Onboard-Evolution mit der Fahrenden Platine

Studienarbeit
von
Ferry Bachmann

Vorwort:

In der folgenden Arbeit sind einige Experimente mit einem kleinen fahrenden Roboter – der Fahrenden Platine – geschildert. Die Steuerung übernimmt ein Künstliches Neuronales Netz, dessen Gewichtungen aus einer künstlichen Evolution hervorgehen. Diese Evolution findet vollständig auf dem Roboter statt – eine so genannte „Onboard-Evolution“. Die hierbei gesammelten Erfahrungen stehen im Mittelpunkt der Arbeit und nicht das Ergebnis der Evolution.

Inhaltsverzeichnis

1. Technische Voraussetzungen

Basis der durchgeführten Experimente ist die „Fahrende Platine“ von Manfred Hild, ein von zwei Servos radgetriebener Roboter. Auf der Platine befindet sich eine mit 8 MHz getaktete Atmel AVR Atmega32 RISC-CPU mit 32KByte Flash-ROM für Programmdaten und einem ISP (In-System-Programmer). Der ISP wird über die Parallele Schnittstelle mit einem PC verbunden und ermöglicht die Programmierung der CPU ohne sie aus der Schaltung entfernen zu müssen. Als Arbeitsspeicher steht ein 2KByte SRAM und als nichtflüchtiger Speicher ein 1KByte EEPROM zur Verfügung dessen Inhalt über den ISP ausgelesen werden kann. Die Platine besitzt zwei SHARP-GP2D02 Infrarot-Abstandssensoren, deren 8Bit-Werte das Programm seriell über eine 1Bit-Leitung auslesen kann. An der Vorderseite der Platine sind zwei low-aktive Bump-Sensoren angebracht, also Schalter, die im „gedrückt“-Zustand eine 0 und sonst eine 1 auf die Leitung geben. Angetrieben wird die Platine durch zwei Servos, die über PWM-Register (Pulse-Width-Modulation) der CPU angesteuert werden. Im hinteren Teil sind zwei LEDs angebracht, die in den Experimenten gemeinsam mit einem Oszilloskop zu Diagnose-Zwecken genutzt wurden. Die Programmierung erfolgte in C mit einigen Inline-Assembler-Abschnitten. Das speziell für Atmel AVR RISC-CPUs frei erhältliche Entwickler-Werkzeug WinAVR stellt für diesen Zweck einen Editor und den GNU-C-Crosscompiler AVR-GCC zur Verfügung.

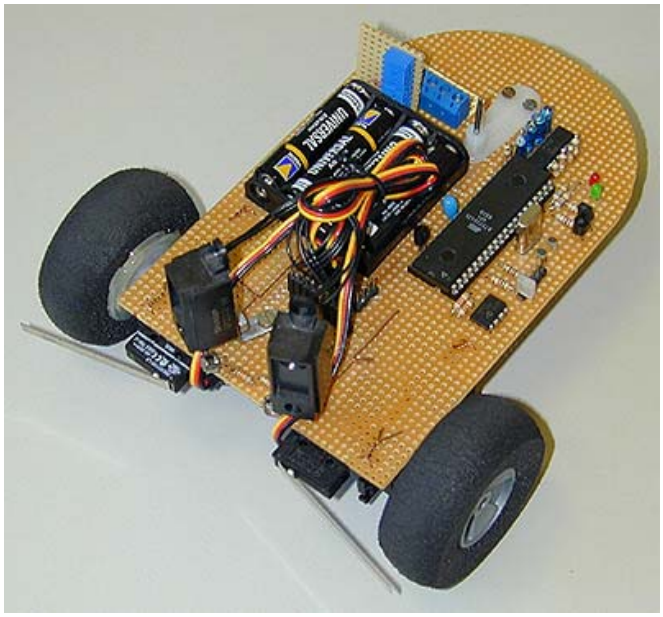


Abbildung 1-1 Die Fahrende Platine

2. Die Experimente

2.1. Einleitung

Ziel ist es über einen evolutionären Ansatz einen Roboter zu schaffen, der den ihm gegebenen Parcours mit möglichst hoher Geschwindigkeit durchfährt ohne gegen die Wände zu stoßen – die Hindernisvermeidung. Ursprünglich waren im Anschluss an diese einfache Aufgabe weitere Experimente zu komplexeren Aufgabenstellungen geplant. Aus der Hindernisvermeidung ergab sich aber bereits eine Vielzahl von Experimenten, so dass weitere Aufgabenstellungen den Rahmen dieser Arbeit gesprengt hätten.

Aus großen Kunststoff-Puzzle-Teile baute ich für die Experimente einen bunten 1.10m x 1.10m Laufstall mit Safttüten und Aktenordnern als Bandenbegrenzungen. Der Parcours wurde bewusst so gestaltet, dass der Roboter sich bei einer im genannten Sinne guten Strategie mal links und mal rechts drehen muss. Sensorik und Aktorik der Platine sind über ein Künstliches Neuronales Netz (KNN) mit fester Topologie verbunden. Da das Netz später nicht analysiert werden soll, ist nur das Folgende für das Verständnis wichtig:

Abbildung 2-1 Der erste Parcours

Die Werte der beiden Abstandssensoren sind auf -1 (≥40cm) bis +1 (0cm) als Ausgangsgrößen der Neuronen *Sl* und *Sr* abgebildet. Die Werte der Neuronen M_l und M_r werden an die entsprechenden Servos gegeben (-1 bei voller Fahrt rückwärts, +1 bei voller Fahrt vorwärts). Die acht Gewichte w_{ij} des KNNs sollen durch eine Künstliche Evolution aus den Experimenten hervorgehen. Da der Arbeitsspeicher und nichtflüchtige Speicher sehr knapp bemessen sind, wurden die Gewichte als 8Bit-Fixpunktzahlen kodiert, deren Wertebereich von $-7\frac{15}{16}$ bis $+7\frac{15}{16}$ geht. Der Suchraum der Künstlichen Evolution ist also:

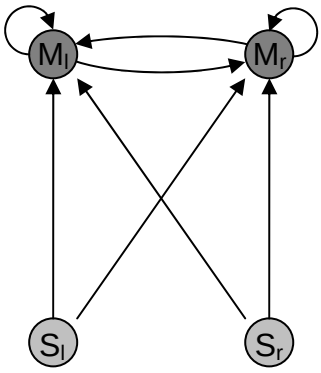


Abbildung 2-2 Das Künstliche Neuronale Netz

$$S = \left\{(x_1,\ldots,x_8) \mid x_i \in \{0,1\}^8\right\},\quad |S| = 2^{64}$$

Ein $x \in S$ wird in den Genetischen Algorithmen Genotyp (oder künstliches Chromosom) genannt. Zu jedem Genotyp gehört ein Phänotyp – der durch das KNN, dessen Gewichte im Genotyp kodiert sind, gesteuerte Roboter. Zu Beginn der Künstlichen Evolution werden 20 Genotypen (die Populationsgröße) zufällig erschaffen, d.h. gleichverteilt über $[-7\frac{15}{16}, +7\frac{15}{16}]$

für jedes Gewicht w_{ij}. Diese 20 Genotypen bilden die Generation Null. Die Phänotypen erhalten nacheinander eine Minute Zeit sich in dem Parcours zu bewegen.
Ihr Verhalten wird durch eine Funktion bewertet. Diese Bewertungsfunktion *Fit:* $S \rightarrow R$ ist die Zielfunktion der Suche (Holland[Holl75] unterscheidet zwischen Bewertungs- und Fitnessfunktion, hier soll darauf verzichtet werden). Nachdem alle Phänotypen der Generation Null bewertet wurden (nach 20 Minuten) werden auf Grundlage dieser Bewertung (der Fitness) Individuen selektiert. Die in diesem Experiment verwendete Strategie nennt sich „Truncation Selection", was bedeutet, dass die Individuen einer Generation nach ihrer Fitness sortiert werden und die besten M (hier 5) Individuen jeweils O (hier 4) Nachkommen erzeugen und somit $N=M \cdot O$ Individuen in die nächste Generation geben. Die N Individuen der Nachfolgegeneration werden zufällig untereinander gepaart und ein One-Point-Crossover durchgeführt, d.h. es wird ein Index innerhalb des Chromosoms zufällig bestimmt und ab diesem Index tauschen die Partner ihre Gene (also die Gewichte des KNNs) aus. Als letzter Operator der Genetischen Algorithmen kommt nun noch die Mutation ins Spiel. Hierbei wird für jedes Bit des Chromosoms eine Zufallszahl generiert und unterschreitet diese einen festen Wert so wird das entsprechende Bit gekippt (die Wahrscheinlichkeit für solch eine Mutation liegt in diesem Experiment bei 3%). Für die aus Selektion, Kreuzung und Mutation entstandene Generation Eins wird das Verfahren von vorne gestartet.
Die Fitness-Funktion des Experimentes hat die folgende Gestalt:

$$Fit = F_1 \cdot F_2 \cdot F_3$$

$$F_1 = \frac{1}{Cycles} \cdot \sum_{1 \le t \le Cycles} \frac{M_l(t) + M_r(t) + 2}{4}$$

$$F_2 = \frac{1}{Cycles} \cdot \sum_{1 \le t \le Cycles} 1 - \frac{\left| M_l(t) - M_r(t) \right|}{2}$$

$$F_3 = \frac{1}{Cycles} \cdot \sum_{1 \le t \le Cycles} \frac{2 - (S_l(t) + S_r(t))}{4}$$

Der Roboter soll sich also mit großer Geschwindigkeit (F_1), wenig Richtungsänderungen (F_2) und großem Abstand zu Hindernissen (F_3) bewegen.
Nolfi und Floreano[Nolfi2000] schlagen zur Charakterisierung von Fitnessfunktionen einen Raum mit drei Dimensionen vor. Die Dimension „funktional-verhaltensorientiert“ ist die Frage, ob die Fitness Zustände des Controllers bewertet (funktional) oder das Verhalten, das der Controller produziert (verhaltensorientiert). Die Dimension „explizit-implizit“ beschreibt die Anzahl der verwendeten Größen. Die Dimension „extern-intern“ gibt an, ob die Größen dem evolvierenden Controller zur Verfügung stehen (intern) oder nicht (extern). Im Falle der verwendeten Fitness ist die Frage nach funktional oder verhaltensorientiert aus meiner Sicht nicht zu beantworten, da die Zustände und das produzierte Verhalten nicht klar voneinander zu trennen sind. Die Fitness ist explizit, da verschiedene Komponenten des gewünschten Verhaltens verwendet werden. Eine implizite Variante könnte die ohne Berührung eines Hindernisses zurückgelegte Strecke messen, was mehr Spielraum für die gesuchte Strategie lässt

anstatt einer Optimierung der expliziten Parameter. Diese aus Sicht des Experiments eher langweilige Fitness hat aber den Vorteil, dass sie intern ist, da alle verwendeten Größen dem Controller zur Verfügung stehen.

