

BA Leipzig
Informatik

Beleg

Benchmarking von genetischen Algorithmen

Bearbeiter: Falko Nater IT2009
Hans Laser IT2009

Modul: Evolutionäre Algorithmen
Abgabe: 07.08.2012

Ehrenwörtliche Erklärung

Wir versichern hiermit ehrenwörtlich durch unsere Unterschrift, dass der vorliegende Beleg selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt wurde. Keine weiteren Personen waren an der geistigen Herstellung des vorliegenden Belegs beteiligt. Der Beleg wurde noch nicht in gleicher oder ähnlicher Form oder auszugsweise im Rahmen eines anderen Belegs vorgelegen.

Falko Nater

Hans Laser

Leipzig, 07.08.2012

1 Einleitung

Genetische Algorithmen sind eine Variante der evolutionären Algorithmenklasse. Die Aufgabe der evolutionären Algorithmen ist es die Grundprinzipien der biologischen Evolution auf Optimierungsprobleme anzuwenden, um diese zu lösen. Dazu werden verschiedene Parameter der Evolution zufällig manipuliert um die Optimierung möglichst schnell zu bewirken. Solche Parameter sind Selektion, Kombination und Mutation. Diese Parameter beeinflussen die Fitness, die zur Bewertung der Generation bzw. der Individuen berechnet wird und dabei das Optimierungsproblem abbilden. Der Vorteil der evolutionären Algorithmen zu anderen Optimierungsverfahren besteht darin, dass über die Natur eines Problems keine genaue Aussage getroffen werden muss. Durch das reine sog. "Durchprobieren" von möglichen Lösungsvektoren des Optimierungsproblems führt in jedem Fall zu einem lokalen Optimum, dennoch nicht immer zum globalen Optimum. Aufgrund dieser Tatsache müssen verschiedene Verfahren für Selektion, Kombination und Mutation betrachtet und ausgewertet werden. Diese Projektarbeit soll sich mit der Auswertung dieser Faktoren beschäftigen und die spezifischen Diversitäten aufzeigen und diskutieren. In diesem Beleg werden aufgrund der Aufgabenstellung Minimierungsprobleme behandelt werden.

Zur Realisierung des genetischen Algorithmus wurde die Programmiersprache C# gewählt. Der implementierte Algorithmus lässt sich über eine Win32-Formular-Anwendung steuern und erlaubt es zur Laufzeit Entscheidungen über die Wahl der Methoden und Information zu treffen. Um das Benchmarking zu unifizieren, wurde eine anfänglich erzeugte Population als Datensatz abgelegt. Diese Population dient bei jedem Programmdurchlauf als Ausgangszustand für den genetischen Algorithmus. Durch diesen gemeinsamen Startpunkt zur Bestimmung der Lösungsvektoren kann eine Aussage über die Bewertung der vorgegebenen Benchmarkingfunktionen getroffen werden. Aufgrund der Komplexität der Testverfahren wurde auf eine Beobachtung der reellen Codierung verzichtet und die binäre Codierung ausführlicher behandelt.

Der genetische Algorithmus hat dabei die folgenden Methoden implementiert:

1.1 Binärdarstellung der Lösungsvektoren

Ein Gen repräsentiert genau eine Variable für die Lösung eines Problems.

| | |
|-------|---------------------------|
| Gen A | 101100101100101011100101 |
| Gen B | 1111111100000110000011111 |

Dabei werden die Gene aneinandergereiht, sodass die Summe der Gene ein Genom bilden. Die reelle Repräsentation einer Variable in einem binären String wird anhand der Granularität bestimmt. Die Granularität wiederum ergibt sich aus den gegebenen Intervallen zur Berechnung eines Problems. So wird bei einer Binärstringlänge von 10 und einem Intervall von [0,1024] eine Granularität [g] von ~ 1,0009 nach der folgenden Formel erzielt:

$$g = \frac{\text{obere Schranke} - \text{untere Schranke}}{2^k - 1}$$

Dies entspricht der reellen Repräsentation des Abstandes zwischen zwei binären Darstellungen.

| | | | | | | | |
|---|---|---|---|---|---|---|----------|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | = 1,0009 |
| | | | | | ↓ | | |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | = 2,0018 |

Tab. 1 - Binärrepräsentation

1.2 Mutationsverfahren

Es wurden konstante, lineare und exponentielle Mutationsverfahren implementiert. Bei einer konstanten Mutationsrate werden in Abhängigkeit der Rate 30 % harte Mutationen ausgeführt. Eine harte Mutation ist das zufällige Vertauschen zweier Werte innerhalb des Genoms, wohingegen eine normale Mutation nur das Umdrehen eines Teilstückes innerhalb des Genoms bewirkt.

Die lineare und exponentielle Mutationsrate ermöglichen den Anstieg der Mutationsrate zwischen 2 Punkten im Wertebereich. Wird die Mutationsrate zu hoch gewählt, so kann das Verfahren in einer primitiven Zufallsuche der Lösungsvektoren münden.

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| | | | | ↓ | | |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 |

Tab. 2 – Beispiel Bitflip

Bei einem Bitflip wird das Genom an fest definierten Positionen invertiert.

1.3 Selektionsverfahren

Die Selektion ist ein Zustand im genetischen Algorithmus, indem Genome einer Eltern- bzw. Kindgeneration für die spätere Rekombination gewählt werden. Dabei kann zufällig gewählt oder aber eine Fitnessfunktion zur Bewertung der Individuen eingesetzt werden. Individuen mit einem hohen Fitnesswert werden somit bevorzugt bei einer Rekombination behandelt (sortierte Generation). Weiterhin wird in diesem Beleg zwischen zwei Wahlverfahren unterschieden.

1.4 Wahlverfahren

Als Wahlverfahren werden die umweltbasierte sowie die elternbasierte Selektion vorgehalten. In einer umweltbasierten Selektion werden unter Umständen nicht alle Individuen der Elterngeneration durch Individuen der Kindgeneration ersetzt. Genome können somit als Ganzes überleben. Bei einer elternbasierten Selektion hingegen wählt immer eine bestimmte Anzahl der besten Individuen und sieht diese für die weitere Rekombination vor.

