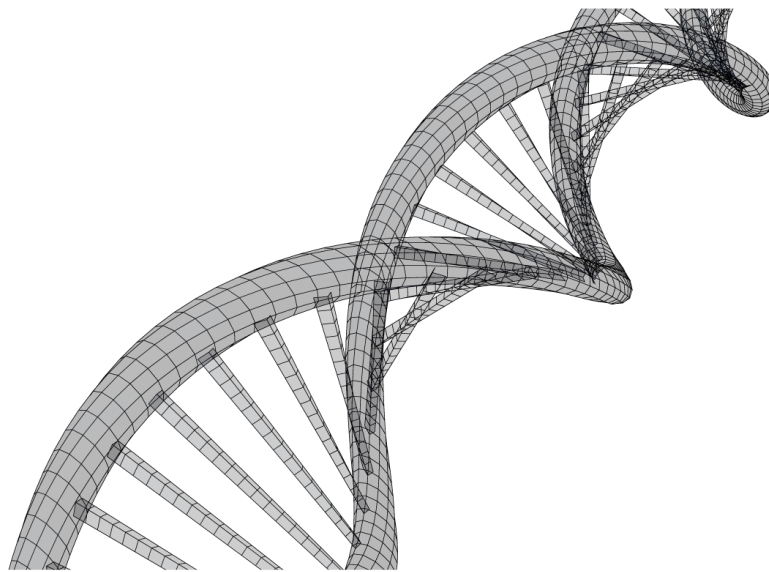


Universität Hamburg
Department Informatik
Knowledge Technology, WTM

Evolutionary computation and the creativity in digital evolution



Proseminararbeit

Für welche Aufgaben lassen sich die Konzepte der
Evolution in der Informatik nutzen?

Christoph Wolfes
christoph.wolfes@informatik.uni-hamburg.de

Abstract

In der folgenden Seminararbeit werden Experimente im Bereich evolutionärer Algorithmen, im folgenden EA abgekürzt, vorgestellt und im Hinblick auf die Lösungsansätze der Implementierung untersucht. Dabei konzentriert sich die Arbeit auf überraschend kreative Lösungskandidaten die sich aus den vorgestellten Projekten herauskristallisierten, als auch die draus abgeleiteten Möglichkeiten, des Konzepts der Evolution, für die Informatik. Vorbereitend zum Thema werden außerdem die benötigten Grundlagen vermittelt und die weiteren Anwendungsgebiete vorgestellt. Abschließend werden die Ergebnisse zusammengefasst, Vor und Nachteile ausgearbeitet, sowie ein Ausblick auf das zukünftige Potential des Forschungsgebietes gegeben.

Inhaltsverzeichnis

1	Evolution als Algorithmus	2
1.1	Grundstein - The origin of species	2
1.2	Selektion, Mutation, Reproduktion	2
1.3	Definition evolutionäre Algorithmen	3
2	Eine Erste Einordnung	4
2.1	Warum Evolutionäre Algorithmen	4
2.2	Unterschied zu anderen Optimierungsalgorithmen	4
2.3	Implementierungsansätze	4
2.4	Anwendungsgebiete	5
3	Unerwartete Lösungsansätze	5
3.1	Antennendesign Mission ST-5	5
3.2	Alternative Fortbewegung von Robotern bei Beschädigung	6
3.3	Simulation von virtuellen Kreaturen	7
4	Evaluation und versteckte Potenziale	8
4.1	Zusammenfassung und Fazit	8
4.2	Nachteile	8
4.3	Vorteile	8
4.4	Abwägung des Einsatzes	9
5	Ausblick	9
	Bibliography	10

1 Evolution als Algorithmus

Die folgenden Abschnitte geben eine erste Einführung in Evolution als Algorithmus und legen die Grundlage für das Verständnis der später vorgestellten Projekte. Um aber zunächst der Frage nach der Nutzung von evolutionären Konzepten in der Informatik auf den Grund zu gehen, ist es zunächst sinnvoll, sich mit den grundlegenden Begriffen der klassischen Evolutionstheorie aus dem Forschungsgebiet der Biologie zu beschäftigen. Daraus ableitend können Konzepte analog auch auf die Informatik übertragen werden und als Methode in Algorithmen überführt werden. [10] Der folgende Abschnitt gibt hierbei eine Einführung in die Grundkonzepte der Evolutionstheorie nach Darwin.

1.1 Grundstein - The origin of species

Den Grundstein für die wissenschaftliche Theorie der Evolution wurde von den Wissenschaftlern Charles Darwin und Alfred Russel Wallace gelegt, welche über viele Jahre verschiedene Tierarten studierten und dabei einen Zusammenhang zwischen Arten herstellten. Ihre Beobachtungen veröffentlichten die beiden 1859, in dem Buch *the origin of species* [8] (1859). Zum Zeitpunkt der Veröffentlichung war die Abweichung von der kirchlichen Lehre besonders schwierig, weshalb es einige Zeit dauerte bis sich die Theorie in der Gesellschaft durchsetzte. [5] Das dort eingeführte Konzept, auch bekannt als “survival of the fittest” oder “natural selection”, liefert somit nicht nur die Grundlage für die Erforschung der Entstehung der Tier und Pflanzenwelt [8] sondern dient auch als Basis für EA in der Informatik die zusammen mit der dynamischen Programmierung eine Unterkategorie der Optimierungsalgorithmen darstellen [1].