2.2. Experiment 1

Im ersten Experiment stellte sich die Energieversorgung der Platine als besondere Herausforderung heraus. Die verwendeten Batterien reichten nur für etwa 1½ Stunden Evolution, Akku-Laufzeiten waren noch kürzer. Abhilfe schaffte hier ein Netzteil, dessen Kabel die Platine aus der Luft mit dauerhafter Energie versorgte. Die Freude dieser Lösung währte nicht lang, da die Platine während des nächsten Experiments das Kabel solange verdrehte, bis sie sich selbst an dem Kabel aufhing, wenn ihr zu wenig Aufmerksamkeit geschenkt wurde. Um nicht ständig nach dem Rechten sehen zu müssen, benutzte ich ein Bauteil namens Antitwist – ein Kabelentwirrer für Telefonhörer. Über diese drehbare Energieversorgung konnte nun ein erstes längeres Experiment gestartet werden. Dieses brachte nach einiger Zeit nur noch Individuen hervor, die in voller Geschwindigkeit gegen die nächstgelegene Wand fuhren. Zu meiner Verwunderung waren die erreichten Fitnesswerte recht hoch. Es fand sich, dass die verwendeten Abstandssensoren keine eindeutigen Werte lieferten. So stieg der gemessene Abstand ab etwa 5cm Entfernung wieder sehr rasch an, so dass sich die gegen die Wand fahrende Platine im sicheren Abstand von 20cm wähnte und so eine im Sinne der Fitness-Funktion optimale Strategie gefunden hatte. So vollzog sich die Evolution wieder einmal zunächst nur auf das Erscheinungsbild der Platine, denn ihr wurde kurzer-

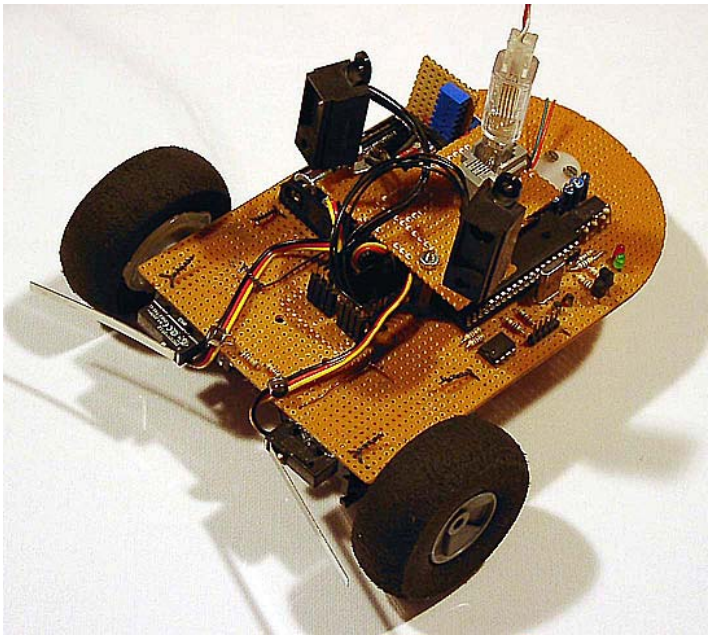


Abbildung 2-3 Die Fahrende Platine mit Antitwist und versetzten Distanzsensoren

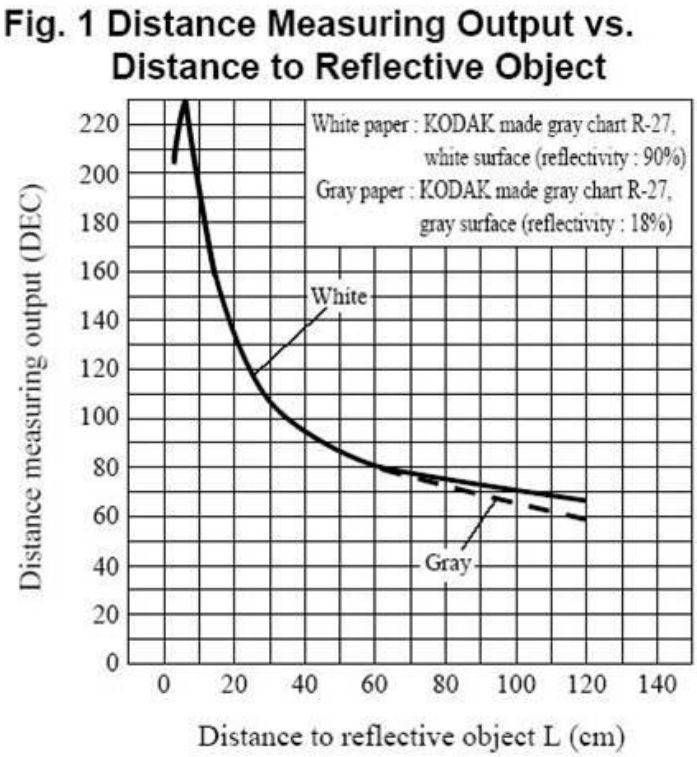


Abbildung 2-4 Werte des Distanzsensors in Abhängigkeit von der gemessenen Entfernung

hand eine zweite Ebene beschert, auf der die Abstandssensoren um 5cm nach hinten versetzt montiert wurden.

2.3. Experiment 2: Endgültige Hardware

Nach weiteren 2 Stunden Evolutionsdauer zeigte sich ein deutlich verändertes Verhalten gegenüber dem vorigen Experiment, allerdings waren keine Verbesserungen des Verhaltens gegenüber der Zufallsgeneration Null sichtbar. Die Individuen blieben häufig an bestimmten Stellen hängen und brachten so ihre jeweiligen Nachfolger in die schwierige Situation sich erst aus diesen befreien zu müssen. Dieser Staffellauf der Individuen bietet andererseits den Vorteil, dass sich keine Spezialisten ausbilden, die nur von einer bestimmten Startposition aus ein angepasstes Verhalten zeigen. Um die Abhängigkeit der Fitness von der Startposition etwas zu reduzieren und somit ähnliche Rahmenbedingungen und Chancen für die Individuen zu schaffen, gibt es zahlreiche Möglichkeiten. Man könnte die Lebenszeit eines Individuums erhöhen, die ersten Sekunden Lebenszeit nicht in die Fitness eingehen lassen oder vor jedem Individuum einige Zufallsaktionen durchführen oder sogar eine Strategie, die die Roboter in ähnliche Startbedingungen bringt. In letztgenanntem Fall wäre aber die Strategie ähnlich komplex wie die Zielstellung des Experiments selbst. Ich entschied mich dafür die ersten 10 Sekunden jedes Individuums nicht in die Fitness einfließen zu lassen und zunächst alle kritischen Stellen aus dem Parcours zu entfernen.

Abbildung 2-5 Der vereinfachte Parcours

2.4. Experiment 3: 10 Sekunden Freizeit

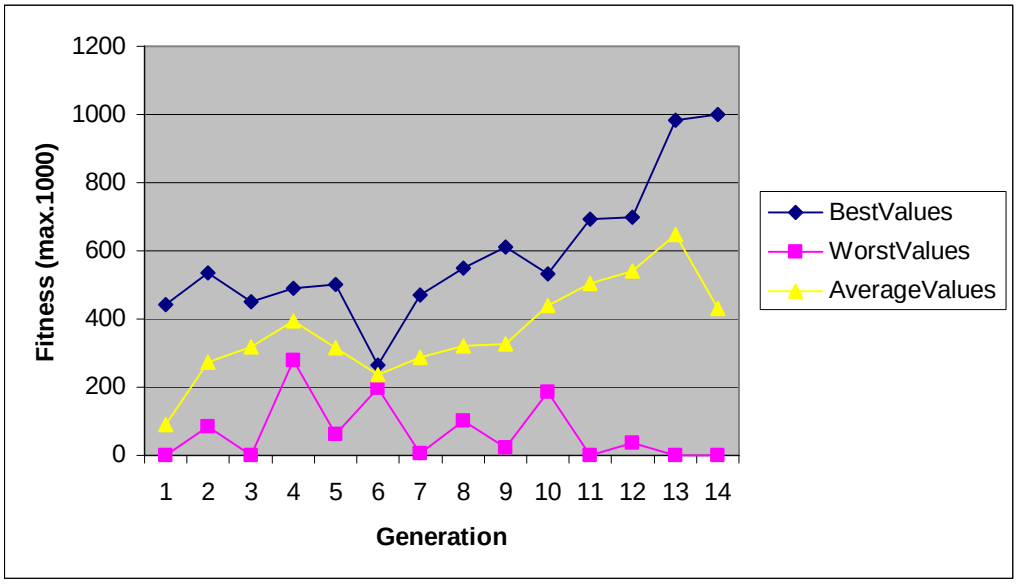
Auch das nächste Experiment ließ keine Konvergenz des Verhaltens in eine bestimmte Richtung erkennen. Individuen der Nachfolgegeneration zeigten komplett andere Strategien als die der Vorgängergeneration. Also entschied ich mich für ein etwas sanfteres Vorgehen im Suchraum, indem jedes der $M \cdot O$ Ausgangsindividuen der neuen Generation nur noch mit einer Wahrscheinlichkeit $P_{Cross} = 0.1$ einem Crossover unterzogen wird (ein Crossover betrifft zwei Genotypen, also ist die Gesamtwahrscheinlichkeit für ein Individuum ein Partner eines Crossovers zu sein gleich $2 \cdot P_{Cross}$). Da in dem verwendeten vollständig rekurrenten KNN die Abhängigkeiten der Neuronenwerte untereinander sehr groß sind, schien es mir eher unwahrscheinlich durch ein Crossover zwei separate Chromosom-Fragmente synergetisch zusammenzufügen – daher der niedrige Wert von P_{Cross}. Zudem wächst durch das verwendete One-Point-Crossover die Wahrscheinlichkeit eines Gewichts ausgetauscht zu werden mit der Posi-

tion innerhalb des Chromosoms. Um dieses so genannte „Positional Bias“ (also die positionsabhängige Austauschwahrscheinlichkeit) zu umgehen, ging ich über zum „Uniform Crossover“. Bei diesem Verfahren wird für jedes Gen (hier also jedes Gewicht) einzeln mit einer Wahrscheinlichkeit von 0.5 ausgewürfelt, ob es ausgetauscht werden soll oder nicht.