1.5 N-Punkt-Rekombination (Cross-over)

Durch eine N-Punkt-Rekombination werden ab einer bestimmten Position auf dem Genom respektive Bit zweier Eltern so selektiert, dass die beiden Kinder so erzeugt, dass ein Kind an n Punkten durch Vater und Mutter gekreuzt werden und somit ein neues Individuum aus den beiden Elternteilen entsteht.



Abb. 1 - Cross-over

2 Benchmarking

Dieser Beleg wird Einflüsse des implementierten genetischen Algorithmus und seiner Parameter auf die C-Funktion aufzeigen und erläutern.

Die Populationsgröße wird, sofern nicht explizit erwähnt immer mit 500 angegeben.

Die Granularität wird möglichst in Abhängigkeit zum Intervall so gewählt, dass sich die dezimale Repräsentation des Binärstrings möglichst an 1 annähert.

2.1 Standardwerte

Zu Beginn des Benchmarking unterstützt ein allgemeiner Überblick für drei ausgewählte Funktionen die Wahl der Herangehensweise an ein spezifisches Problem. Jedes Problem wird mit gleichen Anfangsbedingungen gestartet. Auch die Elternpopulation ist zu Beginn immer identisch. Die Standardwerte haben die folgende Aufstellung:

- Elternpopulation = 100
- Kindpopulation = 100
- Intervall = [-20, 30]
- Gene (n) = 5
- Binärstringlänge (k) = 10
- Anteil der Mutation bei der Rekombination = 25 %
- Rekombinationspunkte = 5
- Generationen = 500
- Selektion = keine

Folgend wurden für die Griewank-, Ackley- und C-Funktion für die 500 Generationen berechnet und drei Parameter gegenübergestellt. Diese drei Parameter sind, der Verlauf der besten Fitness über die Generationen, die 10 besten Individuen nach 500 Generationen und eine grafische Gegenüberstellung.

2.2 Griewank

$$f(x) = 1 + \sum_{i=1}^n \frac{x^2}{400n} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right), \quad [-20, 30], \quad 1 \leq i \leq n$$

| Verlauf der besten Fitness: | |
|-----------------------------|-----------------|
| [1] 25,343 | [495] 100,28511 |
| [2] 25,343 | [496] 85,67937 |
| [3] 23,52051 | [497] 108,78136 |
| [4] 15,90393 | [498] 98,7944 |
| [5] 13,5686 | [499] 80,16455 |
| [...] | [500] 91,9029 |

Tab. 3 - beste Fitness über Generationen

| Die 10 besten Individuen: | |
|---------------------------|----------|
| Wert: | 13,5686 |
| Wert: | 14,14956 |
| Wert: | 15,19225 |
| Wert: | 15,90393 |
| Wert: | 18,71131 |
| Wert: | 18,71131 |
| Wert: | 19,16107 |
| Wert: | 22,95536 |
| Wert: | 23,52051 |
| Wert: | 24,07136 |

Tab. 4 - beste Individuen

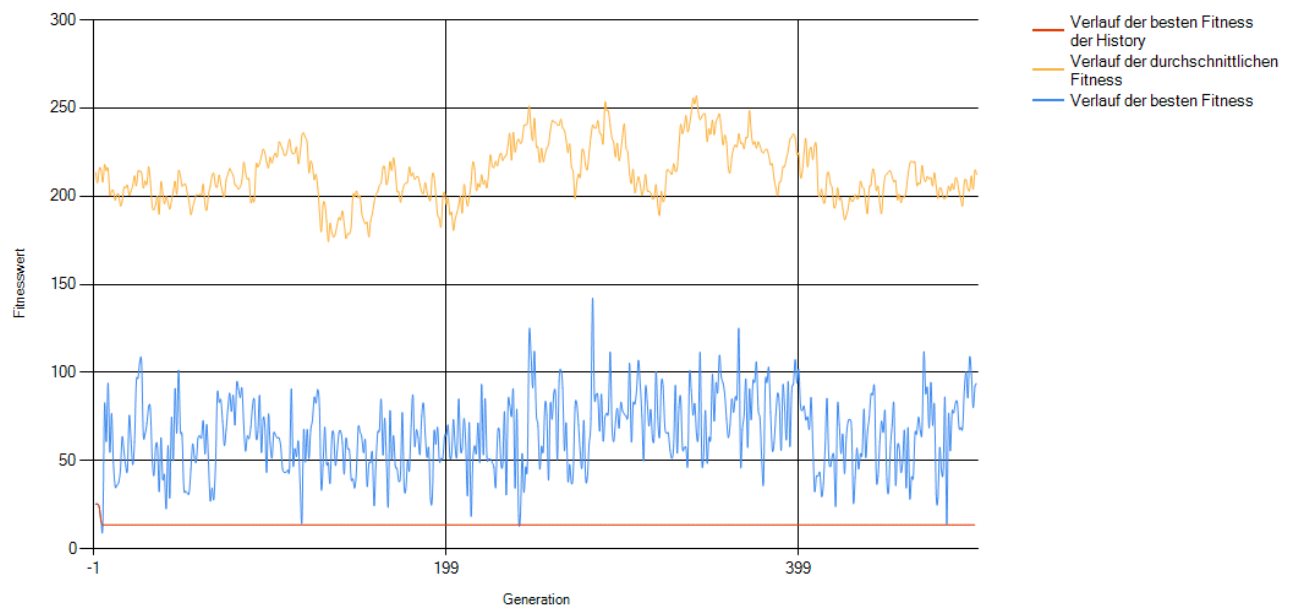


Abb. 2 - Griewank

Wie in Abb. 2 zu sehen ist, fluktuiert die Griewank-Funktion immer sehr stark in ihrer durchschnittlichen Fitness. Durch die natürliche Beschaffenheit der Griewank-Funktion ist dies keine Auffälligkeit, doch ist das Auffinden des globalen Minimums in dieser Konstellation sehr stark an den Zufall geknüpft. Wir haben uns daher nicht für die Griewank-Funktion entschieden.

