Ergebnisse des dritten Experiments denen des zweiten Experiments sehr ähnlich. Die Karten werden etwas besser gefüllt, was durch Erhöhung der Zahl der ausgewerteten Samples und der Akquise- und Ergebnisgenerations grundsätzlich zu erwarten war.

Die Luftwiderstandskoeffizienten der Variante ohne Constraint haben sich nicht verändert, während bei der Variante mit Constraint eine leichte Verbesserung zu verzeichnen. Dies deckt sich mit der Vermutung, dass die Variante mit Constraint mehr Zeit benötigt um ausreichend auszukonvergieren. Interessant ist auch die Veränderung der Constraintwerte. Hier ist bei der Variante mit Constraint keine nennenswerte Änderung zu verzeichnen,

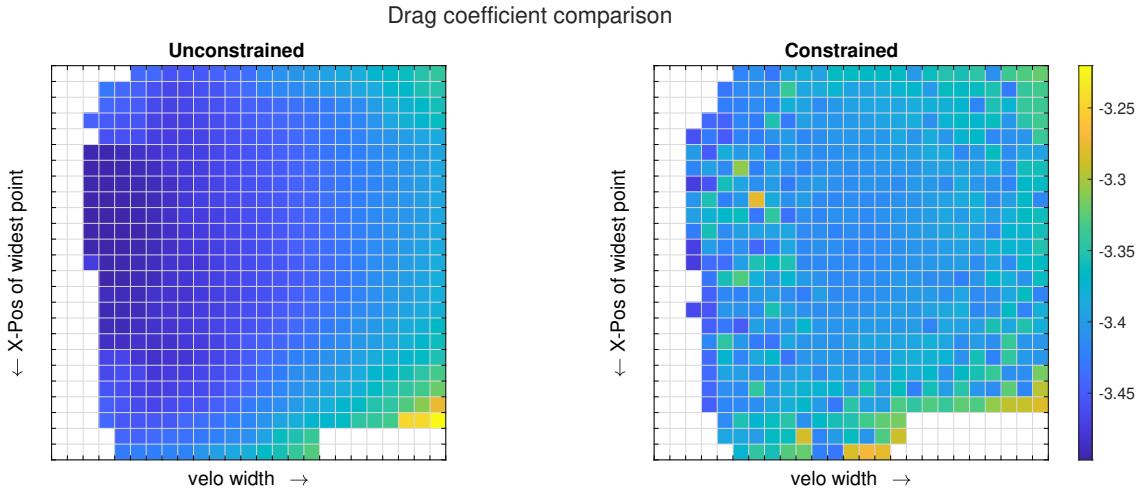


Abb. 25: Karte der Luftwiderstandswerte der Lösungen

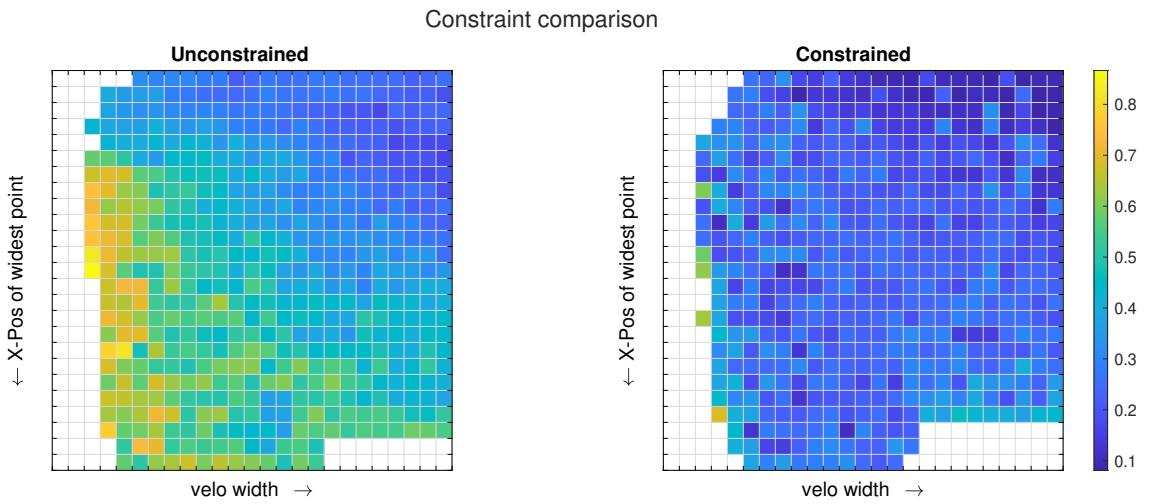


Abb. 26: Karte der Constraintwerte der Lösungen

die Constraintwerte der Variante ohne Constraint verschlechtern sich allerdings. Das ist durchaus interessant, da es bedeuten kann, dass die weitere feine Optimierung bezüglich des Luftwiderstands im Laufe des Algorithmus eine Abkehr von solchen Lösungen darstellt, die den Constraint besser erfüllen. Das dieser Trend in der Variante ohne Constraint nicht zu beobachten ist ein großer Vorteil. Auch ist das interessant, da kein nennenswerter Gewinn bezüglich des Luftwiderstands zwischen ersten und zweitem Experiment zu beobachten ist.

In Abb. 30 ist der Vergleich der Luftwiderstände der berechneten Lösungen der beiden Versionen dargestellt. Die Karte der Variante ohne Constraint bestätigt noch einmal die in Abschnitte 4.1.1 und 4.1.2 bereits beobachteten Trends, dass die Breite des Velomobils negativ mit dem Luftwiderstandskoeffizienten korreliert ist und ein mittiges Optimum bezüglich der x-Koordinate des breitesten Punkts existiert. Es ist festzustellen, dass die Karte etwas besser gefüllt wird als dies noch beim zweiten Experiment der Fall war, dieser Unterschied ist allerdings nur minimal. Außerdem kann an dieser Karte besser als an den vorigen Karten gesehen werden, dass eine Minimalbreite des Velomobils existiert. Auch wenn Deformationen nach innen erlaubt werden können diese keinen Einzug in die Karte halten, da alle Lösungen, bei denen sich der breiteste Punkt am Rand des Radkastens befindet, in die gleiche Spalte eingeordnet werden.

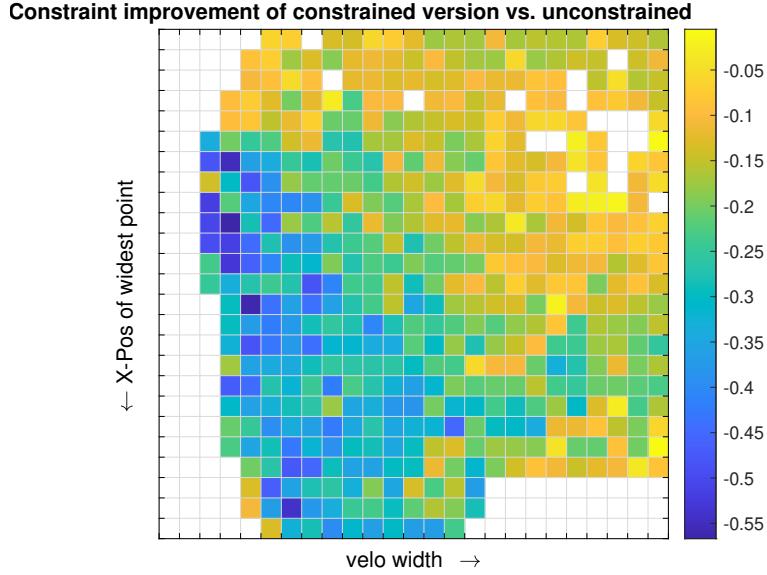


Abb. 27: Verbesserungen des Constraints zwischen Version ohne und mit Constraint

Anzahl initialer Samples	100
Anzahl Samples	1000
Anzahl neuer Samples pro Akquiseschleife	20
Anzahl Generationen Akquise-MAP-Elites	2048
Kinder pro Generation Akquise-MAP-Elites	32
Anzahl Generationen Ergebnis-MAP-Elites	8192
Kinder pro Generation Akquise-MAP-Elites	32
Auflösung der MAP-Elites Karte	25 * 25
Freiheitsgrade	18
Mittelwertgewichtung	1
Varianzgewichtung	2
Constraintgewichtung	1

Tabelle 6: Parametrisierung des dritten Experiments (Änderungen zum zweiten Experiment hervorgehoben)

Auch der Vergleich der Constraintwerte der finalen Lösungen beider Varianten, der in Abb. 31 abgebildet ist, bestätigen sich bereits vorher festgestellte Tendenzen. Die interessanteste Änderung zum zweiten Experiment stellt dabei eine Verschlechterung der Constraintwerte der Variante ohne Constraint im linken unteren Dreieck der Karte dar. Hier bestätigt sich die bei der Betrachtung der Boxplots gemachte Beobachtung, dass vom zweiten aufs dritte Experiment eine Verschlechterung der Constraintwerte stattfindet. An der Karte ist zusätzlich noch zu erkennen, dass dies keine globale Verschlechterung ist, sondern diese nur bestimmte Regionen der Karte betrifft. Dies ist besonders wichtig, da der linke Bereich der Karte, der Bereich ist in dem die Luftwiderstandswerte am besten sind, und damit genau der Bereich ist in dem die Lösungen attraktiver sind. Dass eine Erhöhung der Generationen für Akquise und Ergebnis-MAP-Elites die Verschlechterung von Lösungen bezüglich des Constraints zur Folge hat, wenn dieser nicht explizit behandelt wird, deutet darauf hin, dass die Variante mit Constraint bei längeren Laufzeiten im Vergleich zur Variante ohne Constraint besser abschneidet.

In Abb. 32 ist eine Regression der Lösungen nach Spaltenindex und Luftwiderstandswert zu sehen. Daneben ist für beide Versionen die Regressionsgrade zwischen Spaltenindex und Luftwiderstand aufgetragen. In der Version ohne Constraint ist eine starke Tendenz zu schlechteren Luftwiderstandswerten bei höheren Spaltenindexen, sprich breiteren Velo-

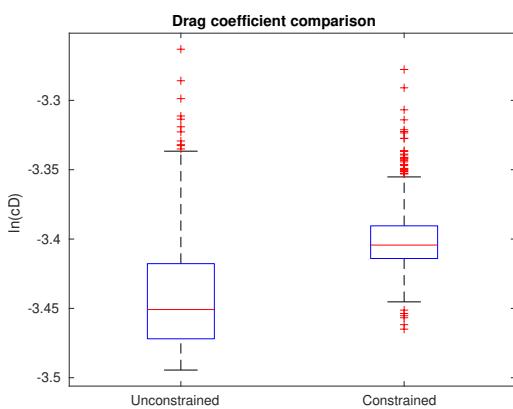


Abb. 28: Vergleich der Luftwiderstände der produzierten Lösungen

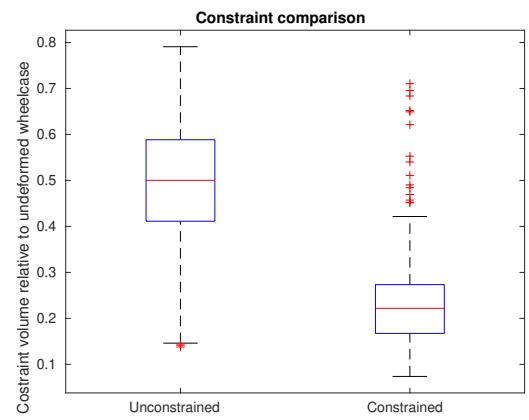


Abb. 29: Vergleich der Constrainterfüllung der produzierten Lösungen

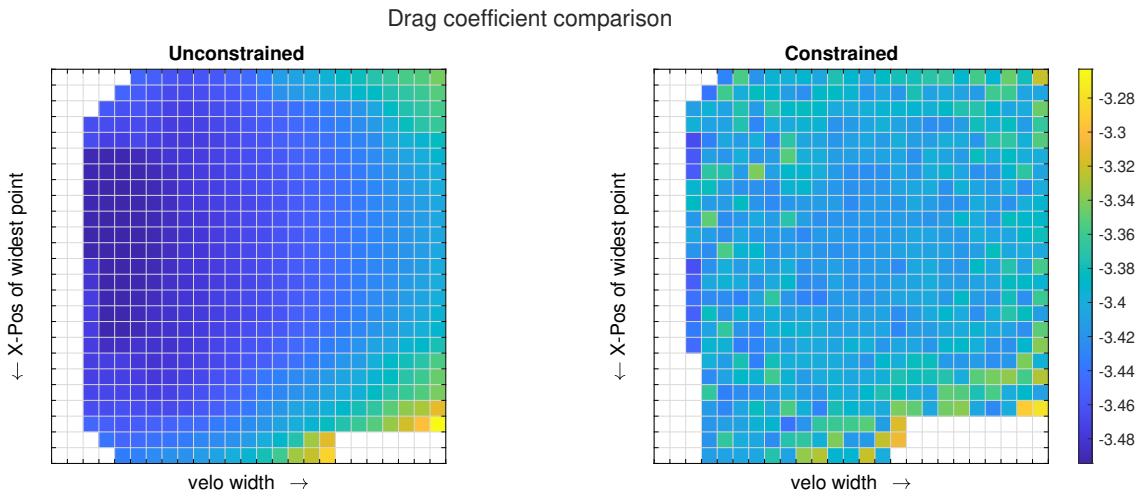


Abb. 30: Karte der Luftwiderstandswerte der Lösungen

mobilien, zu erkennen. Die Korrelation zwischen der Breite des Velomobils und schlechteren Luftwiderstandswerten⁶ wurde in Hypothese **H2** aufgestellt. Der Korrelationskoeffizient von 0,79 verifiziert diese Hypothese. Die Korrelation zwischen Spaltenindex und Luftwiderstand fällt in der Variante mit Constraint wesentlich schwächer aus. Der Korrelationskoeffizient beträgt hier nur 0,27, was allerdings immer noch eine moderate Korrelation zwischen Breite und Luftwiderstand darstellt. Einige Unterschiede zwischen den beiden Versionen sind interessant. Erstens findet eine Verschlechterung der Individuen der linken Hälfte statt. Während in der Version ohne Constraint hier noch verlässlich Wert zwischen $-3,5$ und $-3,45$ erreicht werden, schafft die Version mit Constraint dies nur in einzelnen Ausreißern. Zweitens ist die glatten Ober- und Unterränder in der Variante ohne Constraint zu betonen, die in der Variante ohne Constraint nicht existieren. Zuletzt ist die Spannweite zwischen bestem und schlechtestem Ergebnis in einer Spalte in der Version mit Constraint tendenziell größer, was zum niedrigeren Luftwiderstandskoeffizienten beiträgt.