2.5. Experiment 4: P_{Cross} und „Uniform Crossover“

Das vierte Experiment umfasste 14 Generationen, was bei 20 Individuen einer Laufzeit von 14·20·(60sec+10sec) also ca. 5½ Stunden entspricht. Für die Auswertung wurde in dem EEPROM der Platine für jede Generation das beste Individuum mit seiner Fitness, die Fitness des schlechtesten Individuums und die durchschnittliche Fitness der Generation festgehalten:

Generation	*Fitness (0..1000)*			*Bestes Individuum (8Bytes hexadezimal)*
	Max	*Min*	*∅*	
1	443	0	91	E9 C5 F1 B0 C4 15 8A E5
2	534	84	274	00 AE 78 97 97 2E 35 6A
3	451	0	317	E9 8D F5 B0 E5 17 8B E5
4	490	279	395	CC C5 E9 B0 C4 11 8A E5
5	502	63	316	CC C5 E9 B0 C4 11 8A A5
6	264	196	237	CD C5 ED E0 C4 14 8A E5
7	470	6	286	CC D5 E9 B0 C4 10 CB E5
8	548	101	322	E8 87 79 B0 FC 21 08 E5
9	610	23	327	E8 C7 79 B4 FC 61 08 E5
10	533	186	439	E8 C7 79 B4 FE 61 08 E5
11	694	0	504	AA 97 75 B0 FC 29 0A A5
12	698	37	541	E9 87 71 B0 FC 21 8A A5
13	984	1	649	AA C7 6B B0 FC 61 1B 65
14	1000	0	432	AA E7 6B B0 F4 21 49 65



In der 14. Generation ist ein Individuum entstanden, das das mögliche Optimum der Fitness erreicht. In der 6. Generation kam es zu einem Einbruch der Entwicklung. Auf Basis der Aufzeichnungen lassen sich nur Vermutungen anstellen, aber dieser Einbruch ist meiner Meinung nach ein Ergebnis zweier Faktoren. Der erste betrifft das Selektionsverfahren „Truncation Selection“. Die Evolutionären Algorithmen haben wie andere Suchverfahren mit dem „exploration vs. exploitation“-Problem zu kämpfen, d.h. mit der Frage: Wie viel Aufwand soll in die Erforschung alternativer Lösungen und wie viel in die Weiterentwicklung bereits vorhandener Lösungen gesteckt werden? Hiermit verbunden ist das Problem der vorzeitigen Konvergenz in multimodalen Fitnessumgebungen – das Verharren in einem lokalen Optimum. Das bedeutet bezogen auf den evolutionären Ansatz, dass zu früh bestimmte Individuen die Population dominieren und so alternative Lösungen keine Chance mehr erhalten. Zwar verhindert „Truncation Selection“ die schnelle Übernahme der Population durch ein Super-Individuum im Vergleich zu Verfahren, in denen die Wahrscheinlichkeit Nachkommen zu erzeugen proportional zur Fitness ist. Aber in Kombination mit einer niedrigen Mutations- und Crossover-Rate kann die Population schnell durch Individuen dominiert werden, die nur geringfügig besser sind als die Alternativen. Denn im schlimmsten Falle werden die O erzeugten Nachkommen des besten Individuums unverändert in die nächste Generation übernommen und belegen die ersten M Plätze dieser Generation. Der zweite Faktor für den Einbruch ist das bereits erwähnte Problem des Staffellaufs der Individuen, d.h. ein Individuum startet aus der Endposition seines Vorgängers. Meine Vermutung ist, dass es durch den ersten Faktor zu einer Übernahme der Population durch Individuen gekommen ist, die sich durch vorsichtiges Vorwärtstasten und vorsichtiges Zurücktasten bei einem Hindernis gegenüber zu wagemutigen Individuen durchgesetzt haben. Verhaltensvergleiche der besten Individuen der Generationen stützen diese Vermutung. Durch den zweiten Faktor kommt es nun in der 6. Generation zu besonders widrigen Lebensumständen für das dominante Verhalten und die Fitness bricht sowohl im Durchschnitt als auch beim besten Individuum ein. Interessanterweise gibt es aber keinen „Genetischen Flaschenhals“[WikFlaschenhals], denn bereits in der 8. Generation erreicht ein Individuum die höchste Fitness, dessen Genotyp einen Hamming-Abstand von 17 zum besten Individuum der 7. Generation besitzt. Im folgenden Lauf soll das Crossover, die Mutationen und die genetische Diversität etwas genauer beleuchtet werden.

2.6. Experiment 5: Auswirkungen von Mutation und Crossover und genetische Diversität

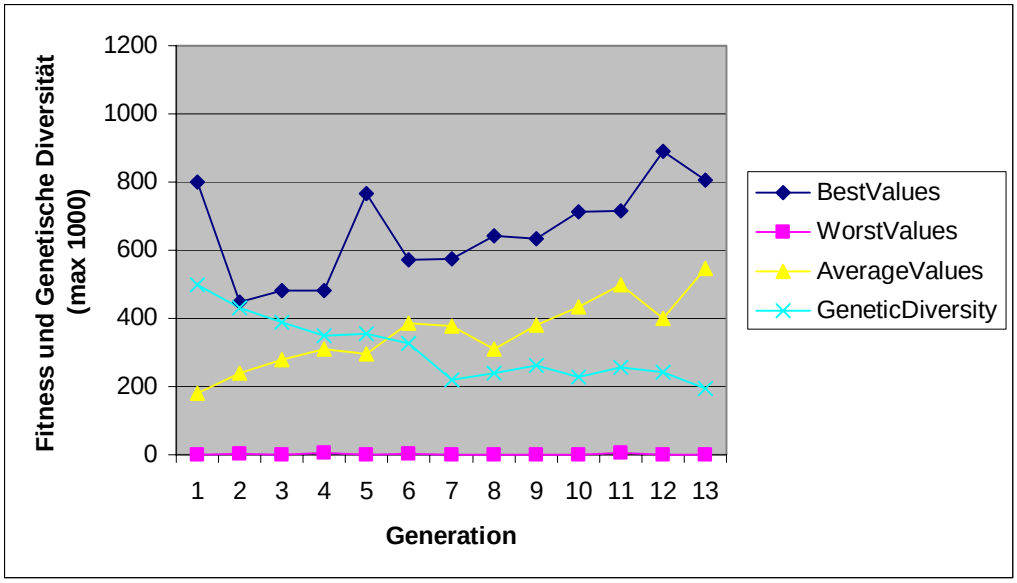
Am Ende jeder Generation muss für das verwendete „Truncation Selection“ eine Fitness-Rangliste der Individuen erstellt werden, bei einer Populationsgröße von 20 also die Plätze 1 bis 20. Um die Auswirkungen von Mutation und Selektion auf die Platzierung des Individuums zu untersuchen, zeichnete ich für jede Generation und jeden Platz 2 zusätzliche C&M-Bits (Crossover&Mutations-Bits) auf:

C&M-Bits des Platzes P	***Bedeutung: Genotyp des Individuums auf Platz P ist …***
00	... weder Ergebnis einer Selektion noch einer Mutation
10	... Ergebnis eines Crossovers
01	... Ergebnis einer Mutation
11	... Ergebnis eines Crossovers und einer Mutation

Dazu kommt in jeder Generation G noch ein Wert der Genetischen Diversität $gendiv(G)$. Ist M die Menge aller möglichen Paare (u, v) von Genotypen der Generation G (bei 20 Individuen ergeben sich 19+18+…+1=190 Paare) und $h(u, v)$ der Hamming-Abstand der Genotypen u und v, so berechnet sich $gendiv(G)$ wie folgt:

$$gendiv(G) = \sum_{(u,v)\in M} h(u,v)$$

Daneben entschied ich mich die im Experiment 3 eingeführte Strategie „10 Sekunden Freizeit“ zu erweitern und die 10 Sekunden in 10 Abschnitte à eine Sekunde einzuteilen. Zu Beginn jedes Abschnittes werden Zufallswerte für die beiden Servos bestimmt und dann eine Sekunde beibehalten. Diese Strategie hilft folgendes Problem zu umgehen: Innerhalb des Parcours können Plätze existieren, in denen die Individuen durch ein nicht gewünschtes Verhalten sehr hohe Fitnesswerte erreichen – bspw. eine hervorstehende Ecke gegen die der Roboter so fährt, dass beide Infrarotsensoren hohe Abstände signalisieren. Auch wenn diese Plätze mit einer sehr niedrigen Wahrscheinlichkeit getroffen werden können, so kann dies im schlimmsten Fall zu einem kompletten Stillstand der Evolution führen.
Das Experiment ging über 13 Generationen mit einer Laufzeit von ca. 5 Stunden (13·20·(60sec+10sec)).



