

# Evaluation des genetischen Algorithmus

Alexander Binsmaier, Manuel Mende, Yaroslav Direy

12. Dezember 2016

## Inhaltsverzeichnis

<b>1</b>	<b>Manuell erzeugte Testfälle</b>	<b>1</b>
<b>2</b>	<b>Automatisch erzeugte Testfälle</b>	<b>3</b>
2.1	Konvergenz und statistisches Verhalten . . . . .	3
2.2	Äquivalenzklassen . . . . .	3
<b>3</b>	<b>Güte der automatischen Testfälle</b>	<b>5</b>
<b>4</b>	<b>Verbesserungsmöglichkeiten</b>	<b>5</b>

## 1 Manuell erzeugte Testfälle

Um die vom genetischen Algorithmus erzeugten Testfälle bewerten zu können, werden Testfälle zum Vergleich benötigt. Einige sinnvolle Testfälle werden in Tabelle 1 präsentiert. Die ersten zwei Testfälle starten sehr weit außen. Sie führen zur Kollision mit einem Hindernis, dennoch ist die Fitness der Chromosome schlecht. Die zweiten Testfälle starten etwas näher an der Parklücke. Die Fahrspur führt eng an den Rändern der engen Parklücke vorbei, es ergibt sich ein entsprechend hoher Fitnesswert. Der fünfte Fall startet direkt Richtung Wand. Dadurch bedingt kommt es zur Kollision. Auch hier ist der Fitnesswert gering. Fall sechs und sieben starten, als wäre die Parklücke auf der linken Seite – also um 180 Grad gedreht. Die hier abgefahrene Spur ist überflüssig komplex. Der Parkassistent versucht ein Rechtsszenario zu realisieren. Im ersten Fall kommt es sogar zur Kollision, trotz riesiger Parklücke. Dennoch ist die Fitness sehr klein. Abbildung 1 zeigt die Gesamtheit der manuell ausgewählten Testfälle.

Position	Winkel	Parklücke	Fitness	Reale Eignung
5/0	0	2.5/1	0.0	Kollision
-5/0	0	2.5/1	0.0	Kollision
4/0	0	2.5/1	0.039828	Beinahe Kollision
-4/0	0	2.5/1	0.039828	Beinahe Kollision
2/0	270	2.5/1	0.0	Kollision
2.5/0	180	5/2	0.0	Falsches Ziel und Kollision
-2.5/0	180	5/2	0.008106	Falsches Ziel
0/4	0	2.5/1	0.020079	Merkwürdige Fahrspur
0/0	0	5/2	0.006673	Merkwürdige Fahrspur

Tabelle 1: Manuell gewählte Testfälle

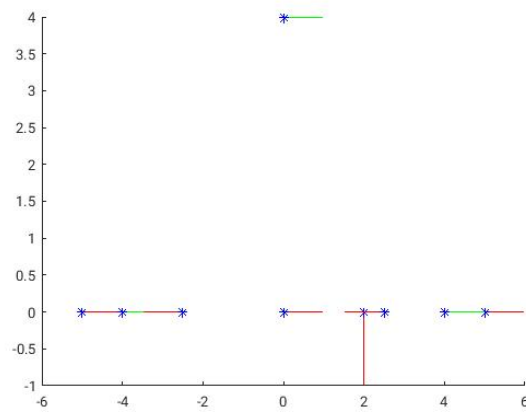


Abbildung 1: Visualisierung der Testfälle

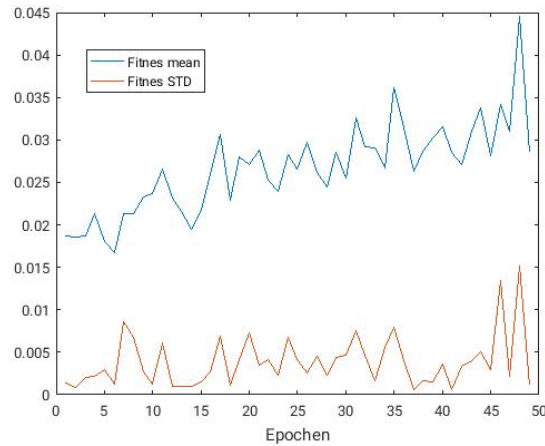


Abbildung 2: Entwicklung der Fitness

## 2 Automatisch erzeugte Testfälle

Der genetische Algorithmus erzeugt Testfälle. Um die Güte der Testfälle beurteilen zu können, sollen zwei Dimensionen analysiert werden. Zum einen wie sinnvoll die ermittelten Testfälle sind und zum anderen wie deterministisch die Ermittlung abläuft.