In Abb. 33 hingegen sind die Luftwiderstandswerte gegen die Zeilenindizes aufgetragen. An den Daten ist klar erkennbar, dass eine lineare Regression nicht angemessen ist, weshalb hier eine quadratische Regression durchgeführt wurde. An den Daten ist die oben bereits gemachte Beobachtung erkennbar, dass ein Optimum für den aerodynamischen Eigenschaften im Bereich [10; 15] existiert. Das heißt Radkästen bei denen der breiteste

⁶Größere Luftwiderstandswerte sind schlechter

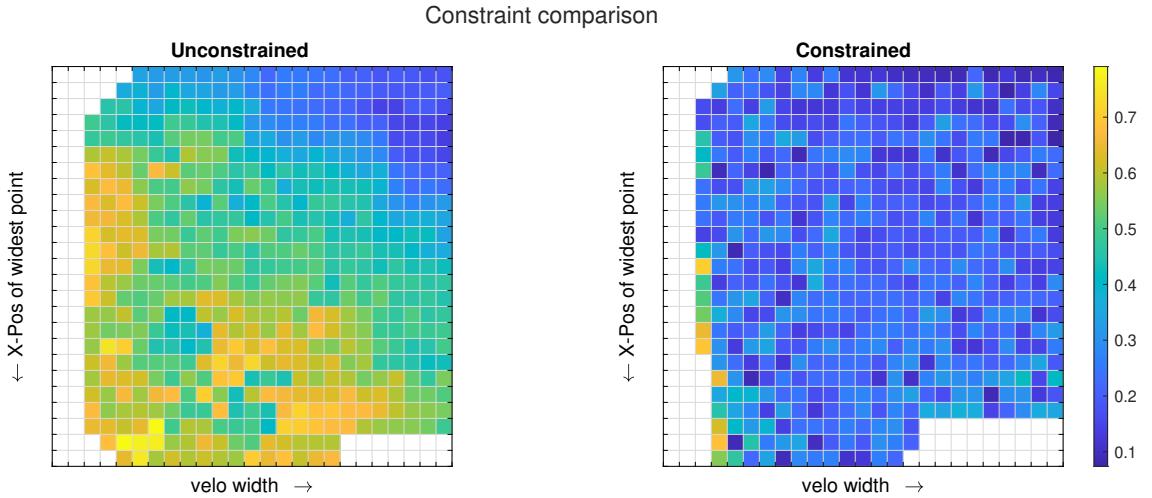


Abb. 31: Karte der Constraintwerte der Lösungen

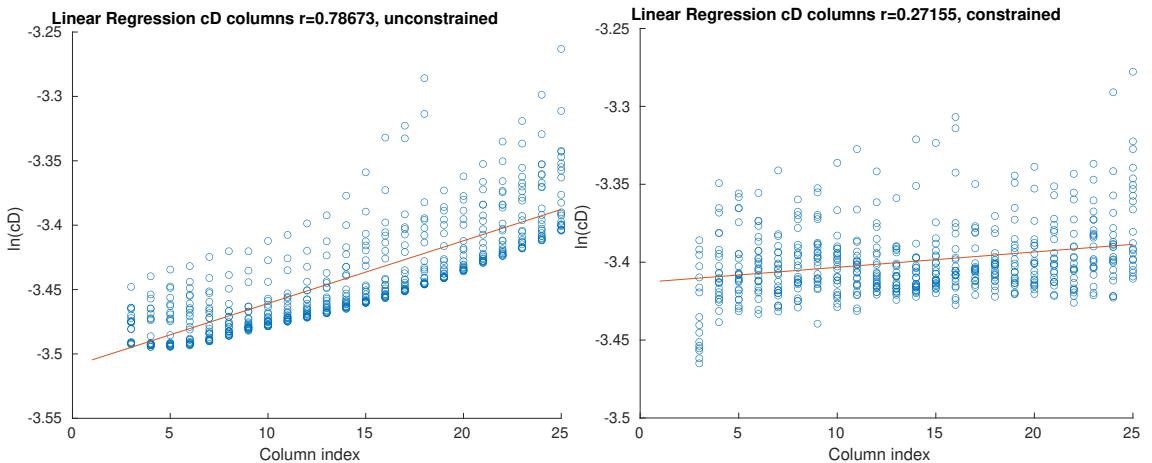


Abb. 32: Regression von Spaltenindex und Luftwiderstand

Punkt zu weit vorne oder zu weit hinten liegt, sind weniger gut als solche bei denen dieser Punkt mittiger liegt. Interessant ist, dass hier ein gleichförmiger und ein gegensätzlicher Trend zur Regression über die Spaltenindizes zu beobachten ist. Auch hier verschwinden die relativ glatten Ober- und Unterbegrenzer von der Version ohne Constraint zur Version mit Constraint. Im Gegensatz zur Regression über die Spalten verringert sich aber die Varianz der Werte und diese klumpen mehr um die Regressionskurve.

In Abb. 34 ist die Constrainterfüllung über die Spaltenindizes für beide Versionen aufgeführt. Oben wurde die Hypothese **H1** aufgestellt, dass breiteren Velomobile grundsätzlich mit einer besseren Constrainterfüllung korreliert sind. Ein Korrelationskoeffizient von $-0,47$ zwischen Spaltenindizes und Constrainterfüllung kann auch diese Hypothese bestätigen. Der Korrelationskoeffizient ist hier zwar nicht so groß wie bei der ersten Hypothese, wo eine sehr starke Korrelation erkennbar war. Unter Beachtung der Tatsache, dass die zweite Kategorie aber die Varianz erhöht, ist er trotzdem groß genug, um von einer starken Korrelation zu sprechen. Auch hier ist der Korrelationskoeffizient der Version mit Constraint niedriger, es ist aber eine klare Verbesserung der erzielten Constraintwerte zu beobachten. Der niedrigere Korrelationskoeffizient hängt hier mit den Individuen in den Spalten 4&5 zusammen, die schlechte Constrainterfüllungen aufweisen. Dadurch, dass die durchschnittliche Constrainterfüllung wesentlich besser ist und die Regressionsgerade tiefer liegt und flacher ist, können diese Punkte durch die Regressionsgerade nicht gut beschrieben werden, wodurch sich ein Korrelationskoeffizient von $-0,28$ ergibt. Daneben ist

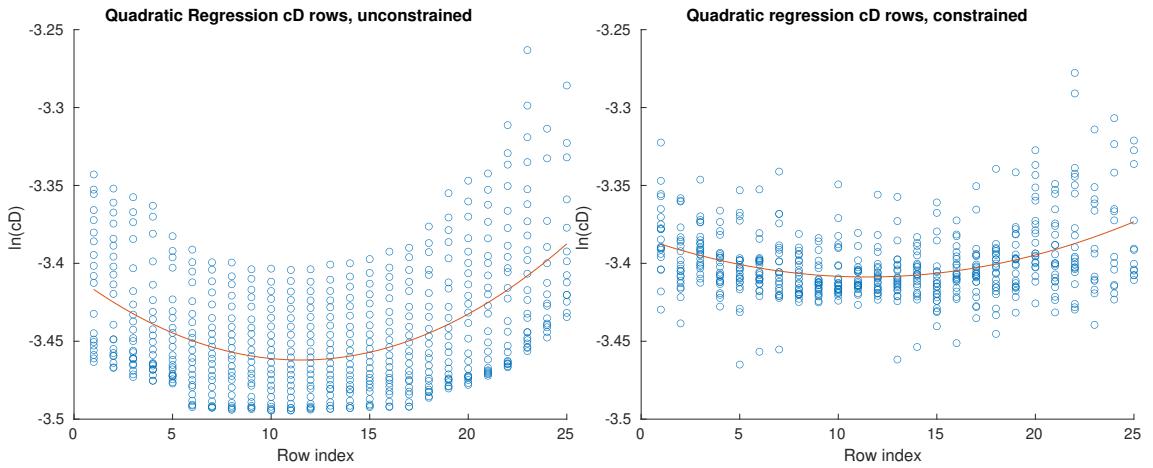


Abb. 33: Quadratische Regression von Zeilenindex und Luftwiderstand

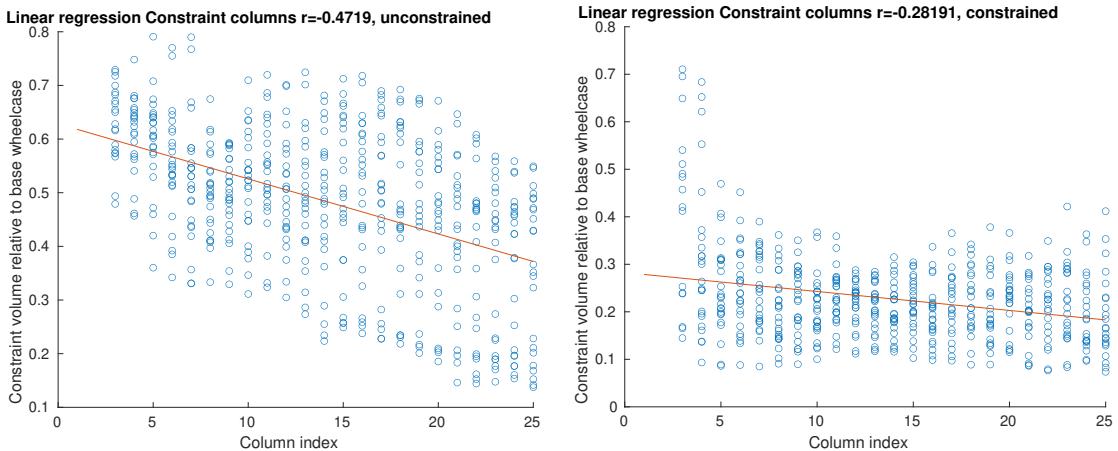


Abb. 34: Regression von Spaltenindex und Constraint

interessant, dass die Korrelationsgerade in der Version mit Constraint wesentlich flacher ist, was bedeutet, dass die Schwierigkeit der Constrainterfüllung weniger stark abnimmt. Da die Gerade aber auch wesentliche tiefer beginnt, ist dass hauptsächlich als Gewinn im linken Bereich (ohne Spalten 4&5) einzuordnen.

Zuletzt wird auch noch für den Constraint die Korrelation zwischen Zeilenindizes, sprich der Position des breitesten Punkts, betrachtet. Die starke Korrelation zwischen einem weiter vorne liegenden breitesten Punkt und einer besseren Constraint interessant. Ein Korrelationskoeffizient von 0,72 ist sehr stark, besonders, da im Voraus keine Vermutungen bezüglich eines so starken Zusammenhangs aufgestellt wurden. Auch hier ist die Regressionsgerade in der Version mit Constraint flacher und niedriger, die beträchtliche Anzahl an Ausreißern nach oben führt zu einem niedrigeren Korrelationskoeffizienten 0,32. Interessant ist die beachtliche Verbesserung bezüglich des Constraints für Individuen, bei denen der breiteste Punkt weiter hinten liegt. Dies ist dadurch zu begründen, dass die Variante mit Constraint mehr Radkästen mit gleichförmigen Breiten generiert. In der Variante ohne Constraint dominieren hingegen Radkästen, die in einem recht kleinen Bereich breit sind.

In Abb. 37 und 40 sind die prozentualen Verbesserungen von Individuen der Akquise-MAP-Elites dargestellt. Die prozentualen Verbesserungen aller generierten Akquisekarten wurden gemittelt und mit einem gleitenden Mittelwert geglättet. Mit dem 2σ Intervall wurde gleichermaßen verfahren. Es ist klar erkennbar, dass beide Varianten konvergieren.

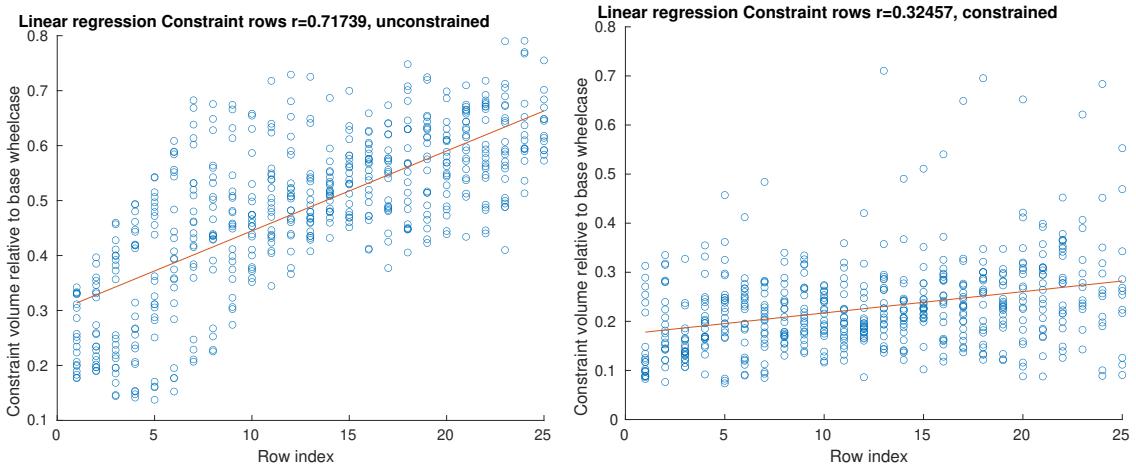


Abb. 35: Regression von Zeilenindex und Constraint

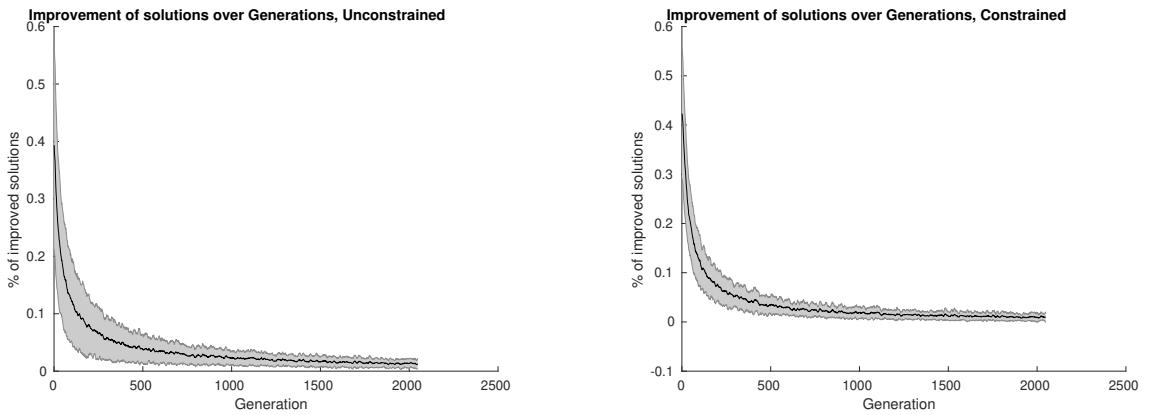


Abb. 36: Prozentuale Verbesserung der Samples in Akquisegeneratio-nen in Version ohne Constraint