2.3 Ackley

$$f(x) = 20 + \exp(1) - 20 \exp \left(-0,2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} * \sum_{i=1}^n \cos[2\pi x_i] \right), [-20,30], 1 \leq i \leq n$$

Verlauf der besten Fitness:

| | |
|--------------|----------------|
| [1] 15,36069 | [495] 59,46562 |
| [2] 84,9218 | [496] 59,46562 |
| [3] 64,67905 | [497] 69,18377 |
| [4] 58,02379 | [498] 99,93483 |
| [5] 58,02379 | [499] 43,5321 |
| [...] | [500] 59,37602 |

Tab. 5 - beste Fitness über Generationen

Die 10 besten Individuen:

| |
|----------------|
| Wert: 13,13946 |
| Wert: 15,36069 |
| Wert: 15,51808 |
| Wert: 17,28576 |
| Wert: 18,33659 |
| Wert: 18,70316 |
| Wert: 22,4666 |
| Wert: 22,71098 |
| Wert: 24,68231 |
| Wert: 25,8146 |

Tab. 6 - beste Individuen

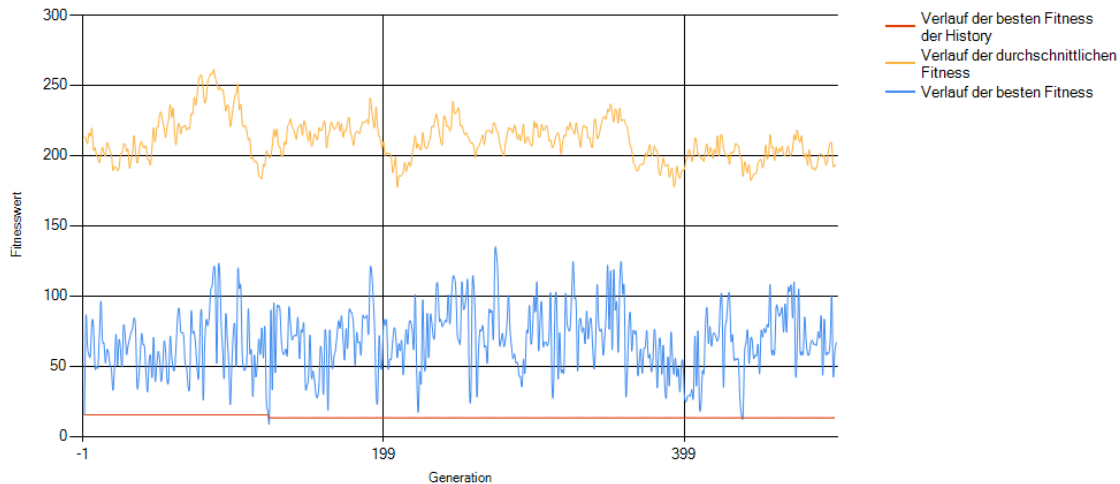


Abb. 3 - Ackley

In der Abbildung und auch in den vorher dargestellten Listen ist zu sehen, dass der geschriebene Algorithmus bei der Anwendung auf die Ackley Funktion schon sehr zeitig ein gutes Ergebnis (15,36069) erreicht (Tab. 6 - Zeile 1). Dieses ist auch nach 500 erzeugten Generationen das zweit beste Ergebnis. Außerdem ist in der Abbildung zu erkennen, die Werte der besten und auch der durchschnittlichen Fitness über die Generationen hinweg sehr stark schwankt. Aufgrund dieses Verhaltens haben wir uns gegen die Ackley-Funktion zum weiteren Testen unseres Algorithmus entschieden.

2.4 C-Funktion

$$f(x) = 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{|x_j - x_i|}{j - i}, [-20, 30], 1 \leq i, j \leq n$$

Verlauf der besten Fitness:

| | |
|--------------|----------------|
| [1] 61,33105 | [495] 52,80222 |
| [2] 55,87325 | [496] 51,01011 |
| [3] 72,33627 | [497] 61,09482 |
| [4] 65,43662 | [498] 33,3985 |
| [5] 70,9596 | [499] 28,69013 |
| [...] | [500] 25,48061 |

Tab. 7 - beste Fitness über Generationen

Die 10 besten Individuen:

| | |
|-------|----------|
| Wert: | 11,36364 |
| Wert: | 16,48746 |
| Wert: | 17,57087 |
| Wert: | 17,57902 |
| Wert: | 17,75008 |
| Wert: | 18,76018 |
| Wert: | 18,82535 |
| Wert: | 19,80287 |
| Wert: | 20,80482 |
| Wert: | 21,3506 |

Tab. 8 - beste Individuen

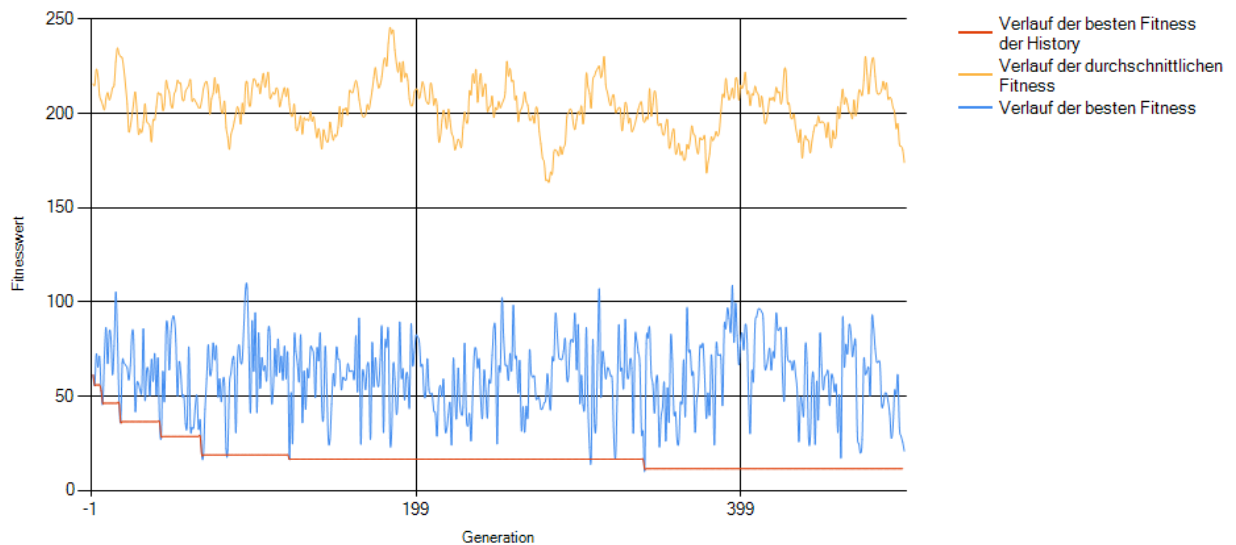


Abb. 4 - C-Funktion

Die C-Funktion scheint sich in dieser Testkonstellation im Vergleich zu den vorhergehenden Funktionen stärker anzunähern. Auch die beste Fitness der Individuen nähert sich über die Generationen eindeutig an und verschlechtert sich nicht wie im Vergleich zu den vorher genannten Funktionen. Aufgrund dieser Beschaffenheit haben wir uns für das Benchmarking der C-Funktion entschieden.