Generation	*C&M-Bits der Plätze 1..20 (durch Leerzeichen getrennt)*
1	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2	00 01 11 01 11 01 01 01 01 01 01 00 00 01 10 11 11 01 01 11
3	01 01 01 01 00 11 01 11 01 01 00 01 01 01 11 00 11 00 11 11
4	00 01 00 00 01 01 01 00 01 01 01 11 11 01 10 01 01 11 01 00
5	11 01 11 01 01 01 00 01 00 00 01 01 01 01 01 00 11 01 11 01
6	10 00 11 00 01 01 00 01 00 01 01 01 01 00 01 01 01 01 01 00
7	11 01 01 11 01 01 01 10 01 11 01 01 01 01 00 01 01 01 01 01
8	01 01 01 01 01 01 01 01 01 01 01 11 11 01 00 01 00 01 00 01
9	01 01 00 00 00 01 01 01 01 00 11 01 01 01 01 01 01 01 11 01
10	01 00 01 01 01 00 00 01 01 01 01 01 00 01 11 11 01 01 01 01
11	01 01 01 01 00 01 11 01 00 01 10 01 01 11 01 01 01 11 01 01
12	11 01 01 00 01 00 01 01 01 01 11 11 11 01 01 11 01 11 01 01
13	01 01 01 01 01 01 01 10 01 00 01 10 11 01 10 01 01 01 01 01

Gekreuzte und nicht gekreuzte Individuen finden sich auf fast allen Platzierungen und die größten Sprünge in der maximalen Fitness gehen sogar aus gekreuzten Individuen hervor (siehe Generation 7 und 12). Das Crossover ist hier also ein wichtiger Faktor, der die Evolution begünstigt.

Mutationen gibt es wie zu erwarten auf jeder Platzierung, da eine ungerichtete Veränderung des Genotyps auch Veränderungen der Fitness in jeder Richtung zulässt.

Ist *P(i)* die Wahrscheinlichkeit, dass genau *i* Individuen der Generation nicht mutiert sind, P_{Ind} die Wahrscheinlichkeit, dass ein einzelnes Individuum nicht mutiert ist und $P_{Mutationsrate}$=3% die Wahrscheinlichkeit, dass ein einzelnes Gen mutiert ist, so ergibt sich bei einer Genomlänge von 64Bit der Erwartungswert $E(\xi)$ von nicht mutierten Individuen pro Generation:

$$P_{Ind} = (1 - P_{Mutationsrate})^{64}$$

$$P(i) = (1 - P_{Ind})^{20-i} \cdot P_{Ind}{}^{i} \cdot \binom{20}{i}$$

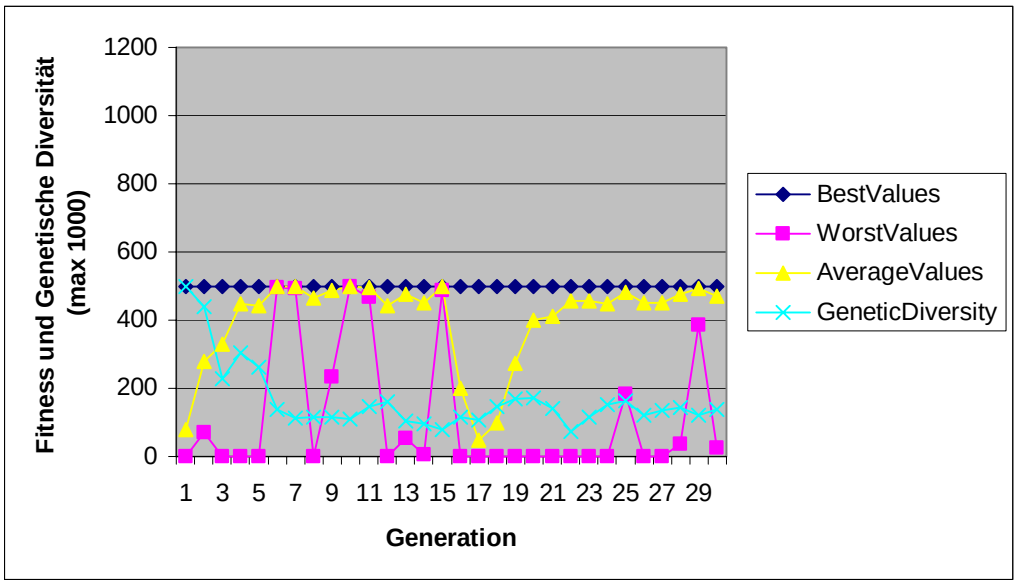
$$E(\xi) = \sum_{i=0}^{20} i \cdot P(i) \approx 2.8$$

Dieser Wert deckt sich mit den C&M-Bits.

Die Genetische Diversität kann ein Parameter sein, um zu beobachten, ob die Evolution in eine Sackgasse gelaufen ist. Ihre Entwicklung in diesem Lauf überrascht nicht bzw. verhält sich sogar gewünscht in dem Sinne, dass mit ihrer Abnahme auch eine Steigerung der durchschnittlichen Fitness verbunden ist. Um das Absinken unter ein gewünschtes Maß zu verhindern, gibt es Ansätze wie Inzestverbot oder Ausselektieren von Duplikaten [Gerdes2004].

2.7. Experiment 6: Inkrementelle Evolution oder doch nicht?

Der sechste Lauf sollte als Ausgangspunkt einer inkrementellen Evolution dienen. D.h. die besten Individuen, die aus diesem Lauf hervorgehen, sollten als Generation Null in einen schwierigeren Parcours starten und die Evolution unter diesen veränderten Rahmenbedingungen fortsetzen. Nicht das erste Mal lief die Evolution aber anders als erwartet ab. Der Lauf ging über 30 Generationen mit einer Gesamtzeit von 6 Stunden und 40 Minuten.



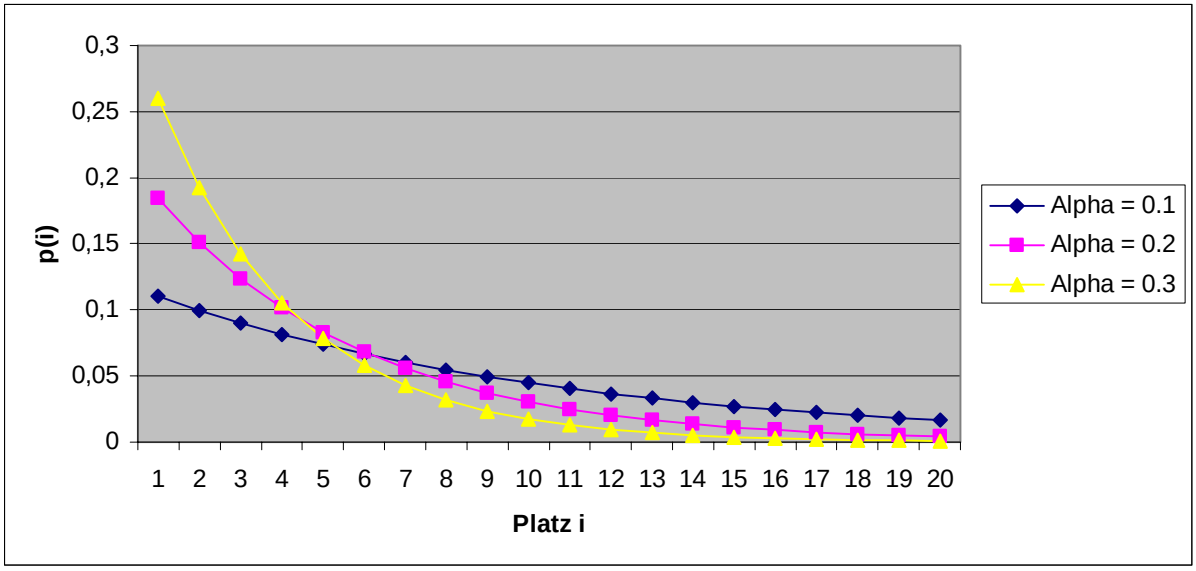
Was zu sehen ist, ist das Beispiel einer vorzeitigen Konvergenz (zum Einbruch der durchschnittlichen Fitness siehe Experiment 4). Die Evolution hat eine Richtung eingeschlagen, die ihr schnelle Fitnesserfolge im Maximum wie im Durchschnitt einbrachte, sich aber langfristig als genetische Sackgasse entpuppte.

Das verwendete Selektionsverfahren „Truncation Selection“ ist ein diskriminierendes Eliteverfahren. Das bedeutet, dass die besten Individuen immer überleben und die Überlebens-

wahrscheinlichkeit von schlechteren Individuen gleich Null ist. Hiermit verbunden sind ein starker Selektionsdruck und eine hohe Wahrscheinlichkeit einer vorzeitigen Konvergenz.
Um dagegen zu steuern implementierte ich für das nächste Experiment eine andere Selektionsstrategie. Zunächst behielt ich die Rangliste der Individuen bei, d.h. vom Fitnesswert wird nur beim Erstellen der Rangliste Gebrauch gemacht. Jedem Platz *i* der Rangliste wird eine Selektionswahrscheinlichkeit *p(i)* zugewiesen. Alle *p(i)* sind dann Ausgangspunkt für ein Selektionsverfahren namens „Stochastic Universal Sampling" (SUS). Bei diesem Verfahren stellt man sich die Selektionswahrscheinlichkeiten wie beim Roulette-Rad-Verfahren[Gerdes2004-2] als Abschnitte eines Kreises *A* mit dem Umfang Eins vor. Jedem Abschnitt ist eine Platzierung *i* (und somit ein Individuum) zugeordnet und die Länge eines Abschnittes entspricht dabei der Selektionswahrscheinlichkeit *p(i)*. Beim SUS gibt es einen zweiten Kreis *B* auf dem *N* Punkte äquidistant auf den Umfang Eins verteilt sind (wobei *N* die Populationsgröße ist). Das Rad *B* wird nun um einen Zufallswinkel gedreht und über das Rad *A* gelegt. Jeder Punkt des Kreises *B* befindet sich nun in einem Abschnitt des Kreises *A* und für jeden Punkt wird das entsprechende Individuum einmal in die nächste Generation übernommen (welches dann ggf. noch gekreuzt und mutiert wird). Das Verfahren garantiert, dass die Anzahl der Kopien eines Individuums des Platzes *i* zwischen $\lfloor p(i) \cdot N \rfloor$ und $\lceil p(i) \cdot N \rceil$ liegt ($\lfloor\rfloor$ Abrundung bzw. $\lceil\rceil$ Aufrundung auf ganzzahligen Wert) [Gerdes2004-3]. SUS ist also ein nichtdiskriminierendes Eliteverfahren. Dies garantiert auf der einen Seite, dass das beste Individuum nicht durch den Selektionsprozess verloren geht, was bei der kleinen Populationsgröße (also einem kleinen Suchfenster) wichtig ist. Auf der anderen Seite haben alle Individuen mit $p(i)>0$ die Möglichkeit in die Folgegeneration kopiert zu werden.
Die Bestimmung der *p(i)* über die Rangliste bedeutet zunächst einmal, dass die Informationen der Fitnesswerte nicht voll ausgeschöpft werden. Die Abstraktion enthält aber die Möglichkeit über die *p(i)* einen über die Generationen konstanten Selektionsdruck festzulegen, der unabhängig von den tatsächlich erreichten Fitnesswerten und somit auch von der Zielstellung des Experimentes ist. Die Funktion *p(i)* muss die Bedingungen $0 \le p(i) \le 1$ und $\sum p(i) = 1$ erfüllen. Ich wählte ein „Exponentielles Ranking"[YaoXin]:

$$p(i) = \frac{1}{\sum_{1 \le j \le N} p(j)} \cdot e^{-\alpha(i-1)}$$

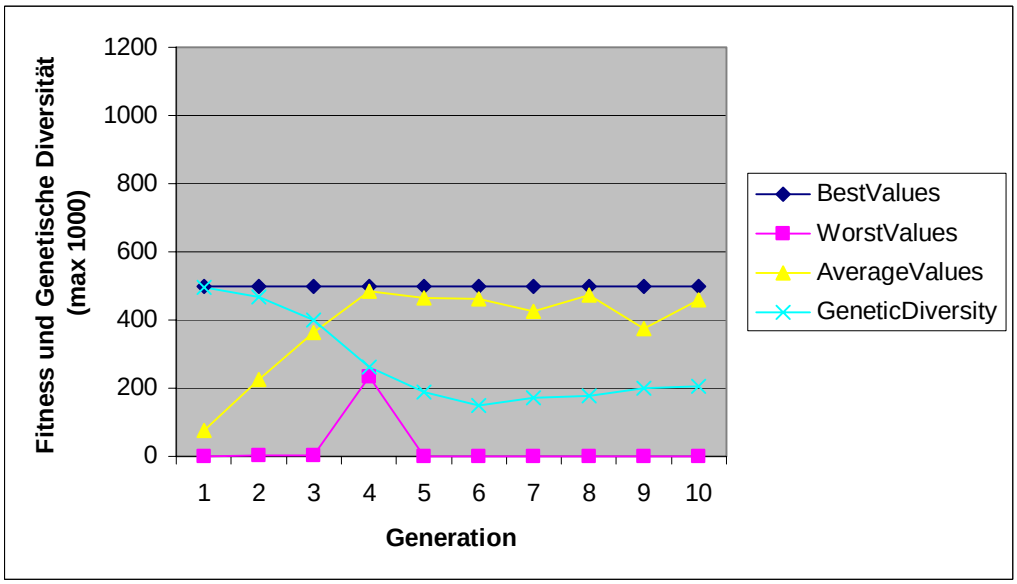
Die Selektionswahrscheinlichkeit nimmt also mit schlechterer Platzierung exponentiell ab. Der erste Bruch dient der Normierung und über α lässt sich der Selektionsdruck regeln.



Im folgenden Experiment ist α=0.2.

2.8. Experiment 7: „Stochastic Universal Sampling“ und „Exponentielles Ranking“

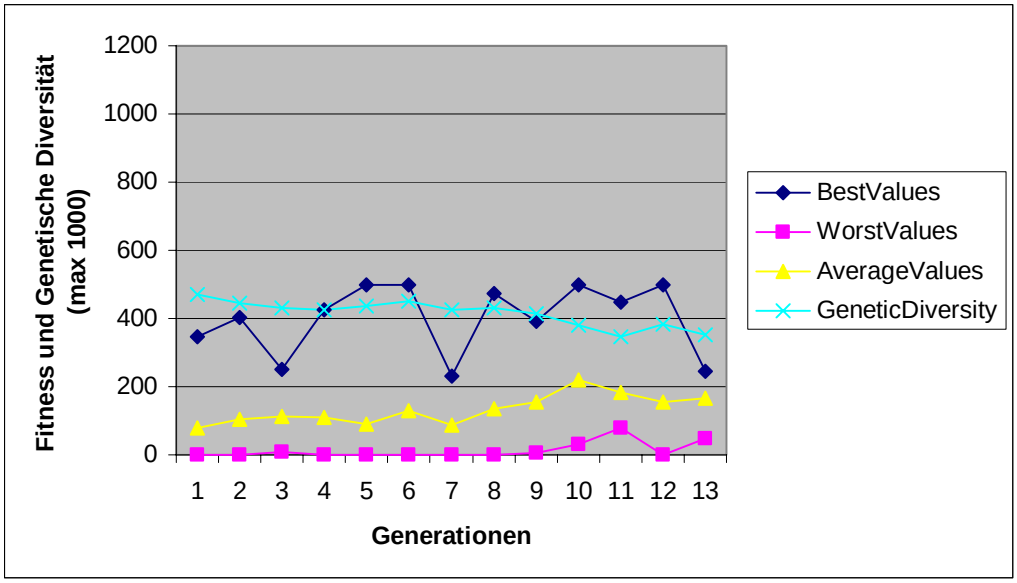
Nach 3 Stunden und 50 Minuten und 10 Generationen ergab sich das folgende Bild:



Die Genetische Diversität erreicht bereits nach vier Generationen die Hälfte ihres Ausgangswertes. Der durchschnittliche Wert der Fitness konvergiert schnell gegen das Maximum, so dass damit zu rechnen ist, dass α noch weiter abgesenkt werden kann, ohne die Konvergenz zu gefährden.

2.9. Experiment 8: $\alpha = 0.15$

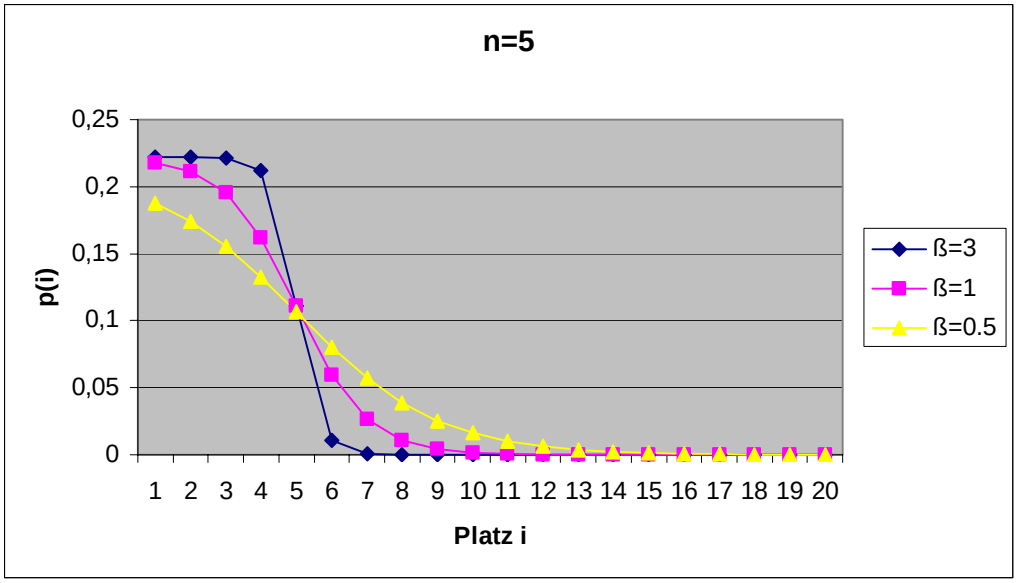
Der Lauf ging über 13 Generationen mit einer Laufzeit von etwa 5 Stunden.

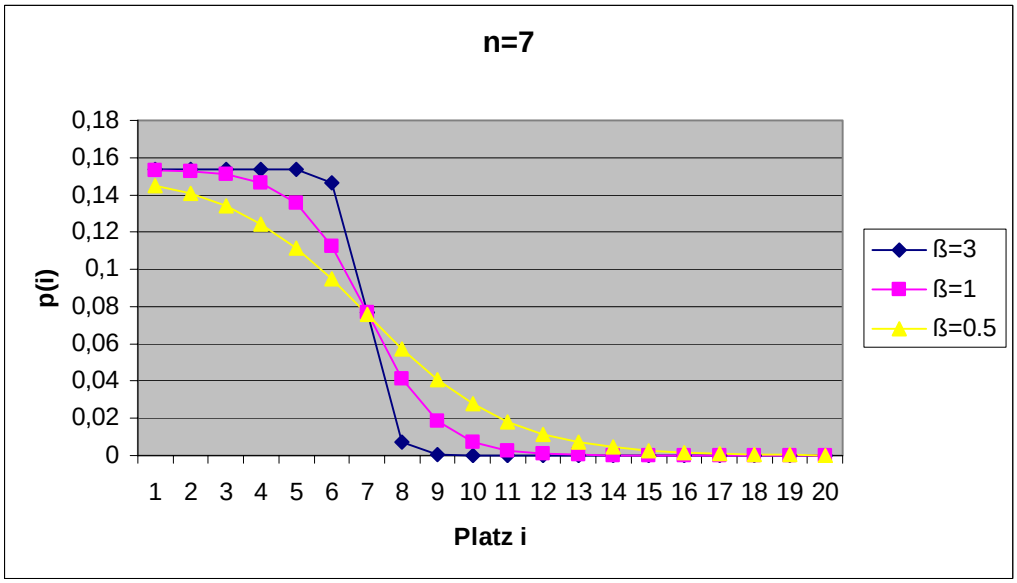


Die Konvergenz wurde durch das Herabsenken von α stark gehemmt. Die Übernahme des besten Individuums der Generation ist zwar durch das verwendete SUS garantiert, allerdings liegt die Zahl der Genotyp-Kopien für das beste Individuum nur noch zwischen drei und vier. Scheinbar reicht dies nicht aus, um die beste Fitness auf hohem Niveau zu halten. Die Fitness bleibt einmal mehr an der 500er-Marke stehen.