Abb. 37: Prozentuale Verbesserung der Samples in Akquisegeneratio-nen in Version mit Constraint

Abbildungen 38 und 39 zeigen die Genotypen der zur präzisen Auswertung ausgewählten Individuen in beiden Varianten. Jeder Boxplot repräsentiert dabei ein Gen des Genotyps. Da in dieser Variante mit 18 Freiheitsgraden gearbeitet wurde ergeben sich 18 Gene. Alle Werte sind auf das Intervall $[0; 1]$ skaliert wobei 0 die minimale (negative) Deformation und 1 die maximale Deformation darstellt. Zuerst ist festzustellen, dass der genetische Raum in beiden Varianten gut ausgefüllt wird. Der Großteil der Gene deckt das gesamte Intervall $[0; 1]$ in deren 2σ Intervall ab. In beiden Varianten existieren fünf unterschiedliche Gene, die das nicht tun, alle dieser Gene weisen allerdings Ausreißer über das gesamte Intervall aus und mit Ausnahme der Gene 13-15 in der Variante mit Constraint decken sie das Intervall mit ihrem 2σ Intervall beinahe ab. Die Tatsache, dass der genetische Problemraum vollständig ausgefüllt wird, ist ein Zeichen, dass nicht zu viele Freiheitsgrade oder zu wenige präzise Funktionsauswertungen genutzt wurden. Das Mapping von Genotyp auf FFD-Deformation fand nach dem folgenden Schema statt:

- 1-3 x-Deformation der drei unteren Deformationspunkte
- 4-6 x-Deformation der drei oberen Deformationspunkte
- 7-9 y-Deformation der drei unteren Deformationspunkte
- 10-12 y-Deformation der drei oberen Deformationspunkte
- 13-15 z-Deformation der drei unteren Deformationspunkte
- 16-18 z-Deformation der drei oberen Deformationspunkte

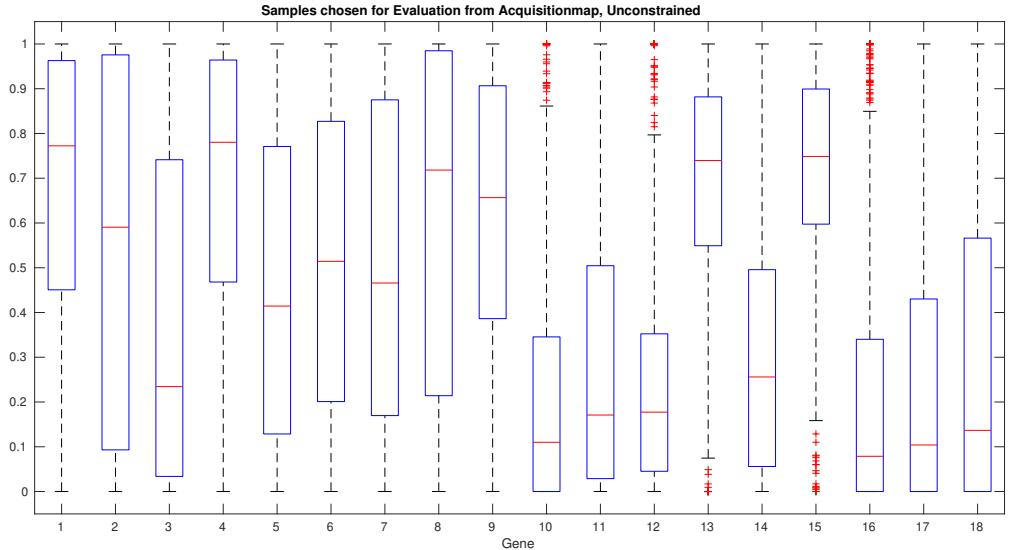


Abb. 38: Genotypen, der für präzise Evaluation ausgewählten Individuen (ohne Constraint)

Einige dieser Triplets sind entweder in sich selbst interessant andere weisen interessante Unterschiede zwischen den beiden Varianten auf.

Das erste besondere Triplet sind die Gene 10-12. Diese sind für die Deformation in y-Richtung der oberen Reihe der Deformationspunkte verantwortlich. In der Variante ohne Constraint fällt auf, dass das gesamte Triplet sehr stark zu kleinen Werten tendiert. Bei allen drei Genen liegt das obere Quartil unter 0,5 und der Median um 0,2. Dies ist eine starke Tendenz zu schwachen beziehungsweise teilweise negativen Deformationen in y-Richtung. Das in der Auswahl der Akquiseindividuen so eine starke Tendenz bezüglich einer der Richtungen existiert lässt darauf schließen, dass in Bereichen mit niedrigen Deformationen der oberen Punkte in y-Richtung mehr Exploitation stattfindet. Besonders interessant ist dies vor allem, wenn man die gleichen Gene in der Version mit Constraint betrachtet. Dort ist die gegensätzliche Tendenz zu erkennen. Zwar fällt die Tendenz zu 1 hier weniger stark aus, als die zu 0 in der Variante ohne Constraint, so ist es doch eine merkliche Tendenz, die durch die Tatsache, dass sie die vorige völlig umkehrt, noch interessanter wird. Die Einführung des Constraints macht Bereiche, die diesen nicht gut erfüllt unattraktiver zur Exploitation. Es ist naheliegend, dass die Deformationen in y-Richtung die sind, die den stärksten Einfluss auf die Erfüllung des Constraints haben. Ebenso ist naheliegend, dass negative Deformationen in y-Richtung den Constraint voraussichtlich nicht gut erfüllen werden. An der Beobachtung, dass der Trend nicht nur neutralisiert, sondern umgekehrt wird, ist erkennbar wie wichtig diese Deformationen zur Erfüllung des Constraints sind.

Diese Beobachtung ist vor allem im Vergleich zu den anderen drei Genen, die auf eine y-Deformation abgebildet werden (7-9), interessant. Bei diesen sind weder so starke Tendenzen, noch eine so starke Umkehr zu beobachten. Daraus kann geschlussfolgert werden, dass y-Deformationen der oberen Reihe wichtige Teile der Lösung des Problems darstellen. Besonders unter dem Blick das diese Gene sowohl zur Optimierung des Luftwiderstands, als auch zur Optimierung des Constraints wichtig sind und diese Gene damit ein Spannungsfeld zwischen diesen beiden Zielen darstellt sollte der Einfluss dieser Gene für das Problem herausgestellt werden.

Das zweite interessante Triplet sind die Gene 13-15. Diese werden Deformationen der unteren Reihe an Deformationspunkten in z-Richtung abgebildet. In der Variante ohne Constraint ist der Unterschied zwischen dem den Genen 13,15 und 14 durchaus interessant.

Offensichtlich scheint das Sampling von Individuen in denen die beiden äußersten Punkte nach oben deformiert werden bevorzugt werden. Gleichzeitig gilt für den mittleren Punkt das Gegenteil, dieser scheint eher nach unten deformiert⁷ zu werden.

Diese Werte dieser Gene sind aber besonders in der Variante mit Constraint interessant. Hier ist ein sehr starker Trend zu negativen Deformationen erkennbar. Der Median der Werte dieser Gene liegt für all unter 0,2. Die Tendenz den mittleren Punkt stärker nach unten zu deformieren bleibt erhalten Aber zumindest für die Gene 13 & 15 ist eine Umkehr von positiven Deformationen in der Variante ohne Constraint zu stark negativen Deformationen in der Variante mit Constraint erkennbar. Die Gene 13 & 15 scheinen also moderat wichtig zur Optimierung des Luftwiderstands zu sein, gleichzeitig aber auch eine Priorität bei der Optimierung des Constraints zu sein. Gleiches gilt auch für Gen 14, nur dass hier die gleichen Tendenzen für beide Optimierungen entstehen. Im Gegensatz zu dem vorigen Triplet und den Genen 13 & 15 entsteht hier also kein Spannungsfeld.

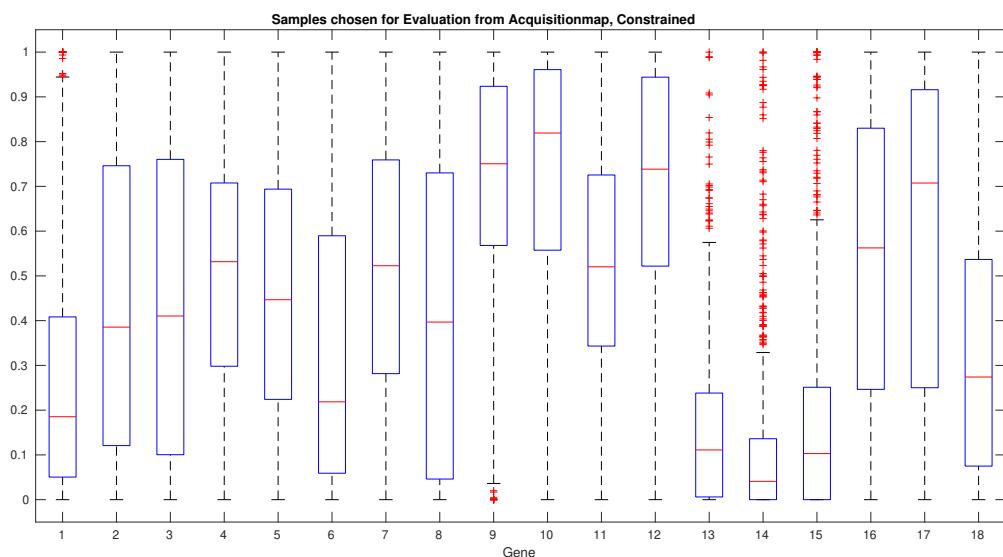


Abb. 39: Genotypen, der für präzise Evaluation ausgewählten Individuen (mit Constraint)

Das letzte interessant Triplet sind die Gene 16-18. Diese werden auf z-Deformationen der oberen Reihe an Deformationspunkten abgebildet. Hier ist ein Trend zu negativen Deformationen in der Variante ohne Constraint zu erkennen. Im Vergleich zur Variante ohne Constraint ist für die Gene 16 & 17 wieder ein umgekehrter Trend zu erkennen, auch wenn dieser nicht so stark wie bei den vorigen beiden Triplets ist. Interessanterweise ist diese Spannung allerdings für das letzte Gen welches auf den hinteren oberen Deformationspunkt abgebildet wird nicht zu erkennen. Insgesamt ist auch hier festzustellen, dass die Gene 16 & 17 wichtig für beide Optimierungsziele sind während diese Spannung für Gen 18 nicht existiert.

Zusammenfassend kann man feststellen, dass die ersten 6 Gene, die auf x-Deformationen abgebildet werden eher unwichtig zu sein scheinen. Stattdessen schienen die größten Spannungsfelder in den Genen zu liegen, die auf y- und z-Deformationen abgebildet werden. Hier sind beträchtliche Änderungen bezüglich der Auswahl der Akquiseindividuen zwischen den beiden Varianten zu erkennen. Interessant ist das die oberen y-Deformationen dabei wichtiger zu sein scheinen als die unteren y-Deformationen und z-Deformationen mehr Einfluss haben als Eingangs vermutet. Auch wichtig ist, dass nicht alle z-Deformationen gleich wichtig sind. Die Gene 14 und 18 scheinen weniger wichtig zu sein, während für die anderen vier Gene Spannungen erkennbar sind.

⁷ z-Deformationen sind symmetrisch um 0,5, alles größer 0,5 ist Deformation nach oben, alles kleiner 0,5 Deformation nach unten

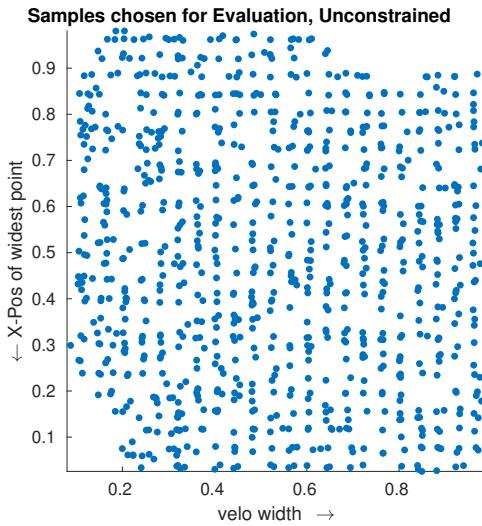


Abb. 40: Kategorisierung der Samples, die für präzise Evaluationen ausgewählt wurden (ohne Constraint)

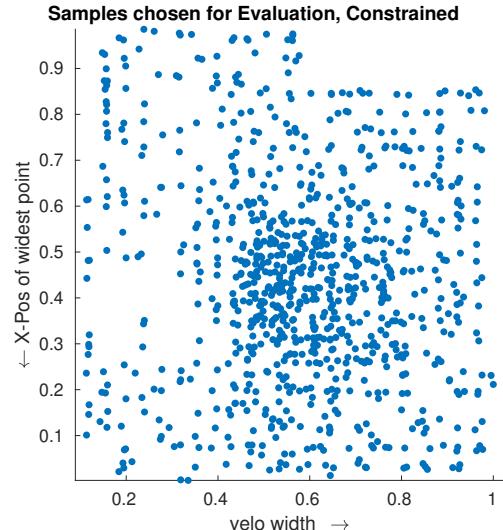


Abb. 41: Kategorisierung der Samples, die für präzise Evaluationen ausgewählt wurden (mit Constraint)

In Abb. 40 und 41 sind die phänotypischen Einordnungen der Individuen, die mithilfe der Akquisekarte zur Auswertung ermittelt wurden, für die Variante ohne und mit Constraint dargestellt. Die Dimensionen sind dieselben wie in den vorigen Karten. Zuerst ist zu beiden Varianten zu sagen, dass diese den Problemraum weitestgehend abdecken. Einzig in der obere rechten und der unteren linken Ecke scheint weniger Exploration stattzufinden. Abgesehen von diesen beiden Bereichen wird der Problemraum allerdings gut abgedeckt. Interessant ist allerdings, dass die Abdeckung des Problemraums in der Variante ohne Constraint wesentlich regelmäßiger ist, als in der Variante mit Constraint. Zwar existieren einige Stellen an denen in der Variante ohne Constraint viele Auswertungen nah beieinander liegen, vor allem im linken Bereich des Graphen, insgesamt wird der Problemraum aber ausreichend weit erkundet.