3 Die C-Funktion

3.1 Untersuchung Größe der Anfangspopulation

Einflüsse durch die Wahl der Anfangspopulation auf ein Problem wird in verschiedene Klassengrößen von Individuen aufgeteilt und bewertet. Um die Anfangspopulation zu untersuchen, wurde das folgende Szenario aufgestellt:

- Mutationsrate = 0 %
- Vier-Punkt-Rekombination
- Kindpopulation = 600
- Binärstringlänge = {10; 15}
- n = {3; 12}
- Generationsanzahl = 50
- Selektionsverfahren = umweltbasierte Kommaselektion

Zum Selektionsverfahren ist zu sagen, dass die umweltbasierte Kommaselektion durch zufällige Auswahl der Individuen realisiert wurde.

3.1.1 Kleine Klasse 50

Die Größe der Anfangspopulation von 50 nähert sich sehr schnell dem Optimum von null an. Das Optimum wurde bei n=3 und der Binärstringlänge = 10 bereits nach 11 Generationen erreicht.

| Beste Individuen | | Durchschnittliche Fitness | |
|----------------------|---------|---------------------------|----------|
| Gen1; Gen2; Gen3; | Wert | Generation; | Fitness |
| 8,338; 8,338; 8,338; | 0 | [1]; | 36,12317 |
| 8,358; 8,358; 8,358; | 0 | [2]; | 10,90187 |
| 8,319; 8,319; 8,319; | 0 | [3]; | 5,28549 |
| 8,299; 8,299; 8,299; | 0 | [4]; | 3,35831 |
| 8,162; 8,162; 8,162; | 0 | [5]; | 1,63423 |
| 8,143; 8,143; 8,143; | 0 | [6]; | 1,02313 |
| 8,299; 8,299; 8,280; | 0,05865 | [7]; | 0,78349 |
| 8,358; 8,338; 8,338; | 0,05865 | [8]; | 0,54111 |
| 8,299; 8,280; 8,280; | 0,05865 | [9]; | 0,35328 |
| 8,319; 8,338; 8,338; | 0,05865 | [10]; | 0,19778 |

Tab. 9 - Vergleich Anfangspopulation 50

3.1.2 Große Klasse 100

Steigert man die Anfangspopulation auf 100, so wird bei der Mutationslosen Vier-Punkt-Rekombination erwartet, dass nach 50 Generationen deutlich mehr Individuen das Optimum erreicht haben. Es konnte unter den gleichen Voraussetzungen beobachtet werden, dass die Anfangspopulation von 100 zum Ende 16 Individuen mit einem Optimum von null erreicht haben. Bei der vorhergehenden Testreihe von 50 konnten nur 6 nach der Generationszahl das Optimum erreichen.

| Beste Individuen | | Durchschnittliche Fitness | |
|----------------------|---------|---------------------------|----------|
| Gen1; Gen2; Gen3; | Wert | Generation; | Fitness |
| 8,299; 8,299; 8,299; | 0 | [1]; | 34,25943 |
| 8,319; 8,319; 8,319; | 0 | [2]; | 10,07032 |
| 8,240; 8,240; 8,240; | 0 | [3]; | 5,41393 |
| 8,456; 8,456; 8,456; | 0 | [4]; | 3,30939 |
| 8,338; 8,338; 8,338; | 0 | [5]; | 2,13852 |
| 8,475; 8,475; 8,475; | 0 | [6]; | 1,38738 |
| 8,358; 8,358; 8,358; | 0 | [7]; | 0,78944 |
| 8,221; 8,221; 8,221; | 0 | [8]; | 0,48157 |
| 8,397; 8,397; 8,397; | 0 | [9]; | 0,27071 |
| 8,416; 8,416; 8,416; | 0 | [10]; | 0,19819 |
| 8,377; 8,377; 8,377; | 0 | | |
| 8,162; 8,162; 8,162; | 0 | | |
| 8,436; 8,436; 8,436; | 0 | | |
| 8,143; 8,143; 8,143; | 0 | | |
| 8,201; 8,201; 8,201; | 0 | | |
| 8,182; 8,182; 8,182; | 0 | | |
| 8,299; 8,299; 8,319; | 0,05865 | | |

Weiterhin konnte gezeigt werden, dass bei ansteigendem n deutlich mehr Generationen benötigt werden, bis sich ein Optimum einstellt. So wurde bei n=12, Binärstringlänge = 15 und einer Anfangspopulation von 50 nach 22 Generationen sowie bei einer Anfangsgröße von 100 nach 23 Generationen ein Optimum erreicht, das in der Testzusammenstellung stabil blieb. Die Anfangspopulation von 50 erzielt dabei aber ein deutlich schlechteres Optimum. Im Testszenario scheint dieser Effekt allerdings zufällig zu existieren, da bei einer umweltbasierten Komma-Selektion kein Einfluss auf die Geschwindigkeit oder Wertigkeit des Optimums gezeigt werden konnte.

3.2 Untersuchung Selektionsverfahren Elterngeneration

Zur Untersuchung der Einflüsse der Wahl der Selektionsverfahren wird zwischen Komma- und Plusselektion verglichen. Zusätzlich wird bewertet, welchen Einfluss Eltern- und Umweltselektion auf die beiden Verfahren haben.