### 2.1 Konvergenz und statistisches Verhalten

Für das Konvergenz-Verhalten ist es sinnvoll zu analysieren, wie stark sich der Fitnesswert durch mehr Generationen verbessert. Dazu wird die Anzahl simulierter Epochen Schrittweise erhöht. Abbildung 2 zeigt einen aufgezeichneten Simulationsverlauf. Die Generationen wurden in Zehnerschritten erhöht und der Mittelwert über alle Chromosome der Endgeneration verglichen. Es kann eine steigende Tendenz festgestellt werden. Je mehr Generationen simuliert werden, desto höher ist der resultierende Wert der Gesamt-Fitness.

### 2.2 Äquivalenzklassen

Äquivalenzklassen erlauben es, Aussagen über die Ähnlichkeit von Testfällen zu machen. Die Tabelle 2 listet die Mittelwerte sowie Standardabweichungen für 100 Chromosome die über die gegebene Anzahl an Epochen durch Evolution ermittelt wurden. Es lässt sich eine Tendenz zu kleinen Parklücken feststellen, vor allem die Länge der Parklücke wird über die Epochen reduziert. Auch die Abweichung ist gering. Die Position des Fahrzeugs streut in x-Richtung sehr stark, das gleiche gilt für die Orientierung des Fahrzeugs.

50 Epochen	Mittelwert	Abweichung
X-Position	0.515882	2.301561
Y-Position	0.888431	0.934355
Orientation	2.456849	2.272890
Slot-Length	2.402382	0.164123
Slot-Depth	1.158000	0.125341
500 Epochen	Mittelwert	Abweichung
X-Position	0.352941	2.883735
Y-Position	-0.538824	0.420933
Orientation	3.691803	2.331866
Slot-Length	2.308667	0.092787
Slot-Depth	1.116588	0.148013

Tabelle 2: Gemittelte Testfälle

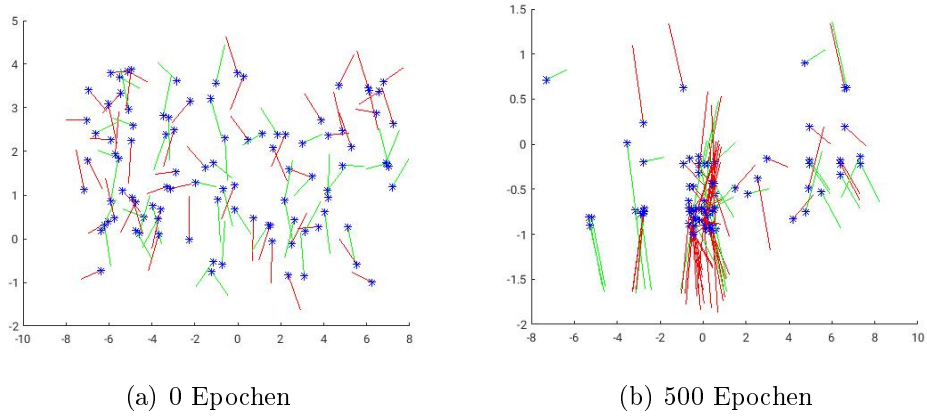


Abbildung 3: 100 Chromosome

Die y-Position des Fahrzeugs stellt sich auf negative Werte ein, auch hier ist eine Tendenz erkennbar. Abbildung 3 zeigt 100 Testfälle, die durch den genetischen Algorithmus ermittelt wurden. Wieder zeigen Rote Testfälle eine hohe Fitness, und grüne Testfälle eine Vergleichsweise geringe Fitness an. Abbildung 3(b) stellt die resultierenden Testfälle nach 100 Generationen dar. Dabei wird die Größe der Parklücke ausgenommen. Es wird deutlich, dass die Testfälle stark dazu tendieren, das Fahrzeug in der Mitte der Parklücke starten zu lassen. Die Orientierung spielt eine untergeordnete Rolle. Besonders beim Vergleich mit der zufälligen Initialbelegung in Abbildung 3(a).