Im Gegensatz dazu ist eine große Häufung an Akquiseindividuen im mittigen Bereich in der Version mit Constraint zu erkennen und eine sehr geringe Konzentration an Akquise-individuen im linken Bereich bei $0,2 \leq x \leq 0,4$. Dies ist besonders interessant dadurch, dass gerade der Bereich $0,2 \leq x \leq 0,4$ der ist, in dem die aerodynamisch besten Individuen liegen. Durch die Einführung des Constraints, der effektiv eine Manipulation des Mittelwerts von UCB darstellt, wird der Bereich in dem Exploitation stattfindet dorthin verschoben, wo dieser Constraint besser erfüllt ist. Dies ist grundsätzlich so gewollt, die Anhäufung von Punkten im mittleren Bereich kann allerdings ein Zeichen dafür sein, dass der Constraint im Vergleich zum Luftwiderstand zu stark gewichtet ist. Gegebenenfalls wäre es sinnvoll den Constraint schwächer zu gewichten oder in der Akquisefunktion vollkommen wegzulassen und nur in der finalen Auswertung mit MAP-Elites zu betrachten.

Im Folgenden werden die Phänotypen einiger Individuen betrachtet und jeweils die Lösung der Zelle der beiden Versionen verglichen. In den folgenden Abbildungen sind jeweils die Draufsicht sowie eine perspektivische Sicht auf den rechten Radkasten abgebildet. Neben dem Velomobil ist auch das modellierte Constraintvolumen dargestellt und zur Differenzierung rot hervorgehoben.

In Abb. 42 sind die von MAP-Elites gefundenen Lösungen der Zelle (4-3) beider Versionen abgebildet. Lösungen in dieser Zelle sind schmal, haben allerdings einen weit vorne liegenden breitesten Punkt. Es ist ersichtlich, dass dies der Fall ist. Es ist allerdings auch erkennbar, dass der Constraint in beiden Versionen nicht nur nicht erfüllt wird, sondern

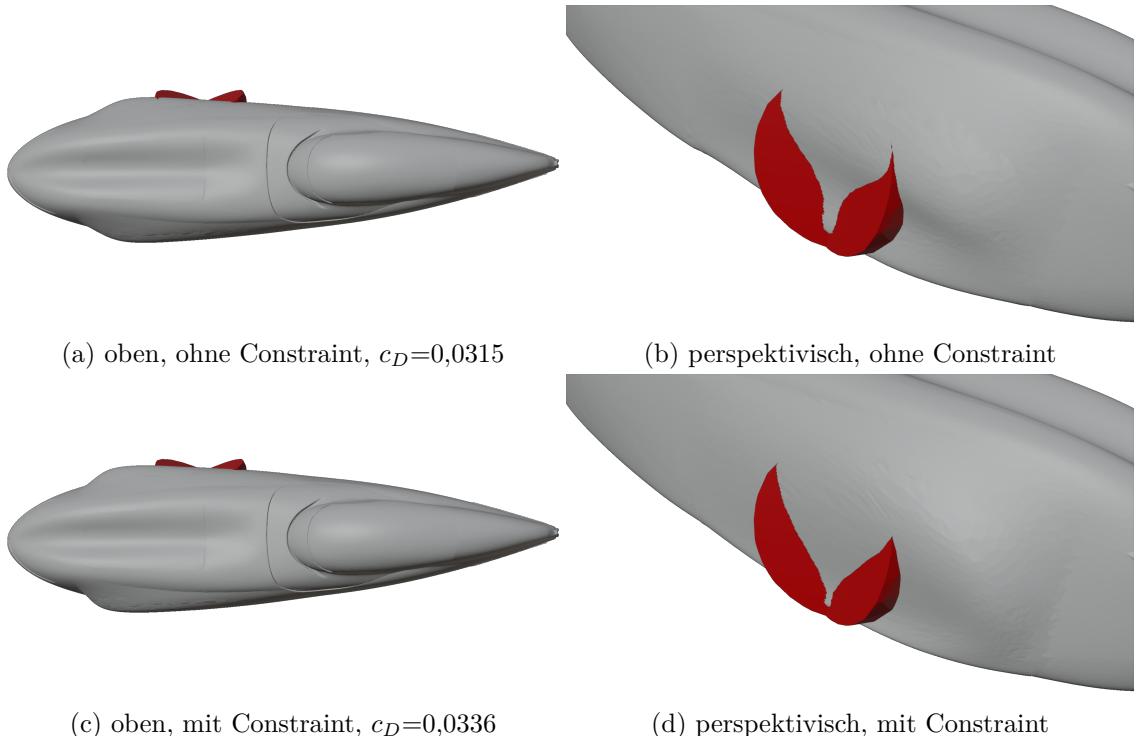


Abb. 42: Die Phänotypen des Individuums (4-3) mit und ohne Constraint

sogar sehr stark verletzt wird. Es findet zwar eine klare Verbesserung des Constraints statt, wenn dieser explizit behandelt wird, der gewünschte Effekt, dass dieser aber vollkommen erfüllt wird ist hier nicht erreicht worden. Dies hängt einerseits mit der Kategorisierung zusammen. Aus den beiden Draufsichten ist klar erkennbar, dass ein so schmales Velomobil den Constraint nie vollständig erfüllen kann. Trotzdem interessant ist die Tatsache, dass die Variante mit Constraint nicht die maximal mögliche Constrainterfüllung aufweist, die bei der Breite möglich ist. Stattdessen gehören beide Lösungen zur gleichen Lösungsklasse, die durch eine breitere Oberseite aber schmalere Unterseite charakterisiert wird. Dass die Lösung, die in der Variante mit Constraint generiert wird, die physikalisch maximal möglich Constrainterfüllung nicht erlaubt, ist dadurch zu begründen, dass Constraint und Luftwiderstand gegeneinander gewichtet werden. Die in der Variante ohne Constraint generierte Lösung stellt hierbei die in dieser Zelle optimale Lösung da. Ein Abweichen von dieser Form muss jede Verschlechterung der aerodynamischen Eigenschaften durch eine Verbesserung des Constraint aufwiegen. Dadurch entsteht aber das Problem der Gewichtung der beiden Ziele gegeneinander, welches enorm schwer zu lösen ist. Will man Lösungen generieren, die den Constraint so gut wie möglich erfüllen unabhängig der damit verbundenen Kosten in der Aerodynamik, oder will man einen Kompromiss, wie er hier generiert wurde finden. Wenn man einen Kompromiss finden will, muss man sich die Frage stellen, wo dieser liegen soll, bei einer 1:1-Gewichtung wie hier oder beispielsweise bei einer 2:1-Gewichtung.

In Abb. 43 sind schmale Velomobile, deren breitesten Punkt im Gegensatz zum vorherigen Beispiel weit hinten liegt, dargestellt. Trotzdem sind viele der in Abb. 42 bereits beobachteten Tendenzen auch hier zu finden. In beiden Varianten wird der Constraint auch hier nicht erfüllt. Es ist zwar auch hier eine leichte Verbesserung zu erkennen, diese ist aber eher gering. Besonders dadurch, dass die schmalen Velomobile den Constraint in der Variante ohne Constraint so stark verletzen lässt eigentlich viel Raum für Verbesserung, der offensichtlich nicht genutzt wird. Dies ist, hängt genau wie im vorigen Beispiel mit der Gewichtung zwischen Luftwiderstand und Constraint zusammen, durch die nur geringfügig

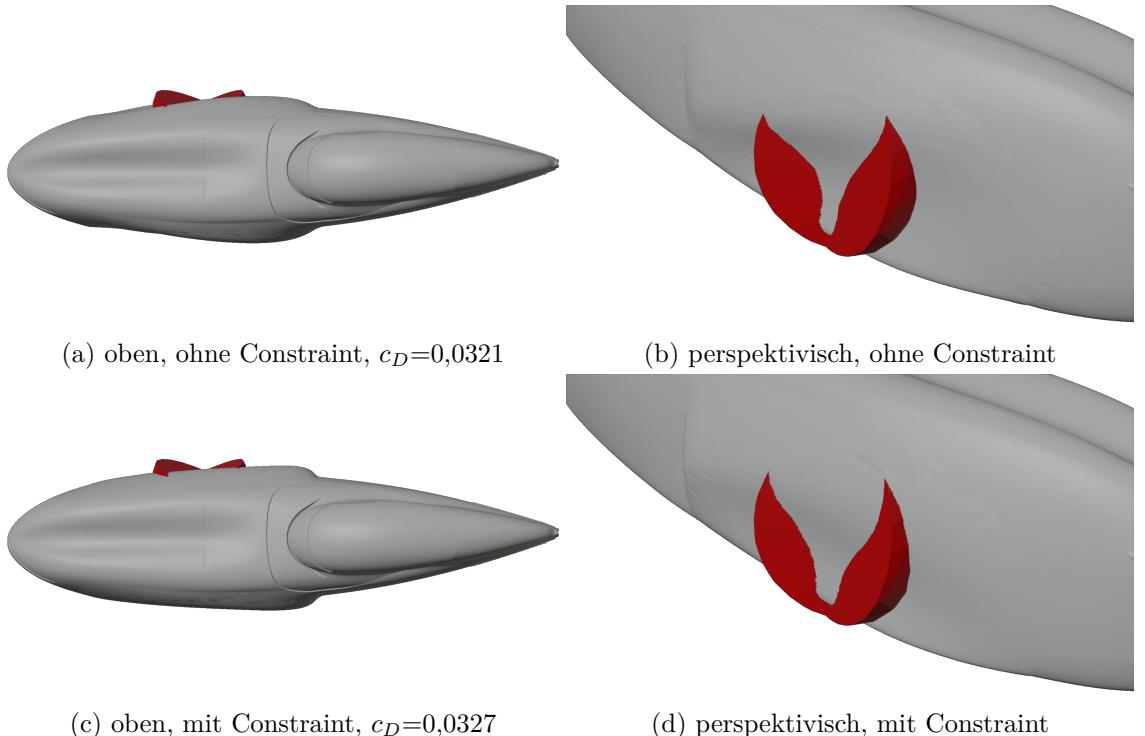


Abb. 43: Die Phänotypen des Individuum (4-24) mit und ohne Constraint

gige Verbesserungen ermöglicht werden. Auch ist interessant, dass dieser Phänotyp genau wie der im ersten Beispiel oben breiter und unten schmäler ist.

In Abb. 44 ist ein Individuum des anderen Extremes dargestellt. Der dargestellte Phänotyp ist sehr breit und dessen breitesten Punkt liegt weit hinten. Es ist zu erkennen, dass der Constraint bereits in der Variante ohne Constraint recht gut erfüllt ist. Dies deckt sich mit den Beobachtungen aus der Karte. Da der Constraint in der Variante ohne Constraint bereits gut erfüllt ist, ist hier nicht viel potenzielle Verbesserung möglich. Trotzdem findet eine Verbesserung statt und in der Variante mit Constraint ragt nur nach minimal aus dem Radkasten heraus. Es ist interessant zu beobachten, dass diese Lösung einer anderen Lösungsklasse angehört als die beiden schmalen betrachteten. Diese Lösung ist unten breiter und oben schmäler, es ist also ersichtlich, dass nicht nur dieselben Lösungen mit stärkeren und schwächeren Deformationen erzeugt werden. Trotzdem stellen diese breiten Phänotypen aerodynamisch suboptimale Lösungen dar, gerade hier ist die Erzeugung von Lösungen zwar einfach aber eher uninteressant. Das Ziel wäre schmalere Lösungen zu generieren die sowohl aerodynamische als auch Constraintoptimalität aufweisen.

Alle vorigen Individuen befanden sich an den Rändern der Karte und stellten damit Extrema dar, zum Verständnis sollten sich daneben allerdings auch zentraler gelegene Individuen angeschaut werden. In Abb. 45 sind zwei solcher weniger extremen Individuen abgebildet. Die größte Sache, die im Vergleich zu den vorher betrachteten Individuen auffällt, ist die auffällig starke Änderung bezüglich des Constraints. Sowohl bei sehr schmalen als auch bei sehr breiten Individuen waren nur relativ kleine Änderungen bezüglich des Constraints zu erkennen. Bei schmaleren Velomobilen lag dies daran, dass die geringe Breite eine vollständige Constrainterfüllung nicht zulässt und dass die Gewichtung zwischen Aerodynamik und Constraint so gewählt wurde, dass keine weiteren Verschlechterungen der Aerodynamik für Constraintgewinne in Kauf genommen wurden. Bei breiten Velomobilen war der Constraint in der Variante ohne Constraint bereits weitestgehend erfüllt, wodurch nur minimale Verbesserungen durch die Einführung eines Constraints erreicht werden konnte.

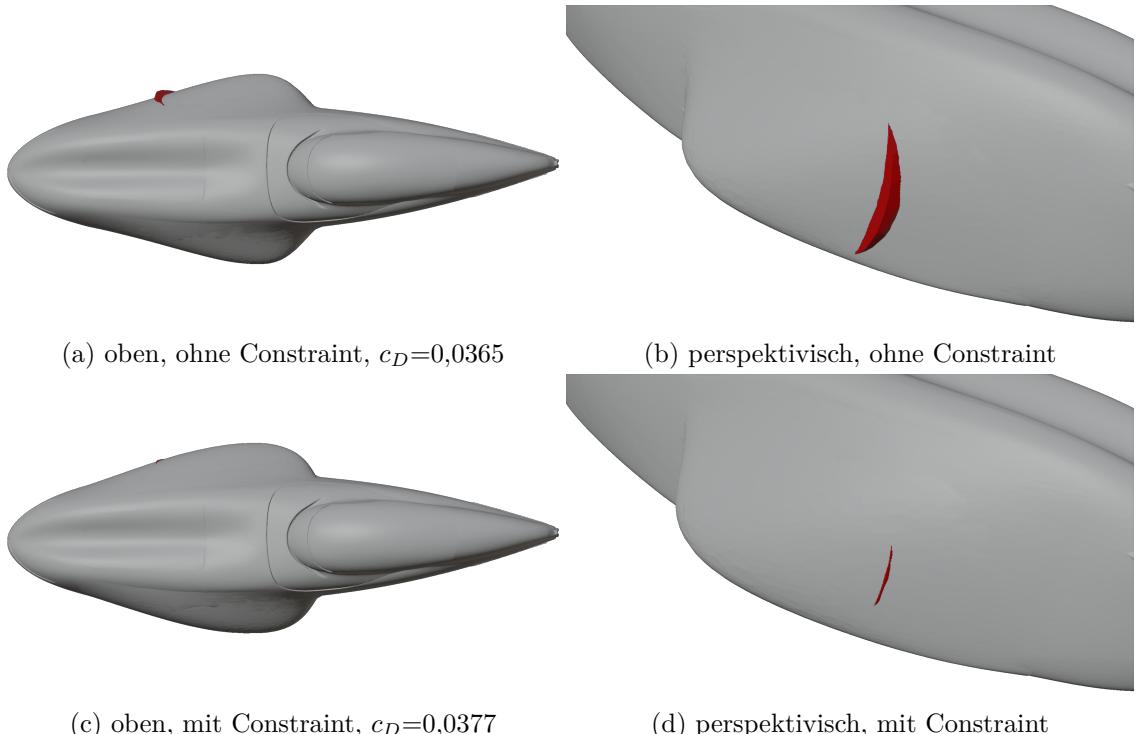


Abb. 44: Die Phänotypen des Individuums (25-22) mit und ohne Constraint

Beide dieser Dinge gelten nicht für ein ausgeglichenes Individuum. In der Variante ohne Constraint ist dieser offensichtlich nicht erfüllt. Die Verbesserung in der Variante mit Constraint ist allerdings wesentlich größer. Der Constraint ist in dieser Variante bereits nahezu erfüllt, es sind noch zwei leichte Bänder auszumachen, in denen das Constraintvolumen aus dem Velomobil ragt, allerdings sind diese sehr klein. Interessant ist auch der Unterschied zwischen der Lösungsklassen. Während in der Variante ohne Constraint noch eine unten breitere oben schmalere Form präferiert wird – Im Gegensatz zu dem bei schmäleren Lösungen präferierten Form, die oben breiter und unten schmäler war – ist die Form in der Variante mit Constraint wesentlich gleichmäßig breiter. Dies gilt in der Vertikalen, in der sowohl der untere Teil des Radkastens als auch der obere Teil breit sind, aber auch in der Horizontalen, in der der Radkasten wesentlich länger breit bleibt.