3.2.1 Komma vs. Plus

Unter Betrachtung zweier Messungen, mit einer Anfangspopulation von 100 Eltern und 600 erzeugten Kindern konnte kaum ein Unterschied im Verlauf der besten Individuen erkannt werden, beide Verfahren bewirken eine logarithmische Annäherung an das Optimum.

| Entwicklung der Beste Individuen (umweltbedingte Komma-Selektion) | | Entwicklung der Beste Individuen (umweltbedingte Plus-Selektion) | |
|--|------------|---|------------|
| Generation; | Fitness | Generation; | Fitness |
| [1]; | 9702,07507 | [1]; | 9702,07507 |
| [2]; | 9159,50036 | [2]; | 8663,52349 |
| [3]; | 5885,2424 | [3]; | 6957,41039 |
| [4]; | 5687,39295 | [4]; | 5656,0594 |
| [5]; | 5647,12247 | [5]; | 4530,32483 |
| [6]; | 3795,93831 | [6]; | 4530,32483 |
| [7]; | 2946,6464 | [7]; | 4333,68493 |
| [8]; | 2946,6464 | [8]; | 3393,18964 |
| [9]; | 2608,38077 | [9]; | 3139,25066 |
| [10]; | 2164,54458 | [10]; | 2912,73633 |
| [...] | | [...] | |
| [40]; | 246,35296 | [40]; | 417,19784 |
| [41]; | 244,18443 | [41]; | 385,72011 |
| [42]; | 237,56855 | [42]; | 360,79275 |
| [43]; | 231,52276 | [43]; | 345,94537 |
| [44]; | 231,20872 | [44]; | 331,94957 |
| [45]; | 229,98125 | [45]; | 263,28906 |
| [46]; | 226,38458 | [46]; | 229,94195 |
| [47]; | 223,98378 | [47]; | 194,6139 |
| [48]; | 221,45306 | [48]; | 192,8966 |
| [49]; | 220,09542 | [49]; | 192,4303 |
| [50]; | 218,69311 | [50]; | 175,63274 |

Tab. 10 - Selektionsverfahren Komma vs. Plus

Allerdings ist festzustellen, dass beide Verfahren eine deutlich schnellere und klarere Annäherung an das Optimum erreichen als ohne Einsatz eines Selektionsverfahrens.

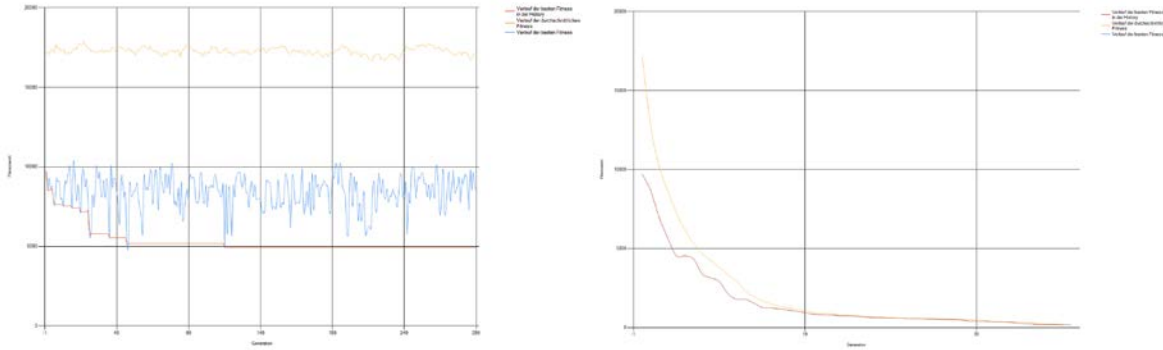


Abb. 5 - Vergleich: Genreationserzeugung ohne Selektionsverfahren (links) und umweltbedingten Plus-Selektion (rechts)

3.2.2 Einfluss der Umwelt / Elternselektion

3.2.2.1 Elternselektion

Für die Elternselektion wurde eine zufällige Auswahl von Eltern für die Generationserzeugung gewählt. Dieses Verfahren wurde im Laufe der Tests nicht weiter untersucht.

3.2.2.2 Umweltselektion

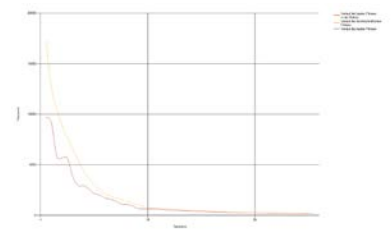
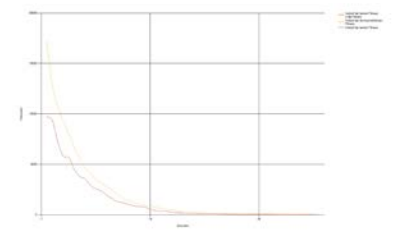
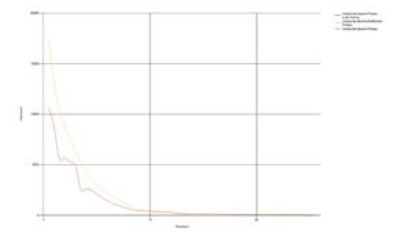
In Bezug auf die Umweltselektion wurde verschiedene Verfahren implementiert:

- Zufällige Auswahl aus der Selektionsmenge (welche abhängig vom Selektionsverfahren ist)
- Nach fitnesssortierter Auswahl aus der Selektionsmenge

Hierbei wird die Selektionsmenge nach Ihren Fitnesswerten sortiert und es werden anschließend die n besten daraus gewählt.

- q-fach Turnier Auswahl aus der Selektionsmenge

Bei der q-fach Turnier Auswahl werden aus der Selektionsmenge jeweils zufällig 2 Individuen gewählt und anschließend wird das besten der beiden als neues Kind übernommen.