### 3 Güte der automatischen Testfälle

Im Anschluss müssen die gefundene Testfälle beurteilt werden. Der Algorithmus scheint zu einer speziellen Art Testfälle – zentral zur Parklücke mit steilem Winkel – zu tendieren. Die manuell ermittelten Testfälle sind anders strukturiert.

Dennoch sind die Ergebnisse zufriedenstellend. Sie zeigen, dass der Genetische Algorithmus zu Chromosomen konvergiert, die das Fahrzeug teilweise in der Parklücke starten lassen, wodurch ein komplexes Einparkmanöver gefordert ist. Solche Manöver führen ein hohes Risiko mit sich, dass das System versagt und somit ein Mangel im Parkassistenten aufgezeigt wird.

Andererseits werden viele interessante Testfälle nicht gefunden. Im besonderen sollten Linksparkszenarien untersucht werden, da hier der Parkvorgang wesentlich effizienter und robuster gestaltet werden kann. Dies würde eine Modifikation der Fitness-Funktion voraussetzen. Auch werden Kollisionen nicht zufriedenstellend behandelt. Zwar sind einige Szenarien sicherlich auszuklammern, weil sie die Systemgrenzen übersteigen – wie direkt Richtung Wand zu starten, aber dennoch sollten Szenarien gut bewertet werden, in denen es nur knapp zu einer Kollision kommt. Diese Testfälle werden in der derzeitigen Implementierung unterbewusst unterdrückt.

### 4 Verbesserungsmöglichkeiten

Zum Abschluss werden nun noch Möglichkeiten gesucht, die derzeitige Implementierung zu erweitern.

Die ausschlaggebendsten Kritikpunkte an den automatisch generierten Testfällen sind die mangelnde Heterogenität und der Umgang mit Kollisionen. Einerseits ist das Ziel von genetischen Algorithmen, eine Population von möglichst optimalen Chromosomen zu generieren, ohne dabei andere Chromoso-

me in Betracht zu ziehen. Andererseits ist im Falle von Testfällen dennoch auch eine gewisse Diversität wünschenswert, da man mit einem einzelnen Testszenario nicht alle Features und möglichen Mängel eines so komplexen Systems wie einem Parkassistenten abdecken kann. Diesbezüglich müsste man eventuell die Fitnessfunktion so anpassen, dass sie zusätzlich die Einzigartigkeit eines guten Testfalles belohnt.

Im Bezug auf Kollisionen muss man das gesamte Einpark-Problem in Teilprobleme aufteilen. Eine Kollision zu einem späten Zeitpunkt im Einparkvorgang sollte als gut betrachtet werden. Eine Ausnahme in diesem Fall, sollte gemacht werden, wenn die Parklücke unzureichend groß ist, da die Bewertung der Parklücke in Bezug auf ausreichende Größe nicht vom Parkassistenten abgedeckt wird, und somit dieser Testfall ein anderes System testen würde. Für den Fall, dass eine Kollision zu einem frühen Zeitpunkt im Einparkvorgang auftritt muss differenziert werden, um welche Art von Systemversagen es sich hier handelt. Entsteht die Kollision nämlich aufgrund eines Implementierungs-Artefaktes, sodass der gewählte Einpark-Algorithmus nicht mit der Situation umgehen kann, sollte man nicht zwingend eine gute Bewertung geben. Andernfalls entstehen viele lokale Maxima, die man in dieser Menge nur schwer verlassen kann, obwohl die generierten Testfälle nur sehr wenige Features abdecken und nahezu trivial sind. Hier muss also eine Lösung gefunden werden, die Kollisionen nur bedingt positiv behandelt, aber auch nicht vollkommen ignoriert oder sogar Testfälle schlecht bewertet, nur weil sie zu einer Kollision führen.