Was sich aus den betrachteten Phänotypen ablesen lässt ist, dass der Constraint überall beachtet wird und sich verbessert oder zumindest nicht verschlechtert. Bei besonders breiten Velomobilen konnten keine starken Verbesserungen erreicht werden, da der Constraint bei solchen breiten Lösungen meist schon fast vollständig erfüllt ist. Hauptgewinne bezüglich des Constraints konnten bei zentraler Individuen gesehen werden, besonders bezüglich der Breite. Bei schmäleren Individuen wurden die Constraintgewinne schmäler was auf eine Kombination aus Konflikt zwischen Aerodynamik und Constraint und in der Tatsache, dass constrainterfüllende Individuen schwerer zu generieren sind, je schmäler diese sein sollen.

4.2 E-Roller

In Abb. 46 ist die von SAIL für die E-Roller-Domäne produzierte Karte abgebildet. Es ist erkennbar, dass die berechneten Fitnesswerte einen erstaunlich großen Bereich abdecken, dafür, dass nur ein eher kleines Bauteil an der Unterseite des E-Rollers deformiert wird. Aufgrund der oben erwähnten Einschränkung bezüglich der Ergebnisse, sollte dies nicht als Zeichen gedeutet werden, dass Deformationen in diesem Bauteil tatsächlich so große Änderungen bewirken können. Dass solche Unterschiede existieren ist zur Anwendung von

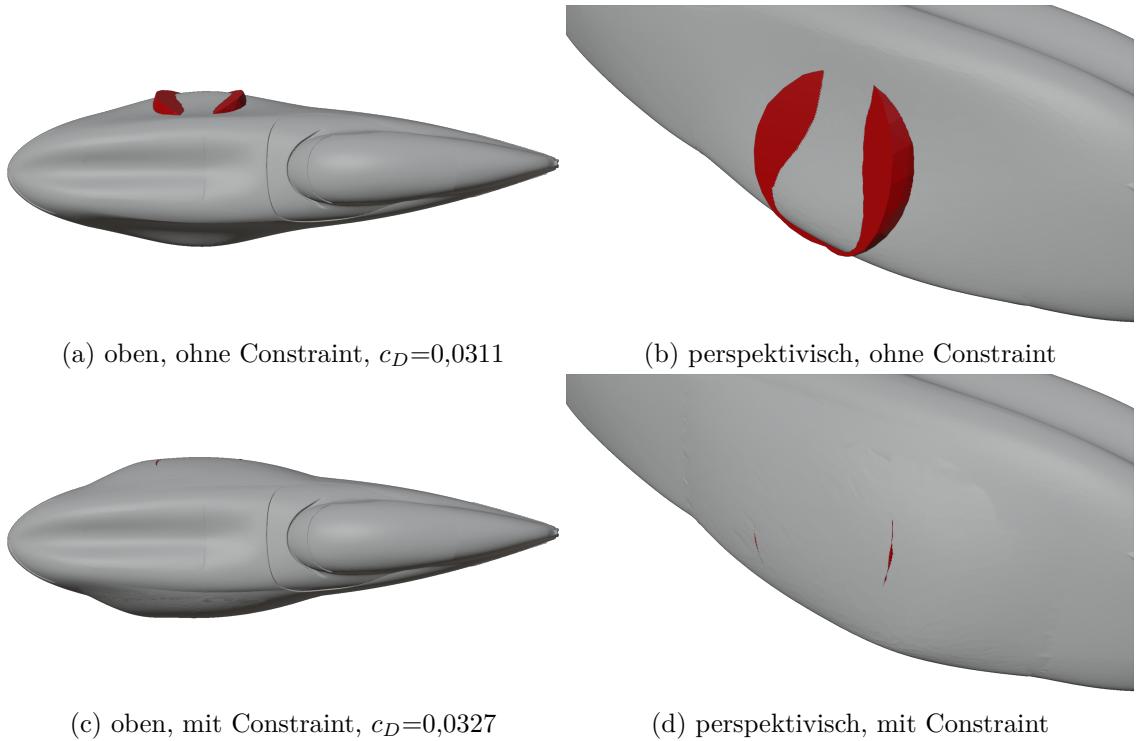


Abb. 45: Die Phänotypen des Individuums (12-12) mit und ohne Constraint

Anzahl initialer Samples	60
Anzahl Samples	500
Anzahl neuer Samples pro Akquiseschleife	10
Anzahl Generationen Akquise-MAP-Elites	2048
Kinder pro Generation Akquise-MAP-Elites	32
Anzahl Generationen Ergebnis-MAP-Elites	4096
Kinder pro Generation Akquise-MAP-Elites	32
Auflösung der MAP-Elites Karte	25 * 25
Freiheitsgrade	20
Mittelwertgewichtung	1
Varianzgewichtung	2

Tabelle 7: Parametrisierung des E-Roller Experiments

SAIL allerdings gut, sind die Variationen zwischen Individuen zu gering kann SAIL nicht korrekt arbeiten, da der als Surrogatmodell genutzte Gaußprozess auf solchen Daten nur schlecht trainiert werden kann. Der Gradient der auf den generierten Daten erkennbar ist lässt allerdings auch darauf schließen, dass die Ergebnisse von Openfoam nicht zu stark Rauschen. Wäre alle Varianz der Ergebnisse von Openfoam bloß auf Rauschen zurückzuführen, wäre es nicht zu erwarten, dass eine Karte mit einem relativ glatten Gradienten erzeugt wird. Diese Tatsache kann auch an den Hyperparametern, die für die E-Roller-Daten optimiert, werden abgelesen werden. Diese sind in Abschnitt 4.2 aufgeführt. Das benötigte Signalrauschen, welches von der Optimierungsfunktion berechnet, wird um die Daten ausreichend beschreiben zu können beträgt 0,0114, und ist damit ungefähr zehnmal kleiner als die Reichweite der Lösungen in der Karte. Aus diesen Beobachtungen kann geschlossen werden, dass die Ergebnisse der Openfoam-Simulation zumindest ausreichen um SAIL mit diesen durchführen zu können. Dies sagt selbstverständlich nichts über die Korrektheit der Openfoam-Simulation aus nur, dass die vorliegenden Daten von ihren qualitativen Eigenschaften für SAIL geeignet sind.

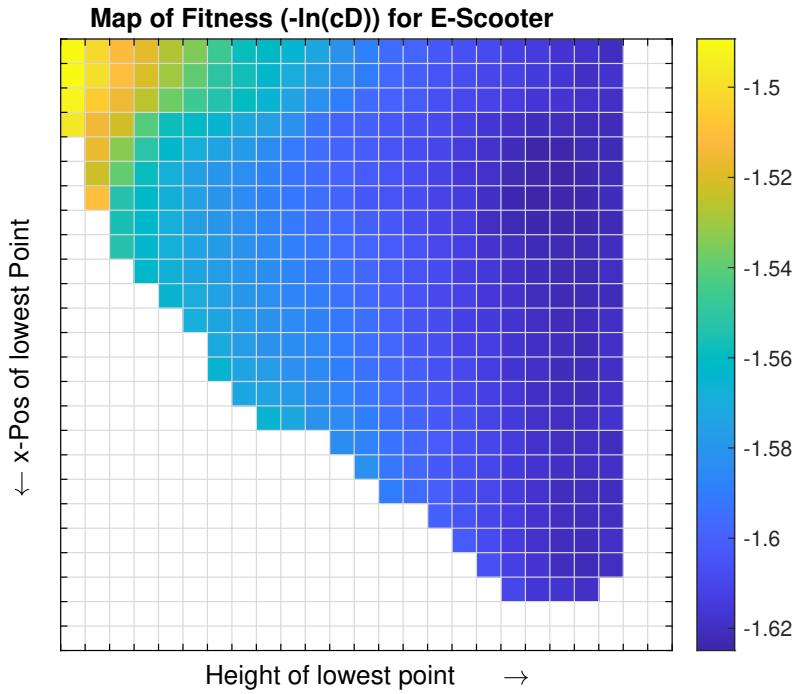


Abb. 46: Karte der Luftwiderstandswerte der E-Roller Domäne

ℓ	2,2791
σ_f	0,0473
c	-1,4904
σ_n	0,0114

Tabelle 8: Die Hyperparameter des Gaußprozesses nach Anpassung

Auch die sonstigen Hyperparameter des Gaußprozesses liegen in plausiblen Größenordnungen. Der Mittelwert von $-1,49$ liegt sehr exakt bei den in der Karte dargestellten Werten. Ein Längenmaß von $\ell = 2,28$ ist zwar eher groß, problematischer wäre es allerdings, wenn dieses zu klein wäre. Ein typisches Zeichen von einem zu groß gewählten Längenmaß ist, dass starkes Rauschen benötigt wird, um die Daten zu erklären. Da das Rauschen, durch σ_n bestimmt, allerdings relativ klein ist, ist ein solches Längenmaß nicht außergewöhnlich. Unter der Annahme, dass die Werte aus den Openfoam-Simulationen ausreichen präzise sind, lassen sich einige Dinge aus der Karte ablesen. Das erste ist der Gradient in der Horizontalen. Lösungen deren tiefster Punkt tiefer liegt weisen schlechtere Luftwiderstandscoeiffizienten auf als solche, die flacher sind. Die zweite Sache, die in der Horizontalen auffällt, ist der strenge Abriss von Lösungen nach der 23ten Spalte. Dies ist durch die Auswahl der ersten Kategorie zu erklären. So ändert sich der tiefste Punkt für Deformationen nach unten, und wird meist einer der deformierten Punkte sein. Für Deformationen nach oben gilt dies nicht, stattdessen wird dort ein Punkt am Rand der FFD-Box zum tiefsten Punkt, der unabhängig von der Stärke der Deformation immer der tiefste Punkt bleiben wird. Alle Bauteile, die entweder sehr flach sind oder nach oben deformiert werden, werden also in die rechteste Befüllt Spalte eingeordnet, da es mit der momentanen FFD-Konfiguration nicht möglich ist Individuen Bauteile zu generieren, deren tiefster Punkt höher liegt. Entlang der anderen Dimension ist kein so starker Effekt zu beobachten, hier fallen allerdings zwei Dinge auf. Hier scheint eine Optimum, etwa in der fünften Zeile, zu existieren. Sowohl Lösungen deren tiefster Punkt weiter vorne liegt, als auch solche, deren tiefster Punkt weiter hinten liegt besitzen leicht schlechtere Luftwiderstandswerte. Die zweite auffällige Eigenschaft in der Vertikalen ist, dass die Anzahl an Lösungen pro

Spalte nicht vergleichbar ist, sondern zunimmt, je höher der tiefste Punkt der Bauteile der Spalte liegt. Dies ist durchaus interessant, da die momentanen Deformationspunkte grundsätzlich in der Lage sein sollten einen tiefsten Punkt weiter vorne zu generieren. Der Grund dafür ist die Grundform des Bauteils. Dieses ist hinten bereits tiefer als vorne. Dadurch, dass das Bauteil trotzdem sehr flach ist, wurde auch die FFD-Box flach gewählt. Da die Verschiebung der Meshpunkte abhängig von ihrer relativen Position zu den Deformationspunkten erfolgt führt dies dazu, dass die vorderen Bereiche, die weiter entfernt von den unteren Deformationspunkten sind, niemals vergleichbar weit nach unten deformiert werden können, wie es für Punkte im hinteren Bereich geschehen kann.

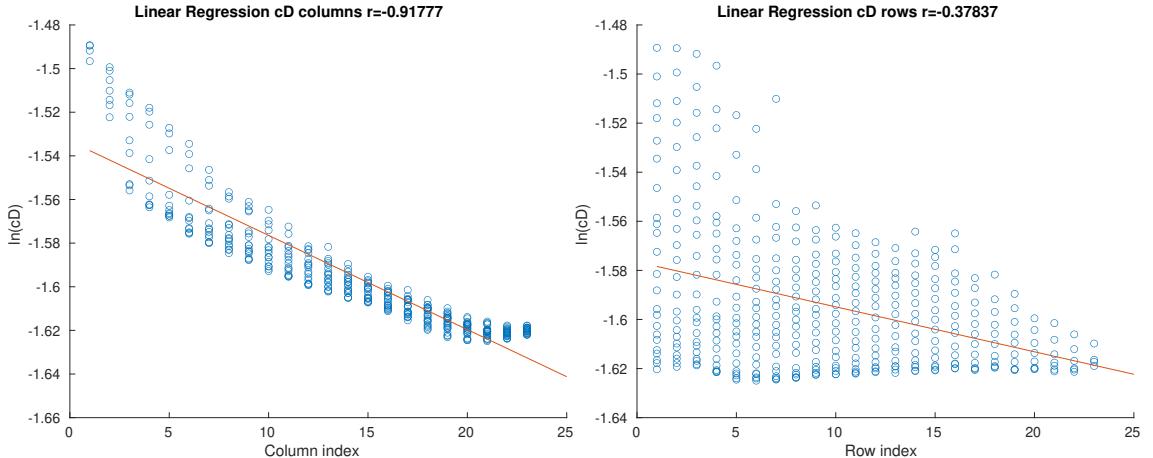


Abb. 47: Regression für c_D gegen Spaltenindex und Zeilenindex

In Abb. 47 sind die Luftwiderstandswerte gegen Spalten- beziehungsweise Zeilenindizes aufgeführt. Auch ist jeweils die Regressionsgerade und der Korrelationskoeffizient dargestellt. Wie bereits oben erwähnt ist eine klare Tendenz zu besseren Luftwiderstandswerten bei weniger stark nach unten deformierten Bauteilen zu erkennen. Diese Tendenz wird auch durch einen Korrelationskoeffizienten von $-0,92$ bestätigt, der auf eine sehr starke Korrelation schließen lässt. Entlang der Zeilen, sprich der Position des tiefsten Punkts, wird zwar auch eine moderate Korrelation mit einem Korrelationskoeffizienten von $-0,38$ ermittelt. Diese sollte allerdings skeptisch betrachtet werden, da die Anzahl an Datenpunkten für höhere Zeilenindizes abnimmt. Dadurch der Schein eines Musters erweckt, das allerdings nur ein Produkt der ungleichmäßig gefüllten Karte sein kann. Ob sich der Luftwiderstand tatsächlich bei steigendem Zeilenindex verbessert kann nur bewertet werden, wenn die Karte gleichmäßiger gefüllt wird. Da in der momentanen Version für größere Zeilenindizes vorrangig weniger stark deformierte Bauteile erzeugt werden, kann der positive Effekt durch steigende Zeilenindizes nur Nebenprodukt, der Korrelation zwischen Spaltenindex und Luftwiderstand sein.