| Entwicklung der Beste Individuen (zufällig umweltbedingte Komma-Selektion) | Entwicklung der Beste Individuen (fitnessorientierte umweltbedingte Komma-Selektion) | Entwicklung der Beste Individuen (q-fach Turnier umweltbedingte Komma-Selektion) |
|---|--|--|
| Generation; Fitness [1]; 9702,07507 [2]; 9159,50036 [3]; 5885,2424 [4]; 5687,39295 [5]; 5647,12247 [6]; 3795,93831 [7]; 2946,6464 [8]; 2946,6464 [9]; 2608,38077 [10]; 2164,54458 [...] [40]; 246,35296 [41]; 244,18443 [42]; 237,56855 [43]; 231,52276 [44]; 231,20872 [45]; 229,98125 [46]; 226,38458 [47]; 223,98378 [48]; 221,45306 [49]; 220,09542 [50]; 218,69311 | Generation; Fitness [1]; 9702,07507 [2]; 9373,10389 [3]; 7268,37808 [4]; 5798,64874 [5]; 5695,75927 [6]; 4543,06778 [7]; 3801,42317 [8]; 3576,60646 [9]; 2926,09632 [10]; 2568,85587 [...] [40]; 47,77346 [41]; 46,60248 [42]; 46,08782 [43]; 43,34779 [44]; 42,68446 [45]; 40,99787 [46]; 40,46051 [47]; 39,11252 [48]; 38,84574 [49]; 38,15589 [50]; 35,9267 | Generation; Fitness [1]; 10598,86645 [2]; 8779,17035 [3]; 5589,26876 [4]; 5589,26876 [5]; 5273,11198 [6]; 4948,07799 [7]; 2552,30036 [8]; 2552,30036 [9]; 2507,41385 [10]; 2097,73016 [...] [40]; 68,59531 [41]; 68,55556 [42]; 63,13983 [43]; 58,46126 [44]; 51,31919 [45]; 51,23968 [46]; 49,78175 [47]; 46,00332 [48]; 41,92381 [49]; 41,92381 [50]; 41,92381 |
|  |  |  |

Tab. 11 - Auswertung eltern, umwelt und q-fach Turnier Selektion

Anhand der Ergebnisse wird deutlich, dass sowohl das fitnessorientierte als auch das q-fach Turnier Wahlverfahren zu weit aus besseren Ergebnissen führt. Das Optimum wird bei allen Verfahren ungefähr gleich schnell erreicht (nach ~ 25 Generationen). Im Vergleich zwischen dem fitnessorientierten und dem q-fach Turnier Verfahren werden ähnliche Optima erreicht. Allerdings führt das fitnessorientierte Verfahren zu einem leicht besseren Ergebnis. Woraus sich ableiten lässt, dass dieses Verfahren etwas besser geeignet ist.

3.3 Untersuchung Cross-over-Positionierung (N-Punkt-Rekombination)

Für die Untersuchung der Cross-over-Positionierung haben wir uns für die Einteilung in zwei Testklassen entschieden. Getestet wurden in der ersten Klasse die Vier-Punkt-Rekombination und in der zweiten Klasse eine Zehn-Punkt-Rekombination. Auf einem Genom mit einer Binärstringlänge von 15 und 12 Genen kann somit an 180 möglichen Stellen rekombiniert werden.

Das Testszenario hatte weiterhin die folgenden Parameter:

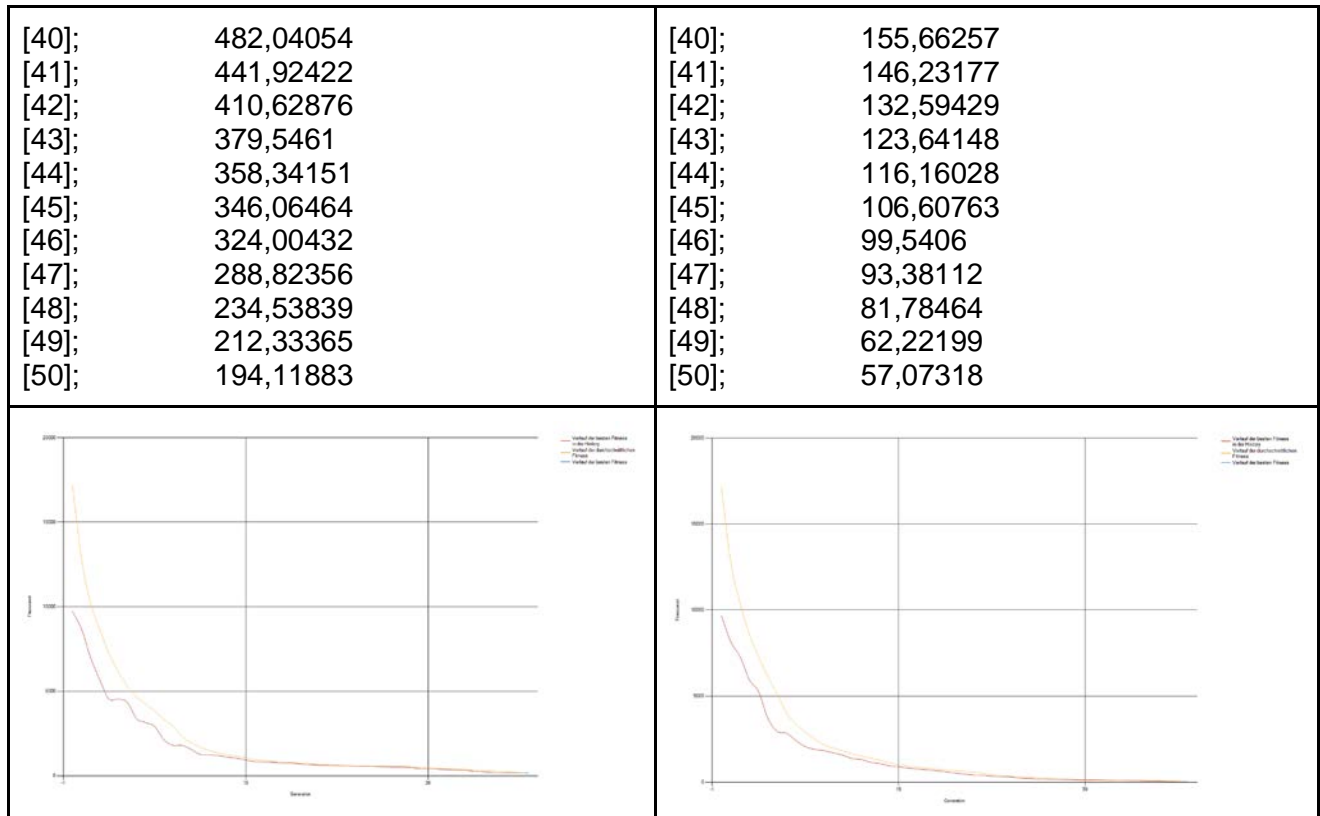
- Anfangspopulation = 100
- Kindpopulation = 600
- n = 12
- Binärstringlänge = 15
- Mutationsrate = 50 %
- Selektionsstrategie = umweltbasierte und elternbasierte Kommaselektion

Weiterhin ist anzuführen, dass bei, wenn bei einer Mutationsrate von 50 % eine Mutation eintritt, dann handelt es sich mit 30 % um eine harte Mutation. Es wird aufgrund der Mutationsrate eine Fluktuation der Messwerte erwartet, doch sollte diese kaum Einfluss auf das zu erreichende Minimum haben.