Wünschenswert wäre ein Verfahren, welches wie Truncation Selection eine schnelle Konvergenz garantiert, aber dennoch genetische Überraschungen parat hält, um eine vorzeitige Konvergenz überwinden zu können. Also entschied ich mich auf das Exponentielle Ranking und seinen sensiblen Parameter α zu verzichten und stattdessen eine Variante der Sigmoid-Funktion zu verwenden:

$$p(i) = \frac{1}{\sum_{1 \le j \le N} p(j)} [1 - SIGMOID(i - n)] = \frac{1}{\sum_{1 \le j \le N} p(j)} [1 - \frac{1}{1 + e^{-\beta(i-n)}}]$$



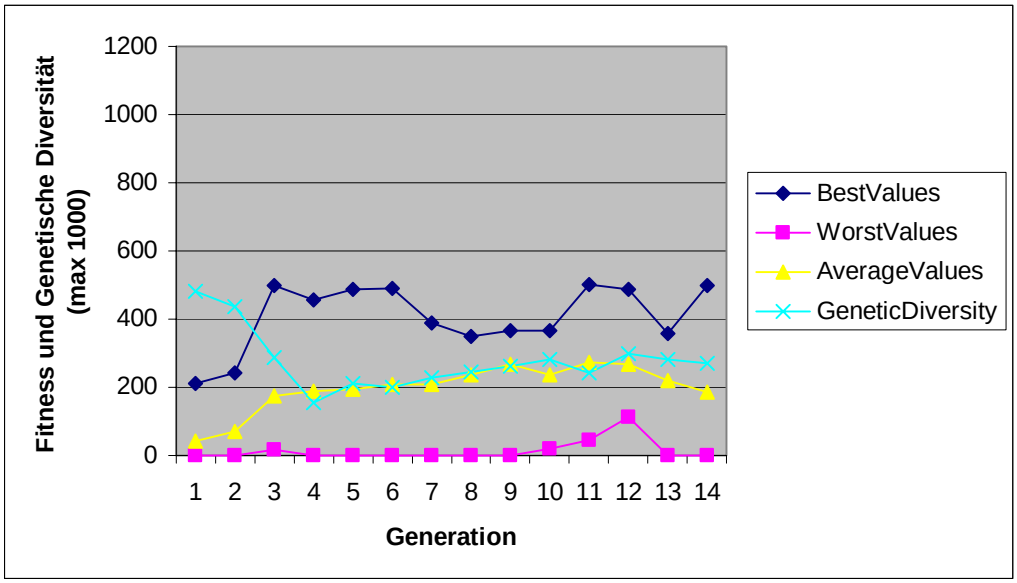


n bestimmt den Punkt des maximalen Abstiegs und *ß* den Abstieg selbst. Für $n=5$ und $ß \to \infty$ besteht also kein Unterschied zu dem in früheren Experimenten benutzten Truncation Selection.

Für das folgende Experiment setzte ich in Anlehnung an frühere Experimente $n=5$ und für einen sanften Übergang $ß=1$.

2.10. Experiment 9: „Sigmoid Ranking“ mit n=5 und ß=1

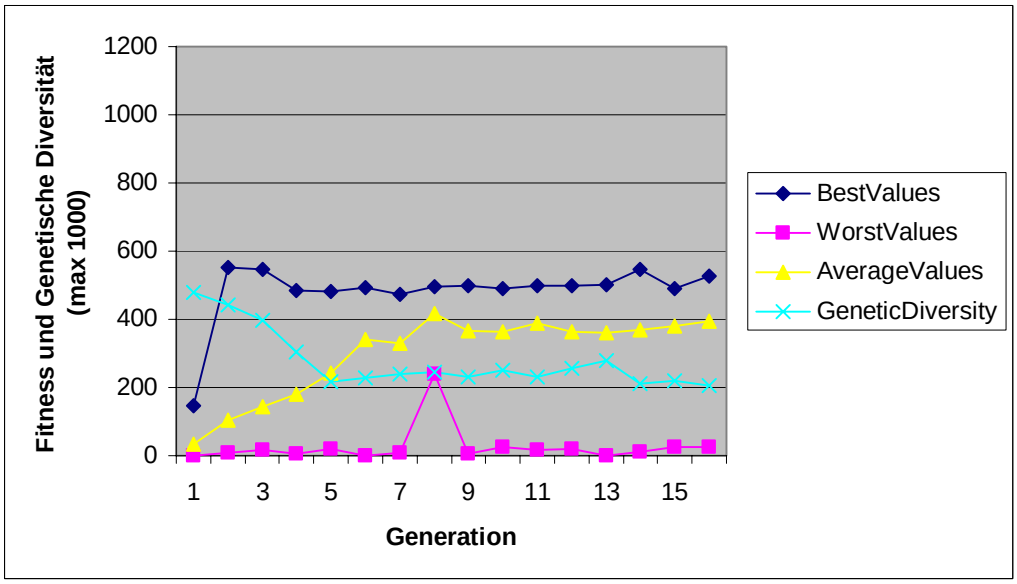
Dieser Lauf ging über 14 Generationen, also etwa 5 ½ Stunden.



Dieser Lauf stand von der ersten Generation an unter einem schlechten Stern. Zum ersten Mal wurden in den ersten beiden Generationen keine Fitness-Werte jenseits der 250 erreicht. Um zu den vorherigen Läufen vergleichbare Daten zu erhalten, schickte ich gleich den nächsten Lauf unter gleichen Bedingungen hinterher.

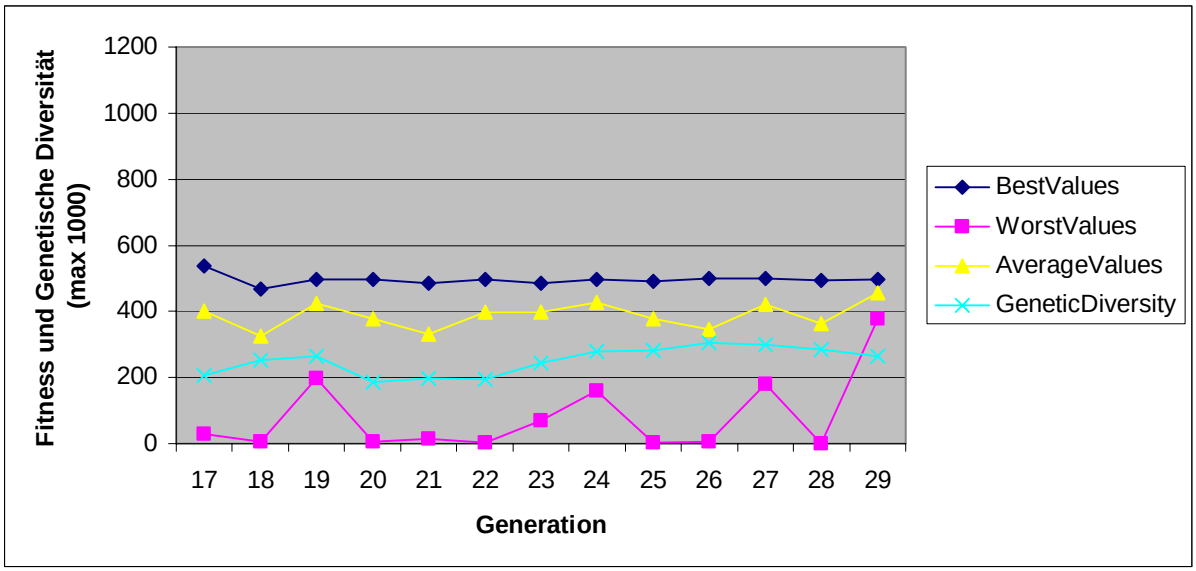
2.11. Experiment 9: Teil2

Dem zweiten Lauf gab ich 16 Generationen und 6 Stunden und 10 Minuten Zeit.



Die Evolution befindet sich auch hier in einem lokalen Optimum. Die Genetische Diversität hält sich jenseits der 200er-Marke und die durchschnittliche Fitness konvergiert sehr langsam gegen die maximale Fitness. Das „Sigmoid-Ranking“ liefert also das gewünschte Verhalten, allein die maximale Fitness lässt zu wünschen übrig. Interessant wäre in der weiteren Entwicklung zu sehen, ob der hohe Wert der Genetischen Diversität ausreicht, um unter „Sig-

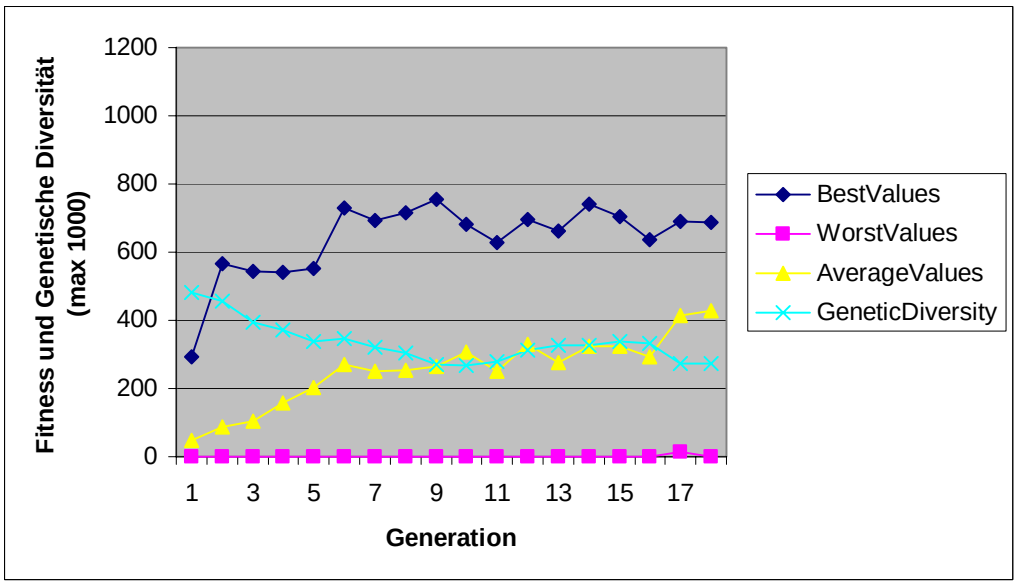
moid Ranking“ und SUS das lokale Optimum zu verlassen. Also gab ich der Evolution noch ein paar Stunden Zeit.