In Abb. 48 ist die prozentuale Verbesserung über die Akquisegenerationen abgebildet. Auch diese sieht so weit nicht uncharakteristisch aus. Es werden Verbesserungen zu Lösungen gefunden und über die Generationen konvergiert die Anzahl an Verbesserungen immer stärker, wodurch die charakteristische logarithmische Form entsteht.

In Abb. 49 (a) ist der Phänotyp des ersten Individuums in der ersten Spalte von vorne dargestellt. In Abb. 49 (b) ist derselbe Phänotyp von der Seite dargestellt. Individuen der ersten Spalte zeichnet aus, dass deren tiefster Punkt besonders tief ist. Individuen der ersten Zeile zeichnet aus, dass deren tiefster Punkt weiter hinten liegt. Es ist sehr klar erkennbar, dass der dargestellte Phänotyp diese Kategorien erfüllt.

Der Phänotyp in Zelle (23,22)(3. Spalte von links, 4. Zeile von unten) dargestellt in Abb. 50 stellt das Gegenteil des vorigen Phänotyps dar. Dessen tiefster Punkt sollte besonders weit oben und eher vorne liegen. Insgesamt ergibt sich dadurch eine sehr flache Form,

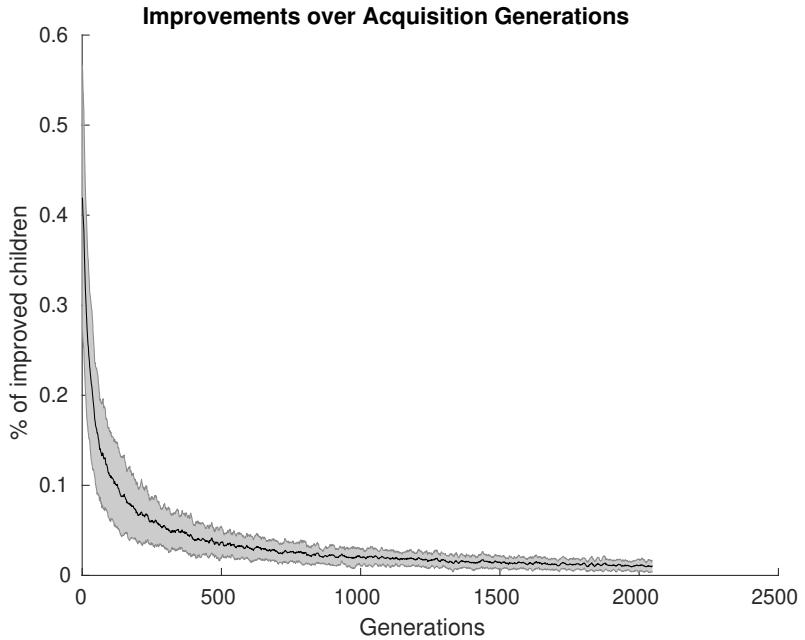


Abb. 48: Prozentuale Verbesserung der Kinder pro Akquisegeneration

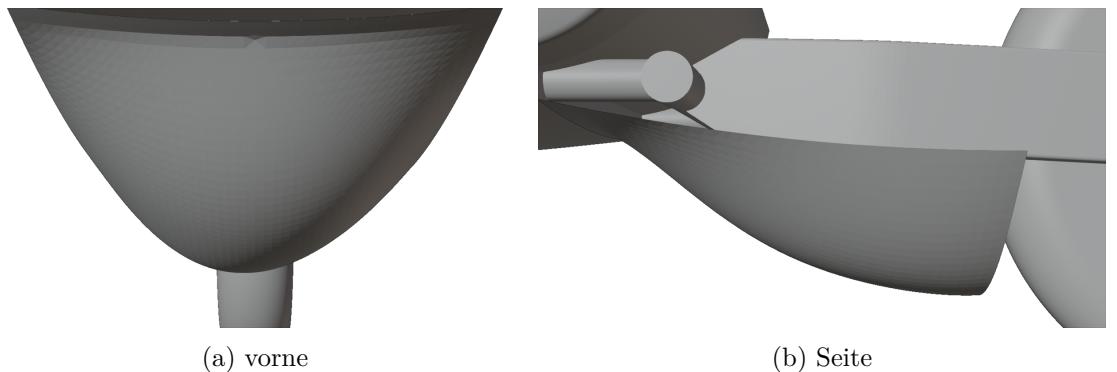


Abb. 49: Die Phänotypen des Individuums (1-1), $c_D=0,2255$

deren tiefster Punkt weit vorne liegt und die sogar leicht das Gestänge der Radhalterung schneidet.

Das dritte Beispiel ist im Gegensatz zu den vorigen beiden keines der Extrema, sondern ein Individuum aus dem mittigen Bereich der Karte. Das Individuum (9,16) dargestellt in Abb. 50 befindet sich am Dreiecksrand der Karte. Dieses Individuum ist nicht so schlecht wie das erste ausgewählte Individuum, bietet aber trotzdem eine interessantere deformierte Form als das untere rechte Individuum. Es ist klar erkennbar, dass eine Deformation nach unten stattfindet, diese aber wesentlich schwächer ausfällt als im ersten Beispielindividuum. Sehr interessant ist auch, dass das Individuum (9-16) nach hinten wieder abflacht beziehungsweise geschlossen ist. Dies steht im Gegensatz zum ersten Beispielindividuum, welches nach hinten geöffnet war. Das bedeutet, dass das Ziel von SAIL nicht nur morphologisch unterschiedliche Lösungen derselben Lösungsklasse zu generieren, sondern eine Diversität an Lösungsklassen zu generieren teilweise erfüllt wird.

Das letzte Beispiel, dessen Phänotyp genauer angeschaut wird liegt im letzten Bereich der Karte auf den bisher noch kein Blick geworfen wurde, der oberen rechten Ecke. In Abb. 52 ist das Individuum (20,5) abgebildet, welches nicht das extremste Individuum der rechten Ecke ist, sondern, dass welches den optimalen Fitnesswert aufweist. Es ist herauszustellen wie ähnlich dieses Individuum dem Individuum (23,22) ist. Zwar liegt der tiefste Punkt weiter hinten, da beide Individuen allerdings sehr flach sind, ändert die Position des tiefsten

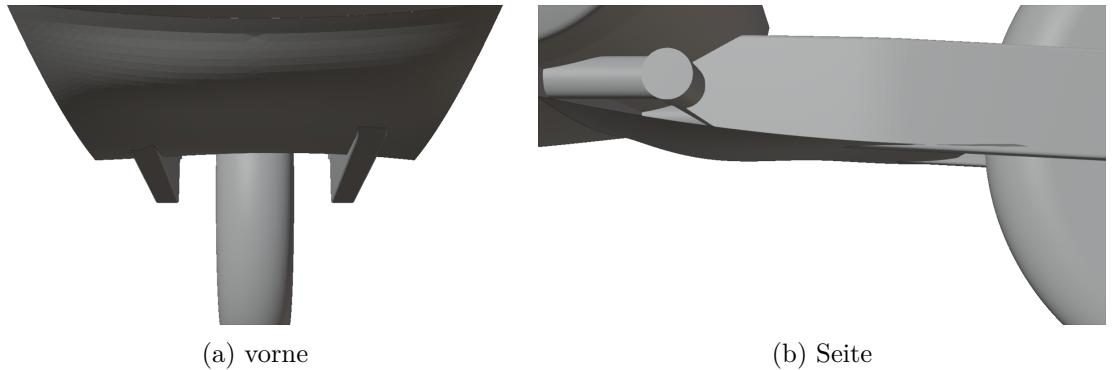


Abb. 50: Die Phänotypen des Individuum (23-22), $c_D=0,1979$

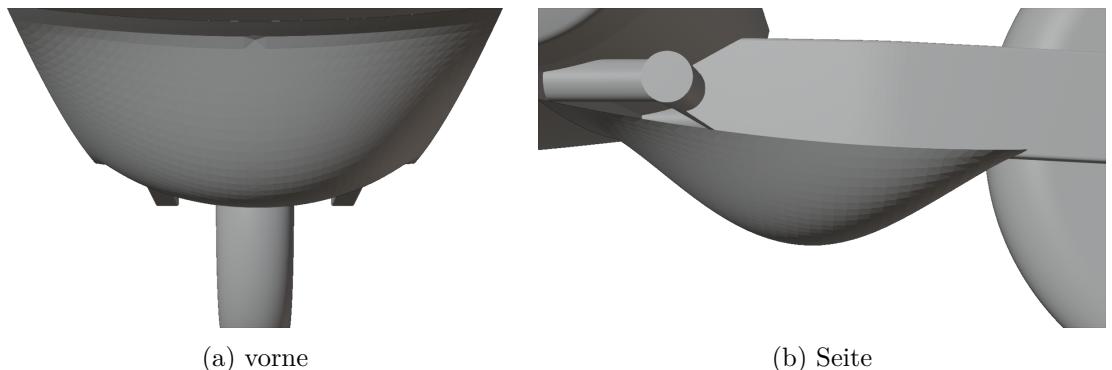


Abb. 51: Die Phänotypen des Individuum (9-16), $c_D=0,2091$

Punkts, der nur leicht unter dem restlichen Bauteil liegt, die Form nicht besonders stark. Dadurch fallen beide in die gleiche Lösungsklasse.

Dies zeigt eine Problematik der gewählten Kategorien auf. Da die Position des tiefsten Punktes bei einer Platte keinen wirklichen Einfluss auf den erzeugten Phänotypen hat, werden hier am rechten Rand eine Reihe von Lösungen entwickelt, die am Ende keine wirkliche phänotypische Diversität aufweisen. Dies bezieht sich zwar nur auf den rechten Rand und weiter links entstehen phänotypische Unterschiede zwischen Individuen in einer Spalte, allerdings sind die Spalten, weiter links in der Karte immer spärlicher gefüllt. Das heißt, sowohl zum linken Rand als auch zum rechten Rand der Karte nimmt die Diversität der Spalten aus unterschiedlichen Gründen ab. Das ist ein klares Indiz, dafür, dass die x-Position des tiefsten Punktes keine gute Kategorie darstellt und vermutlich durch eine andere Kategorie, die in der Lage ist, mehr Diversität zu erzeugen ausgetauscht werden sollte. Auch ist die Anzahl an verschiedenen Lösungsklassen gering. Die Lösungen lassen sich hauptsächlich in nach hinten offene, nach hinten geschlossene und flache Lösungen aufteilen.

Eingangs wurde für das Bauteil die Hypothese **H3** aufgestellt, dass symmetrische Bauteile die besten Lösungen für den symmetrischen E-Roller darstellen. Alle vom Algorithmus generierten Lösungen sind nahezu symmetrisch. Die Lösungen sind so nah an einer symmetrischen Lösung, dass die Asymmetrie nicht auf einen expliziten Vorteil dieser hindeutet, sondern dass die Erzeugung perfekter symmetrischer Bauteile bei 20 Freiheitsgraden aufwendig für den evolutionären Algorithmus ist und deshalb nur fast symmetrische Bauteile entstehen. Besonders in Anbetracht der Tatsache wie viele asymmetrische Lösungen theoretisch möglich sind, stellt die Erzeugung von nur praktisch symmetrischen Teilen einen Trend dar. Das war zwar grundsätzlich zu erwarten, da der E-Roller symmetrisch ist und damit die Erwartung besteht, dass Strömungen um diesen auch symmetrisch sind. In

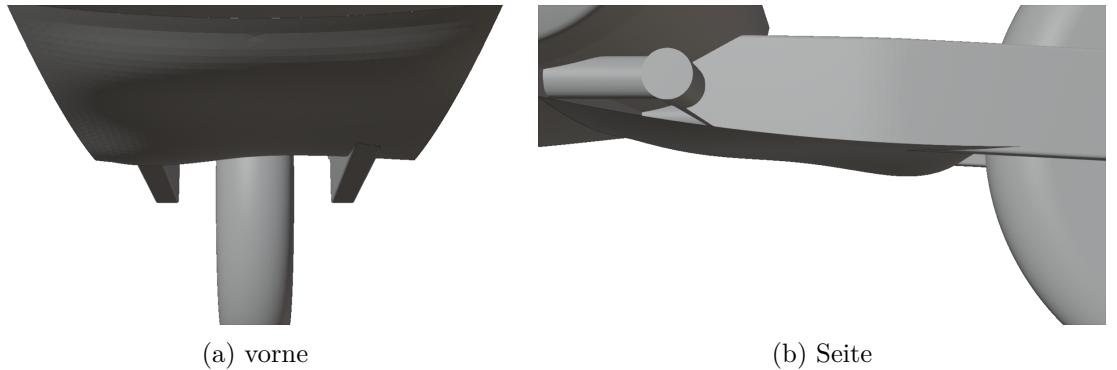


Abb. 52: Die Phänotypen des Individuums (20-5), $c_D=0,1970$

solchen Strömungen wären Bauteile die symmetrisch sind erwartungsgemäß optimal. Die Ergebnisse bestätigen die aufgestellte Hypothese also weitestgehend.