3.3.1 Klasse 4 vs 10 umweltbasierte Plus-Selektion

Im Vergleich der Vier-Punkt-Rekombination mit der Zehn-Punkt-Rekombination wurde im Ersten schritt die umweltbasierte Plus-Selektion als Selektionsverfahren festgehalten. Es konnte beobachtet werden, dass sich beide Verfahren annähernd gleichschnell auf ein Minimum bewegen. Die Zehn-Punkt-Rekombination wurde allerdings zwischenzeitlich schneller als die Vier-Punkt-Rekombination. In der Tabelle wurde die durchschnittliche Fitness über die 50 Generationen ausschnittsweise abgetragen.

| Klasse 4 | Klasse 10 |
|------------------------|---------------------|
| Generation; Fitness 60 | Generation; Fitness |
| [1]; 17145,98355 | [1]; 17145,98355 |
| [2]; 12775,9163 | [2]; 12775,21926 |
| [3]; 10176,14671 | [3]; 10379,54121 |
| [4]; 8624,2993 | [4]; 8619,14876 |
| [5]; 7248,14775 | [5]; 7271,93989 |
| [6]; 6175,16031 | [6]; 6134,15825 |
| [7]; 5299,21442 | [7]; 5097,23164 |
| [8]; 4669,15225 | [8]; 4008,47753 |
| [9]; 4244,2353 | [9]; 3379,81579 |
| [10]; 3822,32539 | [10]; 2893,04076 |
| [...] | [...] |



Tab. 12 - Auswertung 4 vs. 10-Punkt-Rekombination umweltbasiert

Es kann aufgrund der Beschaffenheit der Rekombinationsarten geschlussfolgert werden, dass sich bei nur vier Punkten, die zufällig auf dem Genom gewählt werden etwas langsamer ein Minimum einstellen kann als bei der zufälligen Wahl von zehn Punkten. Es wird weiterhin erwartet, dass die elternbasierte Plus-Selektion einen signifikanteren Verlauf aufweisen sollte.

3.3.2 Klasse 4 vs 10 elternbasierte Plus-Selektion

Bei einer elternbasierten Plus-Selektion hingegen konnte beobachtet werden, dass die Erwartung aus dem vorhergehenden Test leider nicht eingetreten ist. Die Zehn-Punkt-Rekombination scheint sich in diesem Szenario sogar reziprok zur vorhergehenden Beobachtung zu verhalten. So stellt sich ein Minimum später als bei der Vier-Punkt-Rekombination ein.

| Klasse 4 | Klasse 10 |
|--|---|
| Generation; Fitness 76 [1]; 17145,98355 [2]; 12760,56859 [3]; 10538,99111 [4]; 8674,09464 [5]; 7231,33182 [6]; 6038,84127 [7]; 4901,2511 [8]; 3739,1096 [9]; 2927,39595 [10]; 2574,77438 [...] [40]; 177,41499 [41]; 166,08769 [42]; 153,10342 [43]; 146,04708 [44]; 142,75243 [45]; 136,91161 [46]; 133,69387 [47]; 127,36515 [48]; 125,08317 [49]; 122,07554 [50]; 117,85835 | Generation; Fitness [1]; 17145,98355 [2]; 13067,88647 [3]; 10726,59565 [4]; 8922,11747 [5]; 7438,60664 [6]; 5966,9288 [7]; 4935,89269 [8]; 3749,7588 [9]; 3071,90693 [10]; 2802,29056 [...] [40]; 245,31816 [41]; 232,62366 [42]; 208,63695 [43]; 198,39334 [44]; 191,36063 [45]; 182,7288 [46]; 179,80822 [47]; 175,3975 [48]; 170,46099 [49]; 165,01422 [50]; 158,11834 |
| | |

Tab. 13 - Auswertung 4 vs. 10-Punkt-Rekombination elternbasiert

Diese Ergebnisse lassen nun eine neue Schlussfolgerung zu. So hat die im Szenario benutzte Mutationsrate von 50 % einen stärkeren Effekt als vorher erwartet. Weiterhin scheint aufgrund der Messwerte ein Einfluss der elternbasierten Selektion auf die Vier-Punkt-Rekombination sowie ein Einfluss der umweltbasierten Selektion auf die Zehn-Punkt-Rekombination zu existieren. Leider konnten aufgrund der Menge an Tests keine stochastische Auswertung über die Messwerte ermittelt werden. Durch dieses Verfahren hätte diese Fluktuation noch genauer beurteilt werden können.

3.4 Untersuchung Mutationsrate

Zum Test des Einflusses der konstanten Mutationsrate wurden für jeden Testdurchlauf jeweils eine Mutationsrate von 0 % und 50 % angegeben.

Generell führte das Anheben der Mutationsrate zum schnelleren finden eines Optimums.

3.4.1 Untersuchung generationsabh. Mutationsrate

Die Abhängigkeit der Mutationsrate wurde mithilfe der:

1. Linearen Funktion: $f(g) = m_0 + \left(\frac{m_n - m_0}{n}\right) * g$
2. Exponentiellen Funktion: $f(g) = m * e^{\left(\frac{g}{n}\right) * \ln\left(\frac{m_n}{m_0}\right)} = m_0 * \left(\frac{m_n}{m_0}\right)^{\frac{g}{n}}$

Welche jeweils mit den Parametern $m_0 = 10$, $m_n = 90$ und $g = 50$ ausgeführt wurde. Folgende Darstellung verdeutlicht den Verlauf der jeweiligen Funktionen.