Die Fitness bleibt wieder einmal bei der 500er-Marke stehen und Mutation und Crossover reichen nicht aus, um dieses lokale Optimum zu verlassen. Um nicht noch weiter an den Operatoren und den entsprechenden Parametern zu experimentieren, entschied ich mich dazu die Suche in die Breite auszudehnen und bis an die Grenzen des verfügbaren Arbeitsspeichers zu gehen. Im folgenden Experiment erhöhte ich die Populationsgröße N auf 40 und den Parameter n des „Sigmoid Rankings“ auf 10.

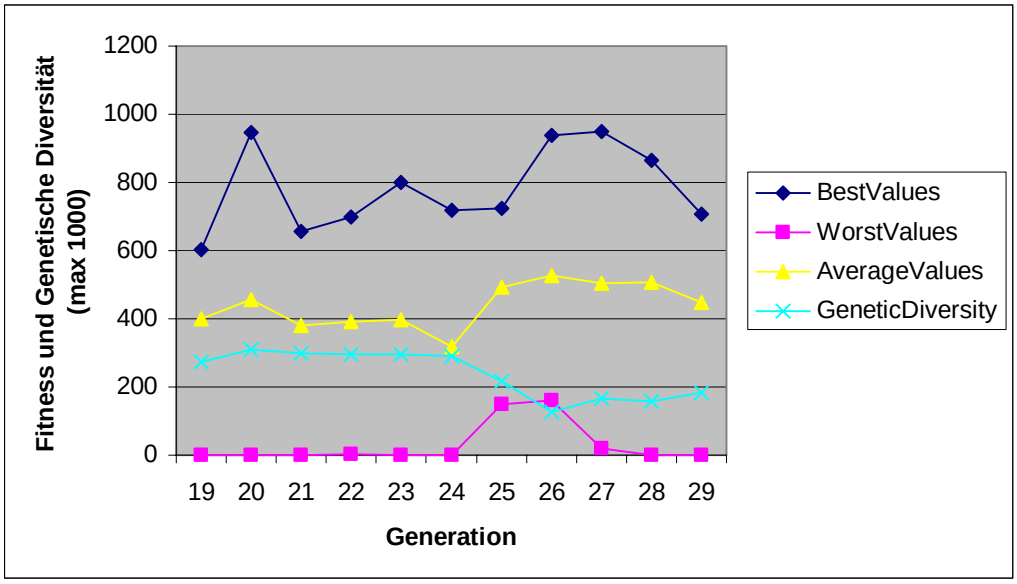
2.12. Experiment 10: N=40 und n=10

Nach 18 Generationen und 14 Stunden Dauer (18·40·(60+10)sec) zeigte sich folgendes Ergebnis.



Das Verfahren konvergiert, allerdings in langsamer Geschwindigkeit. Die maximale Fitness erreicht noch nicht die gewünschten Werte. Dass die schlechtesten Individuen durchgehend Werte um die Null erreichen ist ein Produkt der höheren Populationsgröße in Kombination mit einer hohen Mutationsrate und dem verwendeten „Sigmoid Ranking“, das auch den

schlechtesten Individuen noch eine Selektionswahrscheinlichkeit höher als Null garantiert. Um noch höhere Fitnesswerte zu erreichen, entschloss ich mich ein wenig mit der Brechstange vorzugehen und die begonnene Evolution unter verschärften Bedingungen fortzusetzen. Zum einen senkte ich die Mutationsrate von 3 auf 2% (was zu einer höheren durchschnittlichen Fitness führen sollte) und zum anderen den Faktor *n* des Sigmoid-Rankings von 10 auf 6 (was einem höheren Selektionsdruck entspricht). Das bedeutet, die Suche konzentriert sich nun stärker auf die vorhandenen Lösungen (exploitation) und gibt weniger Raum für die Suche nach Alternativen (exploration).



Die Ergebnisse entsprechen den Erwartungen. Die genetische Diversität sinkt und die Fitness klettert sowohl im Durchschnitt als auch im Maximum. Die Sprünge im Maximum zeigen, dass die Evolution noch kein generell adaptiertes Verhalten hervorgebracht hat, sondern noch ein stark von der Startsituation des Roboters abhängiges. Trotzdem soll die Evolution nun endlich Ausgangspunkt der ursprünglich geplanten Inkrementellen Evolution sein. Hierfür ist die letzte Generation Startgeneration der neuen Evolution, die in einem veränderten, schwierigeren Parcours stattfindet.

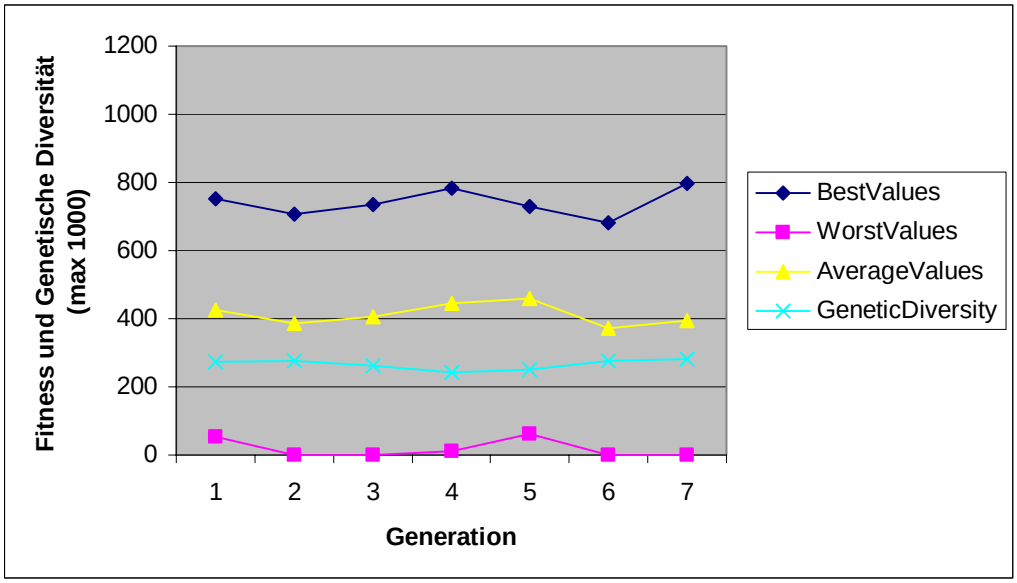
Abbildung 2-6 Der neue, schwierigere Parcours

Die Parameter der Evolution sind wieder eine Mutationsrate von 3%, eine Populati-

onsgröße von N=40 und einem Faktor von n=10 (für mehr Exploration in der neuen Umgebung).

2.13. Experiment 11: Inkrementelle Evolution

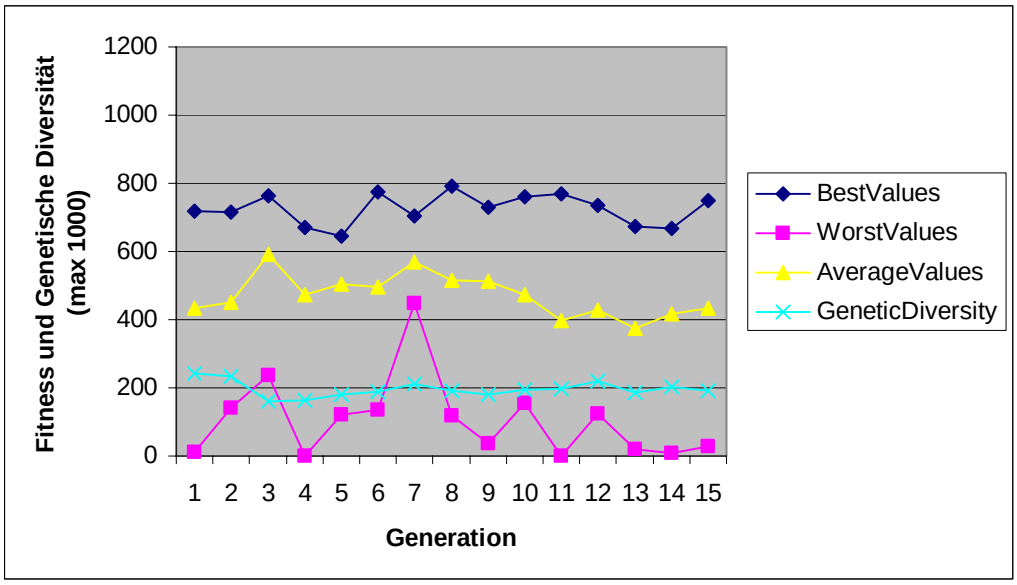
Das elfte Experiment ging über etwa 8 Stunden. Leider lassen sich nur 7 Generationen davon auswerten, da sich die Platine nach dieser Zeit wie ein Käfer auf den Rücken gedreht hatte und die Evolution ab diesem Zeitpunkt unbrauchbar wurde.



Die Evolution machte insgesamt keine Fortschritte. Insbesondere der Abstand zwischen durchschnittlicher und maximaler Fitness bleibt konstant, was wieder auf eine zu hohe Mutationsrate schließen lässt. Also startete ich die Inkrementelle Evolution neu. Dieses Mal mit einer Populationsgröße von N=20 (für eine schnellere Evolution), einem Faktor von n=5 und einer Mutationsrate von 2%. Die besten 20 Individuen der letzten Generation des zehnten Laufes bilden im folgenden Lauf den Ausgangspunkt der neuen Evolution.

2.14. Experiment 12: Inkrementelle Evolution mit Mutationsrate von 2%, N=20 und n=5

Der Lauf ging über 15 Generationen mit einer Dauer von 5 Stunden und 50 Minuten.