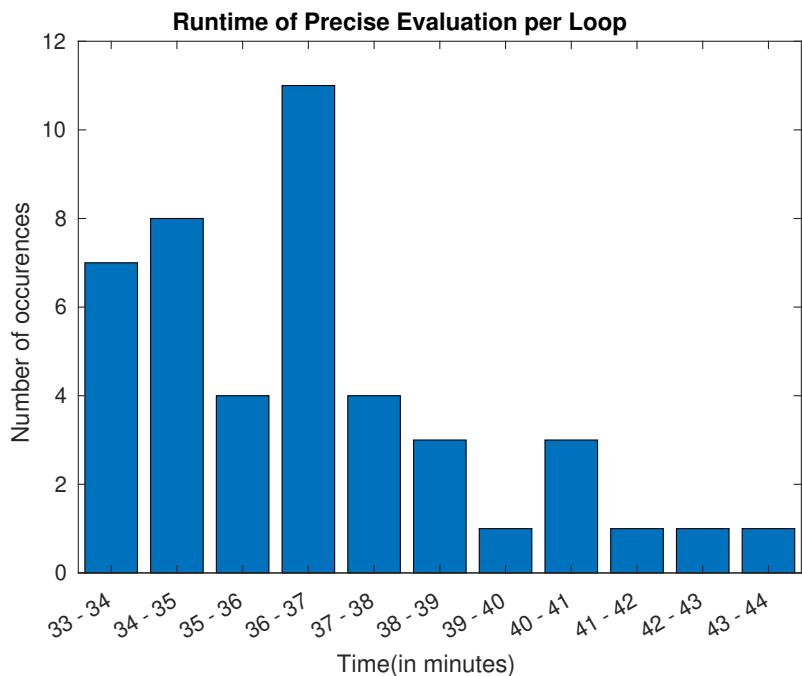


Abb. 53: Laufzeiten zur parallelen Evaluation von 10 präzisen Evaluationen

Ein nicht unwichtiger Teil in der E-Roller-Domäne, der zwar nur indirekt etwas mit SAIL zu tun hat, sind die Openfoam-Simulationen. Besonders auffällig war, dass diese in der E-Roller-Domäne wesentlich rechenaufwendiger ausfielen. In Abb. 53 sind die Laufzeiten der präzisen Auswertungen pro Schleifeniteration dargestellt. Insgesamt wurden in jeder der 44 Schleifeniterationen 10 Individuen⁸ parallel ausgewertet. Die dargestellten Zeiten sind nicht nur die für die Openfoam, sprich Zerlegung snappyHexMesh, Simulation etc., benötigten Zeiten, sondern beinhalten einen Overhead von Matlab, wie die Datenverteilung auf 10 Worker und die Fitnessberechnung. Nichtsdestotrotz stellen die Openfoam-Funktionen den größten Teil des Zeitbedarfs dar. Die beträchtliche Laufzeit, die für Openfoam benötigt wird erzeugt eine Obergrenze an präzisen Funktionsauswertungen die durchgeführt werden können.

⁸500 Gesamtsamples - 60 Initialsamples = 440 Akquiresamples

5 Diskussion

5.1 Radkasten

In dieser Arbeit wurden drei Experimente auf der Radkastendomäne durchgeführt. Im ersten Experiment konnte gezeigt werden, dass SAIL grundsätzlich auf die Domäne anwendbar ist. Es konnte validiert werden, dass die Freiformdeformation der Radkästen sinnvoll deформierte Radkästen, d. h. Radkästen die nahtlos an das restliche Velomobil anschließen, produzieren kann. Die Hypothese, dass die Breite des Velomobils negativ mit dem Luftwiderstand des Velomobils korreliert ist, konnte bestätigt werden. Ebenfalls konnte die Hypothese, dass der Constraint bei breiteren Velomobilen einfacher zu erfüllen ist, bestätigt werden. Auch wenn der Constraint in der Variante ohne Constraint teilweise schon erfüllt wurde, besonders bei breiten Lösungen hat das Experiment gezeigt, dass die explizite Einbindung des Constraints zu einer nennenswerten Verbesserung diesbezüglich führt. Zudem konnte gezeigt werden, dass erhebliche Constraintverbesserungen durch relativ kleine Fitnesseinbußen möglich sind.

Mit diesem Wissen wurde das zweite Experiment durchgeführt, in dem die Anzahl an Freiheitsgraden dramatisch erhöht wurde. Die Hoffnung war durch mehr Freiheitsgrade mehr mögliche Deformationen zu erlauben um interessantere Bauteile generieren zu können. Durch die Erhöhung der Freiheitsgrade wurde die Karte wesentlich besser gefüllt als dies noch beim ersten Experiment der Fall war. Es ist herauszustellen, dass der Abstand der Constraintwerte zwischen den beiden Versionen vom ersten zum zweiten Experiment angestiegen ist. Besonders da im zweiten Experiment mehr breitere Lösungen in der finalen Karte erhalten waren, ist diese Tendenz interessant. In beiden Versionen wird die Karte stärker gefüllt. Die zusätzlichen Individuen sind vorrangig stärker deformiert, was zu einer größeren Spannweite an erreichten Luftwiderstandswerten in beiden Versionen führt.

Um sicherzustellen, dass MAP-Elites vollständig konvergiert ist wurde dieses Experiment noch einmal mit erhöhten Laufzeiten durchgeführt. Im dritten Experiment konnte zuerst einmal festgestellt werden, dass sich die Constraintfüllung bei erhöhter Laufzeit in der Version ohne Constraint sogar etwas verschlechtert. Insgesamt war der Unterschied zwischen dem zweiten und dritten Experiment allerdings so klein, dass gesagt werden kann, dass das zweite ausreichen konvergiert ist. Die Analyse der genetischen Diversität der Akquiseindividuen zeigt, dass Deformationen in y und z-Richtung für gute Lösungen wichtiger sind, da in diesen Dimensionen Exploitation in engen Bereichen stattfindet, während Deformationen in x-Richtung eher unwichtig scheinen. Die genetische Diversität ist in beiden Varianten ähnlich groß und der genetische Suchraum wird ausreichend abgedeckt. Allerdings findet eine Verschiebung des Fokus von der Variante ohne Constraint zu der mit Constraint statt. Es werden beispielsweise mehr Individuen mit stärkeren y-Deformationen evaluiert.

Die beiden Eingangs formulierten Hypothese, dass es breiteren Velomobilen einfacher fällt den Constraint zu erfüllen (**H1**) und dass breitere Velomobile mit schlechteren Luftwiderstandswerten korreliert sind (**H2**) konnten verifiziert werden. Interessant ist, dass der Effekt bezüglich des Constraints abnimmt, wenn dieser explizit behandelt wird, aber immer noch vorhanden ist. Ähnliches war auch bei der zweiten Hypothese zu beobachten. In der Variante ohne Constraint ist die Korrelation zwischen Luftwiderstand und Breite des Velomobils ist sehr stark, während diese in der Variante mit Constraint wesentlich schwächer ausfällt.

In der phänotypischen Diversität fallen zwei Dinge auf. Erstens werden die Karten in der Variante mit Constraint nicht so weitreichend gefüllt wie die ohne. Das hängt damit zusammen, dass die Einführung des Constraints ein zusätzliches Ziel zur Exploitation⁹, wodurch

⁹Da der Constraint an jedem Punkt berechnet wird und rauschfrei ist, besitzt der Constraint eine Varianz von 0

sich die Balance zwischen Exploitation un Exploration verschiebt. Dies ist auch erkennbar, wenn man überprüft , wo die Akquiseindividuen in der gewählten phänotypischen Karte liegen. In der Variante mit Constraint ist eine Häufung von Akquiseindividuen im mittigen Bereich der Karte zu erkennen. Der mittige Bereich der Karte ist gerade der Bereich, in dem die größten Verbesserungen bezüglich der Constrainterfüllung zu finden waren. Für schmalere Individuen ist es schwerer den Constraint zu erfüllen, während breitere Lösungen aerodynamisch schlechtere Eigenschaften aufweisen. Deshalb findet Exploitation hauptsächlich in diesem Bereich statt.

Bei der Betrachtung der produzierten Lösungen ist eine Verbesserung des Constraints erkennbar, allerdings ist diese nicht für die ganze Karte gleich stark. Da breitere Individuen in der Variante ohne Constraint schon dazu tendieren den Constraint zu erfüllen ist dort nicht viel Raum für Verbesserung. Trotzdem werden auch breite Individuen leicht verbessert. Schmalere Individuen hingegen verzeichnen auch keine starken Verbesserungen, obwohl diese hier möglich sein sollten. Dies hängt mit der Problematik der Gewichtung bei der Nutzung eines Soft Constraints zusammen. Die Gewichtung ist dafür verantwortlich den gewollten Kompromiss zwischen Constraint und Aerodynamik zu bestimmen. Bei schmaleren Individuen führte die Gewichtung dazu, dass zwar kleine Verbesserungen stattfanden, größere Constraintverbesserungen allerdings durch die damit verbundenen Kosten in der Aerodynamik vermieden wurden. Dadurch ähneln sich die Phänotypen der schmaleren Individuen zwischen beiden Varianten sehr stark und es sind dort auch keine großen Constraintverbesserungen zu verzeichnen. Wo tatsächliche gute Constraintverbesserungen erreicht wurden, ist der mittlere Bereich der Karte.

Interessant ist auch, dass viele Individuen existieren, die den Constraint fast erfüllen. Kleine Reste des Constraintvolumens stoßen noch aus dem Velomobil hervor. Höchstwahrscheinlich hängen diese kleinen Restvolumen mit der Vereinfachung des Radkastens zusammen. Um die Constraintberechnung ausreichend schnell durchführen zu können wurde der Radkasten vereinfacht. Der vereinfachte Radkasten und der originale Radkasten decken sich dadurch nicht, wodurch, besonders wenn diese deformiert werden, kleinere Ungenauigkeiten entstehen können. Dadurch werden diese Teilvolumen in der Constraintberechnung nicht mehr beachtet, da sie außerhalb des deformierten vereinfachten Radkastens liegen.

5.2 E-Roller

Für den E-Roller wurde in dieser Arbeit nur ein Experiment durchgeführt. Aus dessen Ergebnisse können trotzdem einige Schlussfolgerungen gezogen werden.

Zuerst muss klargestellt werden, dass die berechneten Lösungen aufgrund der Unsicherheit, die mit der Korrektheit der Openfoam-Simulationen zusammenhängt, nicht als absolute Wahrheit behandelt werden dürfen. Alle Beobachtungen, die auf Basis dieses Experiments getätigt wurden, benötigen weitere Verifikation der Korrektheit der Simulationen. Trotzdem können einige Dinge an diesem Experiment analysiert werden und für eventuelle zukünftige Experimente genutzt werden.

Die erste Beobachtung, die an diesem Experiment gemacht werden konnte, ist, dass die Anwendung von SAIL auf diese Problemdomäne möglich ist. Es konnte erfolgreich ein Surrogatmodell trainiert werden auf dessen Basis MAP-Elites durchgeführt werden konnte. Es konnte eine Karte aus phänotypisch diversen Lösungen generiert werden, auch wenn diese Karte nur wenige Lösungsklassen enthielt. Es kann nicht abschließend gesagt werden , ob für das Problem nur Lösungen aus diesen Lösungsklassen brauchbar sind, oder ob die Limitierung auf diese Lösungsklassen ein Produkt der Konfiguration ist. Auch ist es möglich, dass durch die Fehlkonfiguration von Openfoam Effekte, die nicht-triviale Lösungen haben könnten überhaupt nicht gemessen werden können, oder von der Stärke der Oszillation überdeckt werden.

Es konnte festgestellt werden, dass praktisch nur symmetrische Bauteile generiert wurden. Damit konnte die Hypothese **H3**, dass symmetrische Bauteile für den symmetrischen E-Roller präferiert werden, bestätigt werden.

Es wurde grundsätzlich die Tendenz beobachtet, dass flachere Bauteile bessere Luftwiderstandswerte aufweisen. Dieses Ergebnis ist zwar etwas ernüchternd, da die Hoffnung bestand außergewöhnliche Bauteile zu generieren, die bessere aerodynamischen Eigenschaften aufweisen, als ein sehr banales. Allerdings gilt wie bereits erwähnt, dass die Ungenauigkeit der Openfoam-Simulation kleine Effekte, die durch anders förmige Bauteile entstehen würden schlicht ignorieren kann.

6 Ausblick

6.1 Radkasten

In der Arbeit konnte gezeigt werden, dass eine Einbindung eines Constraints in ein divergentes Optimierungsverfahren möglich ist.

In der Arbeit wurden nur zwei FFD-Konfigurationen genutzt. Die erste war sehr einfach um an dieser zu testen, dass der Algorithmus die richtigen Tendenzen aufzeigt. Nachdem das erfolgt war, wurde eine komplexere FFD-Konfiguration mit Freiheitsgraden gewählt, um mehr Deformationen zu erlauben. Selbst diese erlaubt allerdings nur begrenzt viele Deformationen. Es wäre sinnvoll die Anzahl an Freiheitsgraden weiter zu steigern und besonders auf die Punkte zu achten, die wichtig für Lösungen sind. So wurde in der Arbeit festgestellt, dass Deformationen in y, wie erwartet sehr wichtig sind, aber auch, dass auch Deformationen in z wichtig sind. Deformationen in x-Richtung waren weitergehend unwichtig und stellten damit Dimensionen dar, die keine interessanten zusätzlichen Deformationen erlaubten. Es sollt also auch vermieden werden alle Punkte in allen Dimensionen zu deformieren und es sollte sich stattdessen auf y-Deformationen in allen und z-Deformationen in den oberen und unteren Rändern fokussiert werden.

Ein Problem, welches sich in dieser Arbeit gezeigt hat, ist das Problem der korrekten Gewichtung von Luftwiderstand und Constraint. Das Problem der Gewichtung bei der Nutzung von Soft Constraints ist ein häufig auftretendes und schwer zu lösendes Problem. Stattdessen könnte in Erwägung gezogen werden den Constraint, statt Strafwert als zusätzliche Feature-Dimension aufzunehmen. Dadurch würde die problematische Gewichtung zwischen Luftwiderstand und Strafwert wegfallen und es würde eine Diversität an Lösungen entlang der Constrainterfüllung generiert. Damit könnte man garantieren, dass Lösungen verschiedener Constrainterfüllung generiert werden und es könnte eine Untersuchung stattfinden wo überhaupt Lösungen generiert werden, die den Constraint erfüllen, da diese nicht wie in der momentanen Version von aerodynamisch besseren Lösungen die den Constraint schlechter erfüllen verdrängt werden können.

Es sollte außerdem eine Validierung gegen den realen Constraint stattfinden. Der tatsächliche Constraint bestand darin einen größeren Radausschlag zu ermöglichen. In der Arbeit wurde immer nur mit dem Volumen des Radausschlags gearbeitet. Dies war eine Vereinfachung um den Constraint schnell und effizient berechnen zu können. Zwar ist anzunehmen das ein kleineres Außenvolumen mit einem größeren Radausschlag korreliert ist, können zahlreiche Beispiele konstruiert werden, in denen das Außenvolumen minimiert wird und dabei kein größerer Radausschlag erreicht wird. Den Constraint so zu formulieren wie es in dieser Arbeit geschehen ist, ist ein vertretbarer Ansatz, die produzierten Ergebnisse sollten aber auf die Erfüllung des tatsächlichen Constraints geprüft werden.