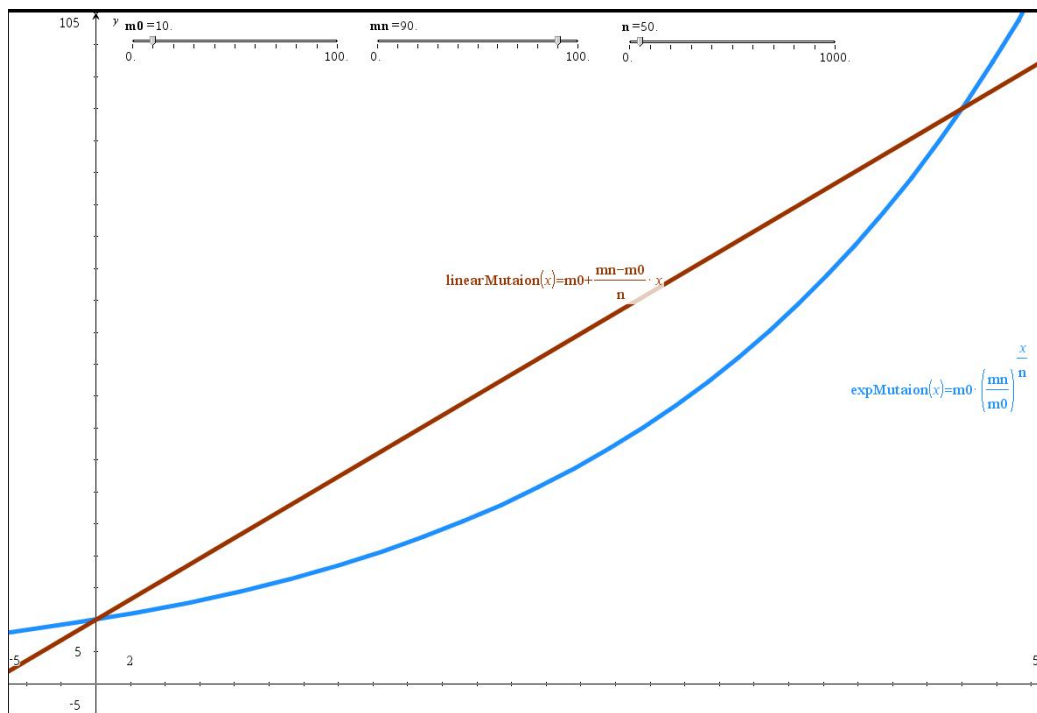


Abb. 6 - Vergleich Mutationsraten

Diese wurden zum Test auf ausgewählte Generationserzeugungen angewendet. Auffällig war bei diesem Test, dass bei Verwendung der generationsabhängigen Mutationsraten, die lineare bei deterministisch Plus-selektierten Generationen zu einem besseren Optimum führte als die exponentiell verlaufende. Wohingegen bei deterministisch Komma-selektierten Generationen, dies genau umgekehrt der Fall war. Jedoch beide Mutationsraten führten zu besseren Ergebnissen als eine konstante Mutation.

| | konst. Mutation | lineare Mutation | exp. Mutation |
|---|---|--|--|
| Beste Individuen (deterministische Komma-Selektion) | Individuum;Wert [1];257,38572 [2];257,40056 [3];257,43418 [4];257,44902 [5];257,48176 [6];257,4966 [7];257,49748 [8];257,50446 [9];257,5193 [10];257,5311 [...] | Individuum;Wert [1];136,49087 [2];137,52031 [3];137,7429 [4];137,82752 [5];137,84361 [6];137,94417 [7];137,95788 [8];138,06213 [9];138,20018 [10];138,46437 [...] | Individuum;Wert [1];37,04543 [2];41,16792 [3];41,31473 [4];41,38484 [5];41,4585 [6];41,63204 [7];42,31895 [8];42,34369 [9];42,34674 [10];42,43794 [...] |
| Beste Individuen (deterministische Plus-Selektion) | Individuum;Wert [1];608,71802 [2];608,78795 [3];608,88507 [4];608,91446 [5];608,92561 [6];608,94241 [7];608,98439 [8];608,99717 [9];609,01233 [10];609,05213 [...] | Individuum;Wert [1];37,2167 [2];42,47519 [3];43,3764 [4];44,39122 [5];46,92011 [6];47,08166 [7];47,60723 [8];48,35662 [9];49,08255 [10];49,46775 [...] | Individuum;Wert [1];290,45013 [2];291,42072 [3];291,66162 [4];292,16351 [5];292,34046 [6];292,41742 [7];292,70205 [8];293,0087 [9];293,34063 [10];293,468 [...] |

Tab. 14 - Auswertung Mutationsraten

4 Auswertung

4.1 Auffällige Parameter

Aufgrund der Beobachtungen scheint die Wahl des n bei der N-Punkt-Rekombination keinen signifikanten Einfluss auf die Geschwindigkeit oder die Wertigkeit des Ergebnisses zu haben. Die C-Funktion scheint nicht von dem Einfluss der N-Punkt-Rekombination zu profitieren. Es wird vermutet, dass die anderen Probleme der Aufgabenstellung besser beeinflussbar durch die Rekombinationsverfahren sind, denn die C-Funktion ist es nicht.

Weiterhin konnte beobachtet werden, dass der Einsatz eines Selektionsverfahren einen großen Einfluss auf die Bildung eines Optimums hat. Im Beispiel der C-Funktion stellte sich die Anwendung der fitnessorientierten Komma-Selektion als das effektivste Verfahren heraus.

4.2 Randeffekte

- Zufall

Aufgrund der großen Abhängigkeit der Verfahren von zufällig erzeugten Zahlen ist zu beachten, dass alle dargestellten Ergebnisse bei weiteren Testläufen zu besseren beziehungsweise schlechteren Ergebnissen führen kann. Daher sind gerade, im Verhältnis, kleinere Unterschiede kritisch zu betrachten und es kann somit nicht verallgemeinert gesagt werden, dass zum Beispiel eine Komma-Selektion besser ist als eine Plus-Selektion.

- Genpoolausdünnung

Aufgrund der speziellen Implementierung, welche sowohl bei Rekombination als auch bei Mutation versucht keine doppelten Kinder zu erzeugen. Kommt es Verlauf der Generation zu einer Ausdünnung des Genpools. Dies hat zur Folge, dass nach einer gewissen Anzahl an Generationen die Erzeugung neuer unbekannter Kinder immer stärker abnimmt. Am stärksten war dieser Effekt bei der Anwendung der Plus- beziehungsweise Komma-Selektion mit einer konstanten Mutationsrate von 0 % zu merken.

Quellenverzeichnis

Vorlesungsscript und Mitschriften aus dem Modul evolutionäre Algorithmen, Dozent: Dr. habil. Wolfgang Kliesch