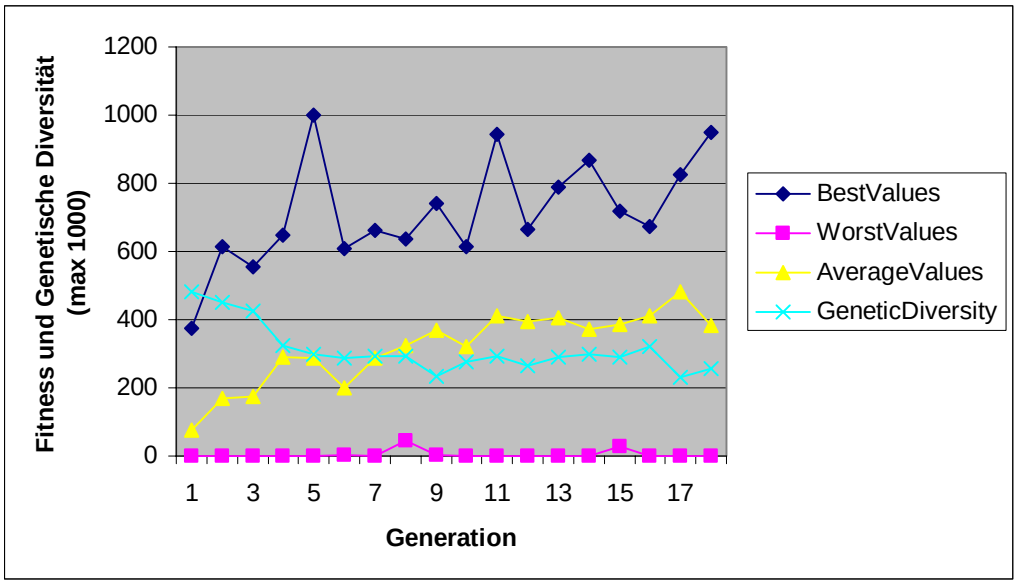


Die Senkung der Mutationsrate liefert nicht den gewünschten Effekt. Die Genetische Diversität sinkt auf 200 ab. Aber scheinbar ist das Verhalten im Durchschnitt nicht besser als die durchschnittliche Fitness es vermuten lässt. In bestimmten Startpositionen erreicht es eine hohe Fitness, aber eben nur in sehr wenigen – also wieder ein Fall einer vorzeitigen Konvergenz (dieses Mal sichtbar in der Stagnation der durchschnittlichen Fitness).

Um dem ganzen Verfahren noch eine letzte Chance zu geben, startete ich noch ein letztes Experiment. Die Evolution wird von vorne gestartet, dieses Mal allerdings in dem neuen Parcours. Die Populationsgröße beträgt wieder N=40 (bei n=10) für eine breitere Suche und die Mutationsrate ist 2% für eine etwas schnellere Konvergenz.

2.15. Experiment 13: N=40, n=10 und Mutationsrate 2% in neuem Parcours

Nach 18 Generationen und 14 Stunden lieferte die Evolution das folgende Bild.



Erfreulich sind die hohen erreichten Werte im Maximum, was zeigt, dass die Evolution dieses Mal nicht in ein lokales Optimum gelaufen ist, sondern optimales Verhalten im Sinne der Fitness liefert - allerdings wieder nicht von allen Startpositionen.

An dieser Stelle gebe ich mich mit dem erreichten Resultat zufrieden, da das Vermeiden von kritischen Stellen auch Bestandteil des Verhaltens des besten Individuums ist.

Abbildung 2-7 Die schwierigsten Startpositionen

3. Ergebnisse und Ausblick

Die durchgeführten Experimente gaben einen Einblick in die speziellen Herausforderungen einer Onboard-Evolution. Zunächst sind dies Schwierigkeiten bei der Durchführung der Evolution selbst:

- Die Energieversorgung musste über einen langen Zeitraum gewährleistet sein, was zu einem Betrieb über ein Netzteil führte.
- Zeit ist eine besondere Herausforderung der Onboard-Evolution. Was in einer Simulation zeitlich beschleunigt werden kann, spielt sich auf der Hardware in Echtzeit ab, so dass die Evolutionen bis zu 14 Stunden in Anspruch nahmen.
- Räumliche Ressourcen müssen zur Verfügung gestellt werden, die zum einen durch störende Einflüsse von außen geschützt sind und zum anderen die Umwelt vor den störenden Einflüssen der Evolution selbst schützt. So beiläufig dieser Punkt auch klingen mag, so empfehle aus eigener Erfahrung dies nicht zu unterschätzen.
- Hardware verschleißt unter der Dauerbelastung einer Evolution. Im Laufe der geschilderten Experimente mussten vier Servomotoren und zwei Chips ausgetauscht werden. Neue mechanische Bauelemente (wie Servomotoren) verhalten sich nicht wie die alten und machen eine neue Kalibrierung der Software und ggf. eine neue Anpassung der gewonnenen Controller nötig.
- Das Debuggen der Software gestaltet sich auf der Hardware schwierig. In den Experimenten wurden Debug-Informationen über ein angeschlossenes Oszilloskop ausgelesen.

Weitere Schwierigkeiten ergeben sich aus der beschränkten Hardware des Roboters:

- Die Auswertung der Evolutionen ist über aufgezeichnete Evolutionsdaten möglich. Hierfür stand ein 1KByte großer EEPROM zur Verfügung, so dass jedes Bit an Informationen wohlüberlegt sein musste (siehe C&M-Bits).
- Der geringe Arbeitsspeicher ermöglicht nur kleine Populationsgrößen. Zu diesem Zweck wurden die Gewichte in 8Bit-Gleitkommazahlen kodiert, so dass Größen von N=40 überhaupt realisierbar wurden.

Kleine Populationsgrößen führen zu einer erhöhten Gefahr einer vorzeitigen Konvergenz, wie sie im Experiment 6 zum ersten Mal aufgetreten ist. Auf der anderen Seite ist Zeit ein kritischer Faktor einer Onboard-Evolution, so dass ein zu geringer Konvergenzdruck keine geeignete Lösung ist. In der Arbeit wurde so zunächst ein Blick auf die genetischen Operatoren geworfen. Die verwendeten Operatoren sollten hierbei problemunabhängig gestaltet sein, damit die gewonnenen Erkenntnisse auf Experimente mit einer anderen Zielstellung übernommen werden können. Das „Uniform Crossover“ (siehe Experiment 4) behob das Problem der positionsabhängigen Austauschwahrscheinlichkeit eines Gens. Dies erweist sich dann als nützlich, wenn kein Problemwissen in die Kodierung des Genoms gesteckt wird (bspw. Neuronenverbände in einer Gensequenz).

Ebenso hat sich die Einführung einer Crossover-Wahrscheinlichkeit als effektiv erwiesen (auch im Experiment 4). Wurde jedes Individuum einem Crossover unterzogen, waren die Sprünge im Suchraum zu groß und die Konvergenz blieb auf der Strecke. Die Aufzeichnung

der C&M-Bits machte aber deutlich, dass das Crossover ein wirksamer Operator sein kann (siehe Experiment 5).
Die so erreichten Resultate waren jedoch nicht reproduzierbar. Die Nachfolgeexperimente zur Inkrementellen Evolution scheiterten an der vorzeitigen Konvergenz. An dieser Stelle hätte bereits die Populationsgröße auf 40 angehoben werden können, was vermutlich schneller zum Erfolg geführt hätte als der gegangene Weg über die Selektionsstrategien.
Für die Untersuchung der Selektionsstrategien hat sich die Aufzeichnung der genetischen Diversität nützlich gezeigt. So wurde der Abfall der genetischen Diversität durch Übergang zum nichtdiskriminierenden Selektionsverfahren „Stochastic Universal Sampling“ (SUS) deutlich gebremst (ab Experiment 7). Das „Exponentielle Ranking“ konnte nicht überzeugen, da der Konvergenzdruck bei Variation des Faktors α starken Schwankungen unterlag (siehe Experiment 7 und 8). Besser zu regulieren ist das „Sigmoid Ranking“ mit seinen beiden Parametern β und n. Im Falle $\beta \to \infty$ geht das „Sigmoid Ranking“ in Kombination mit SUS in „Truncation Selection“ über. Letztlich brachte die Anhebung der Populationsgröße den gewünschten Erfolg und adaptiertes Verhalten in dem schwierigeren Parcours (siehe Experiment 13). Die vorzeitige Konvergenz konnte in den Experimenten 9 bis 12 auch durch die höhere genetische Diversität nicht überwunden werden.
Als letzte Schwierigkeit führte der Staffellauf der Individuen zu unterschiedlichen Rahmenbedingungen für die Individuen und somit zu unregelmäßigen und verzerrten Fitnessverläufen. Zwar reduziert dies die Konvergenzgeschwindigkeit, führt aber im Endeffekt zu robusterem Verhalten der Individuen der späteren Generationen. Die Strategie der Zufallsaktionen zu Beginn des Laufes eines Individuums könnte so ausgebaut werden, dass diese Aktionen in den Lauf selbst gestreut werden, so dass sich das Individuum während eines Laufes aus verschiedenen Startpositionen behaupten muss. Dies würde zunächst zu einer geringeren Gesamtfitness führen, aber den Fitnessverlauf weiter glätten.
Darüber hinaus könnte der erreichte durchschnittliche Fitnesswert durch einen dritten mittigen Distanzsensor, ein größeres neuronales Netz oder durch längere Experimente weiter angehoben werden.
Eine weitere Möglichkeit für die Fortführung der Experimente könnte eine neue Zielstellung bspw. unter Verwendung der Helligkeits- und Bumpsensoren sein.

4. Kurzes Nachwort

Ich möchte mich noch kurz für die Geduld bedanken, die mir von Manfred Hild, Professor Burkhard und meinen Mitbewohnern entgegengebracht wurde – die Arbeit hat nicht nur an meinen Nerven gezehrt (aber trotzdem Spaß gemacht).

5. Literaturverzeichnis

[Nolfi2000] – „Evolutionary Robotics“, Stefano Nolfi und Dario Floreano
[Holl75] – „Adaptation in Natural and Artificial Systems“, Holland J.H.
[WikFlaschenhals] - http://de.wikipedia.org/wiki/Genetischer_Flaschenhals
[Gerdes2004] – siehe Seite 62 in „Evolutionäre Algorithmen“, Gerdes, Klawonn und Kruse
[Gerdes2004-2] – siehe Seite 39
[Gerdes2004-3] – siehe Seite 82
[YaoXin] – http://www.dlc.sjtu.edu.cn/online%20lecture/YaoXin-Lecture.pdf