Zur Formulierung des Constraint wurde hier einige Vereinfachungen vorgenommen. Es traten immer wieder kleine Restvolumen auf, die aus dem Velomobil herausstachen. Diese auftretenden Restvolumen entstanden durch die Vereinfachung des Radkastens, bei der nicht garantiert war, dass sich das Volumen nur vergrößern kann. Es wäre sinnvoll eine

bessere Vereinfachung zu wählen, sodass diese niemals kleiner als der reale Radkasten ist, um solche Restvolumen zu vermeiden.

Gegebenenfalls ist die Untersuchung ob und wie gut ein Surrogatmodell auch für die Berechnung des Constraints genutzt werden kann. Selbst mit den getroffenen Vereinfachungen nahm die Berechnung des Constraints einen großen Teil der Laufzeit in Anspruch, der vergleichbar mit der Laufzeit der präzisen Evaluationen war. Es war in dieser Arbeit möglich den Constraint direkt zu berechnen, sollte dieser aber komplexer werden, dann wird die direkte Berechnung sehr bald nicht in annehmbarer Zeit durchzuführen sein. Da sowieso schon ein Surrogatmodell für den Luftwiderstand genutzt wird, liegt die Erweiterung und die Nutzung eines zweiten Surrogatmodells für den Constraint nahe. Sollte dies möglich sein würde das die restriktive obere Komplexitätsgrenze der Constraintberechnung aufheben, wodurch realitätsgetreuere Constraints ermöglicht werden könnten.

Auch wäre eine saubere Definition des Constraintvolumens nötig. In der genutzten Definition ist der Bereich des Radausschlags unterhalb des Velomobils Teil des Constraintvolumens. Dadurch entspricht der vollständige ermöglichte Radausschlag nicht dem Constraintvolumen von null. Auch schränkt dieser Teil die Bewegungsfreiheit der Räder nicht ein. In weiterführenden Arbeiten sollte der Constraint soweit besser formuliert werden, dass dieser Teil kein Teil des Constraintvolumens ist um einen sauberen Nullpunkt zu erhalten.

Auch sollte über die Parametrisierung des Gaußprozesses nachgedacht werden. Besonders die isotropische Kovarianzfunktion sollte gegen eine mit Automatic relevance determination ausgetauscht werden, da es denkbar ist, dass Dimensionen, die auf y-Deformationen abgebildet werden andere Längenmaße benötigen, als solche, die auf z-Deformationen abgebildet werden.

Die hier durchgeführten Experimente wurden mit einem vereinfachten Modell des Velomobils, welches die Räder nicht enthält, berechnet. Da die Radkästen mit hoher Wahrscheinlichkeit Einfluss auf die Turbulenzen um die Räder des Velomobils haben werden, wären Experimente, die die Räder des Velomobils einschließen wünschenswert. Dies könnte zwar die benötigte Auswertungszeit der Simulationen erhöhen, aber vermutlich realistischere Ergebnisse generieren.

6.2 E-Roller

Aufgrund der Schwierigkeiten der Anwendung von SAIL auf die E-Rollerdomäne lag der Fokus der Untersuchung dieser Arbeit in der Radkastendomäne. Das Hauptproblem der in dieser Arbeit erzeugten Ergebnisse ist die Unsicherheit bezüglich deren Korrektheit. Der Grund dafür ist der Fakt, dass die c_D -Werte in der gewählten Konfiguration OpenFoam nicht konvergierten. Der Grund dafür kann eine fehlerhafte Parametrisierung Openfoams sein, die Anpassung dieser Konfiguration sprengt allerdings den Umfang dieser Arbeit. In zukünftigen Untersuchungen sollte also, vor der Nutzung von SAIL, zuerst sichergestellt werden, dass die zugrundeliegende Simulation in Openfoam realitätsgetreue Ergebnisse liefert. Ist das nicht garantiert können alle möglichen Ergebnisse von SAIL nur beschränkte Aussagekraft haben.

Des Weiteren sind die erzeugten Formen für das Bauteil nicht besonders kreativ oder bahnbrechend. Das kann daran liegen, dass für dieses Bauteil keine guten unintuitiven Lösungen existieren. Die Anzahl an möglichen Lösungen wird allerdings durch die gewählte Deformationskonfiguration eingeschränkt. Es sollte möglicherweise über andere FFD-Konfigurationen, beziehungsweise Deformatiormationen in anderen Punkten und/oder Richtungen nachgedacht werden. Besonders, dass die seitlichen Ränder des Bauteils in der hier gewählten Konfiguration nicht deformiert werden können, stellt eine starke Einschränkung dar. Auch sollte in Erwägung gezogen werden die linke und rechte Hälfte symmetrisch zu verformen. Erstens ist anzunehmen, dass dadurch, dass der E-Roller an

sich symmetrisch ist die Strömungen, die das Bauteil treffen ebenso symmetrisch sind. Lösungen die in solchen Strömungen am besten abschneiden werden mit hoher Wahrscheinlichkeit symmetrisch sein. Im Experiment konnte auch beobachtet werden, dass alle erzeugten Bauteile fast symmetrisch waren. Alle Asymmetrien waren meist leicht und eher auf die Größe des genetischen Suchraums zurückzuführen, als auf einen Vorteil durch eine asymmetrische Bauart. Auch ist fragwürdig ob asymmetrische Bauteile überhaupt in die Designparameter des E-Rollers passen. Ästhetik ist schwer quantifizierbar und Fertigungstauglichkeit sowie Einbettbarkeit des Bauteils in den E-Roller wurden in dieser Arbeit nicht untersucht, alle drei könnte aber von vornherein gegen asymmetrische Bauteile sprechen. Durch die symmetrische Verformung des Bauteils könnte allerdings die Anzahl an benötigten Freiheitsgraden halbiert, beziehungsweise mit der gleichen Anzahl an Freiheitsgraden feinere Deformationen erlaubt werden. In Anbetracht dieser drei Tatsachen scheinen symmetrische Verformungen bei denen die Verformungen der linken und rechten Hälfte des Bauteils aus dem gleichen Genom abgeleitet werden sinnvoller.

In dem durchgeföhrten Experiment waren die besten Ergebnisse bezüglich des Luftwiderstands solche die effektiv flach oder leicht nach oben deformiert waren. Es besteht die Möglichkeit, dass nach oben deformierte Bauteile noch optimaler sind. Dieser Lösungsraum sollte nach Lösungen untersucht werden. Allzu stark deformiert Lösungen wurden in dem durchgeföhrten Experiment durch die beschränkte minimale Deformation ausgeschlossen. Es wäre möglich mehr Deformationen nach oben zu erlauben, um solche Lösungen genauer zu untersuchen. Zwar muss dann eine Möglichkeit gefunden werden, dass solche Bauteile, den restlichen E-Roller nicht schneiden, sodass beispielsweise um dessen Bauteile herum deformiert wird. Dies könnte beispielsweise durch die Einföhrung eines Constraints geschehen, der testet, ob das Bauteil mit anderen Teilen des E-Rollers kollidiert.

Durch die Wahl des tiefsten Punkt als eines der Features war es selbst für leicht nach oben deformierte Bauteile schwer Einzug in die Karte zu erhalten. Deshalb sollte zusätzlich über die Kategorien nachgedacht werden, nach denen die Bauteile phänotypisch kategorisiert werden. Die gewählten Kategorien wurden hauptsächlich aus dem Grund gewählt, dass diese einfach berechenbar und klar verständlich sind. Es wäre vermutlich sinnvoll für die Domäne abgewägte Kategorien zu wählen und zu untersuchen wie sich dies auf die Lösungen auswirkt.

Zuletzt stellte die benötigte Laufzeit für OpenFoam trotz der Nutzung des vereinfachten E-Roller-Modells eine nicht unbeträchtliche Hürde dar. Dies ist vor allem auf die komplexere Form des E-Rollers im Vergleich zur aerodynamisch optimierten Projektilform des Velomobils und die längere Simulationsdauer von 1000 Sekunden zurückzuföhren. Eine qualitative Untersuchung welche Teile des E-Rollers wegrationalisiert werden können, ohne signifikante Ungenauigkeit in die Ergebnisse einzuföhren, könnte die Laufzeit reduzieren, wodurch beispielsweise wieder präzise Funktionsauswertungen ermöglicht werden, was mehr Freiheitsgrade erlauben kann.

7 Literaturverzeichnis

Alec Jacobson and others 2018

ALEC JACOBSON AND OTHERS: *gptoolbox: Geometry Processing Toolbox*. <https://github.com/alecjacobson/gptoolbox>. Version: 2018

Anderson 2017

ANDERSON, John D.: *Fundamentals of aerodynamics*. 6th edition. New York, NY : McGraw-Hill Education, 2017 (McGraw-Hill series in aeronautical and aerospace engineering). – ISBN 9781259251344

Auer 2002

AUER, Peter: Using Confidence Bounds for Exploitation-Exploration Trade-offs. In: *Journal of Machine Learning Research* 3 (2002), S. 397–422. <http://dx.doi.org/10.1162/153244303321897663>. – DOI 10.1162/153244303321897663

Caretto et al. 1973

CARETTO, L. S. ; GOSMAN, A. D. ; PATANKAR, S. V. ; SPALDING, D. B.: Two calculation procedures for steady, three-dimensional flows with recirculation. Version: 1973. <http://dx.doi.org/10.1007/BFb0112677>. In: CABANNES, Henri (Hrsg.) ; TEMAM, Roger (Hrsg.): *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics* Bd. 19. Berlin and Heidelberg : Springer, 1973. – DOI 10.1007/BFb0112677. – ISBN 978-3-540-06171-7, S. 60–68

Duvenaud 2014

DUVENAUD, David: *Automatic model construction with Gaussian processes*, Diss., 2014

Gaier et al.

GAIER, Adam ; ASTEROTH, Alexander ; MOURET, Jean-Baptiste: *Data-Efficient Design Exploration through Surrogate-Assisted Illumination*. <http://dx.doi.org/10.1162/evco{ }a{ }00231>

Jin 2011

JIN, Yaochu: Surrogate-assisted evolutionary computation: Recent advances and future challenges. In: *Swarm and Evolutionary Computation* 1 (2011), Nr. 2, S. 61–70. <http://dx.doi.org/10.1016/j.swevo.2011.05.001>. – DOI 10.1016/j.swevo.2011.05.001. – ISSN 2210-6502

Lehman und Stanley 2011

LEHMAN, Joel ; STANLEY, Kenneth O.: Abandoning objectives: evolution through the search for novelty alone. In: *Evolutionary computation* 19 (2011), Nr. 2, S. 189–223. <http://dx.doi.org/10.1162/EVC0{ }a{ }00025>. – DOI 10.1162/EVC0_a_00025

Mahfoud 1996

MAHFoud, Samir W.: *Niching Methods for Genetic Algorithms*. USA, Diss., 1996

Meagher 1982

MEAGHER, Donald: Geometric modeling using octree encoding. In: *Computer Graphics and Image Processing* 19 (1982), Nr. 2, 129–147. [http://dx.doi.org/10.1016/0146-664X\(82\)90104-6](http://dx.doi.org/10.1016/0146-664X(82)90104-6). – DOI 10.1016/0146-664X(82)90104-6. – ISSN 0146-664X

Mouret und Clune

MOURET, Jean-Baptiste ; CLUNE, Jeff: *Illuminating search spaces by mapping elites*. <http://arxiv.org/pdf/1504.04909v1>

OpenCFD

OPENCFD: *OpenFOAM® - Official home of The Open Source Computational Fluid Dynamics (CFD) Toolbox.* <https://www.openfoam.com/>

OpenMPI

OPENMPI: *Open MPI: Open Source High Performance Computing.* <https://www.open-mpi.org/>

PREEN und BULL 2016

PREEN, Richard J. ; BULL, Larry: Design Mining Interacting Wind Turbines. In: *Evolutionary computation* 24 (2016), Nr. 1, S. 89–111. http://dx.doi.org/10.1162/EVCO{_}a{_}00144. – DOI 10.1162/EVCO_a_00144

Rasmussen und Williams 2008

RASMUSSEN, Carl E. ; WILLIAMS, Christopher K. I.: *Gaussian processes for machine learning.* 3. print. Cambridge, Mass. : MIT Press, 2008 (Adaptive computation and machine learning). – ISBN 9780262182539

Robbins 1952

ROBBINS, Herbert: Some aspects of the sequential design of experiments. In: *Bulletin of the American Mathematical Society* 58 (1952), Nr. 5, S. 527–536. <http://dx.doi.org/10.1090/S0002-9904-1952-09620-8>. – DOI 10.1090/S0002-9904-1952-09620-8. – ISSN 0002-9904

Sederberg und Parry 1986

SEDERBERG, Thomas W. ; PARRY, Scott R.: Free-form deformation of solid geometric models. In: EVANS, David C. (Hrsg.): *Proceedings of the 13th annual conference on Computer graphics and interactive techniques.* New York, NY : ACM, 1986. – ISBN 0897911962, S. 151–160

Shir 2012

SHIR, Ofer M.: Niching in Evolutionary Algorithms. Version: 2012. http://dx.doi.org/10.1007/978-3-540-92910-9{_}32. In: ROZENBERG, Grzegorz (Hrsg.) ; BÄCK, Thomas (Hrsg.) ; KOK, Joost N. (Hrsg.): *Handbook of natural computing* Bd. 8. Berlin : Springer, 2012. – DOI 10.1007/978-3-540-92910-9_32. – ISBN 978-3-540-92909-3, S. 1035–1069

Si 2015

SI, Hang: TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. In: *ACM Transactions on Mathematical Software* 41 (2015), Nr. 2, S. 1–36. <http://dx.doi.org/10.1145/2629697>. – DOI 10.1145/2629697. – ISSN 00983500

Simon 2013

SIMON, Dan: *Evolutionary Optimization Algorithms.* Hoboken : Wiley, 2013 <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=597827>. – ISBN 9780470937419

Sobol' 1967

SOBOL', I.M: On the distribution of points in a cube and the approximate evaluation of integrals. In: *USSR Computational Mathematics and Mathematical Physics* 7 (1967), Nr. 4, S. 86–112. [http://dx.doi.org/10.1016/0041-5553\(67\)90144-9](http://dx.doi.org/10.1016/0041-5553(67)90144-9). – DOI 10.1016/0041-5553(67)90144-9. – ISSN 00415553

Yu und Gen 2010

YU, Xinjie (Hrsg.) ; GEN, Mitsuo (Hrsg.): *Decision Engineering.* Bd. 0: *Introduction to Evolutionary Algorithms.* London : Springer-Verlag London, 2010.

<http://dx.doi.org/10.1007/978-1-84996-129-5>. <http://dx.doi.org/10.1007/978-1-84996-128-8> – ISBN 978-1-84996-128-8