1.2 Selektion, Mutation, Reproduktion

Im Folgenden werden die wichtigsten Rahmenbedingung der Evolutionstheorie die von Darwin und Wallace postuliert wurden zusammengefasst. Im Anschluss kann das Konzept der “natural selection” in einen Algorithmus überführt werden, welcher die Basis aller weiterer EA darstellt:

- Eine oder mehrere Populationen stehen in **Konkurrenz** um eine begrenzte Anzahl an Ressourcen
- Die Population verändert sich stetig durch die **Reproduktion** einzelner Individuen
- Ein Individuum wird durch Eigenschaften definiert, welche durch mehrere Generationen **vererbt** werden können
- Die Kinder eines Individuums haben ähnliche Eigenschaften sind aber niemals gleich (**Crossover**)

- Eigenschaften der Individuen werden gegen die Umgebung evaluiert (**Fitnesswert**) wobei ein höherer Fitnesswert die **Selektion** und somit die Reproduktion begünstigt.
- **Mutationen** verändern zufällig Eigenschaften des Individuums, welche den Fitnesswert positiv, negativ oder gar nicht beeinflussen.

Auf eine detailliertere Erklärung der Begriffe wurde aus Fokusgründen verzichtet, weiterführende Informationen finden sich in [5] und [10]

1.3 Definition evolutionäre Algorithmen

Mit den grundlegenden Begrifflichkeiten im Gepäck, kann nun ein EA skizziert werden, dafür wird zunächst ein kurzer Einblick in die Entstehung des Forschungsgebiets gegeben um ein breites Verständnis für die Arbeiten auf dem Gebiet zu erhalten. Bei einem evolutionären Algorithmus (EA) handelt es sich um Populationsbasierte, von der Natur inspirierte Algorithmen. [12] Da diese Art von Algorithmen aufgrund ihrer Struktur, die hauptsächlich auf der Simulation einer großen Population beruht, sehr rechenintensiv sind, ist das Forschungsgebiet der evolutionären Algorithmen noch recht jung und geht auf Arbeiten von Lawrence J. Fogel und John Holland in den 1960er Jahren zurück [10]A2.3. Durch die wachsende Berechnungskapazitäten, und großer Fortschritte im Bereich der Simulation ergeben sich eine Vielzahl von neuen Anwendungsmöglichkeiten. In der Literatur werden aus diesem Grund weiter differenzierte Implementierungsansätze unterschieden auf die im Abschnitt Implementierungsansätze näher eingegangen wird [12]. Der folgende Ablauf zeigt zunächst die grundlegenden Mechanismen die von Kenneth de Jong im Buch Evolutionary Computation - A Unified Approach beschrieben werden. [12]

- 1) $P_0 \leftarrow$ generate initial population of m **individuals**
- 2) Set generational counter $k = 1$
- 3) Evaluate P_0 for **fitness**
- 4) Begin iteration until termination (number of **generations** or termination criteria reached)
 - a) **Select** parents $P_{par} \leftarrow P_{k-1}$
 - b) Get offspring $P_{offsp.}$ by **recombining** parents
 - c) **Mutate** some offspring
 - d) **Select** population to survive unto next generation $P_k \leftarrow P_{k-1} \cup P_{offsp.}$
 - e) Iterate generation counter $k = k + 1$

Abbildung 1: Ablauf eines evolutionären Algorithmus [7]

2 Eine Erste Einordnung

2.1 Warum Evolutionäre Algorithmen

Der Begriff Evolution beschreibt einen natürlichen Optimierungsprozess der zwar nicht zwingend optimale Lösungen produziert, aber trotzdem komplexe und hoch angepasste Lösungen zu einem gegebenen Problem findet. Beispiele dieser angepassten Lösungen lassen sich in Vielzahl in der Natur beobachten: So schützt beispielsweise das Streifenmuster des Zebras dieses vor Infektion durch Moskitos und die Anordnung der Flügel der Hummel lassen diese fliegen. Auch die Evolution des Auges, ein Erfolgskonzept der Evolution, löst auf sehr viele verschiedene Weise immer wieder das selbe Problem. [10] Analog lässt sich dieses Phänomen aber auch in der Entwicklung technischer Systeme durch den Menschen wieder finden. Dort spricht man dann von Nature inspired technology wie es zB. der Lotuseffekt der sich bei dreckabweisenden Beschichtungen zu nutze gemacht wird. Ähnliche Herausforderung finden sich aber auch in der Lösung von Optimierungsproblemen in der Informatik, sowie der Programmierung von Intelligenten Maschinen. Im folgenden Abschnitt wird auf diese Herausforderungen näher eingegangen und die Anwendungsgebiete näher beleuchtet.

2.2 Unterschied zu anderen Optimierungsalgorithmen

2.3 Implementierungsansätze

Der Begriff der EA hat sich nach seiner Einführung in den 1960er Jahren weiter verfeinert. Wobei sich die drei Hauptrichtungen EA herauskristalisierten:

- Evolutionäre Programmierung (**evolutionary programming** EP)
- Evolutions Strategien (**evolution strategies** ES)
- Genetische Algorithmen (**genetic algorithms** GA)

Dabei unterscheidet man bei den Verfahren die Repräsentation der Individuen. Die Eigenschaften eines Individuums können beispielsweise in eine Binärzahl codiert werden. Durch diese Art der Repräsentation ergibt sich eine starke Analogie zur Genetik weshalb man häufig bei dieser Art der Darstellung von GA spricht. Ein weiterer Unterschied besteht in der Art des Population Managements und dem hinzufügen neuer Generationen, dabei können entweder mehrere Generationen co-existieren oder die alte Generation wird durch die neue ersetzt. Darüber hinaus wird unterschieden zwischen verschiedenen Reproduktionsverfahren bspw. durch die Implementierung von verschiedenen Geschlechtern. Weitere wichtige erweiternde Konzepte abseits der Reproduktion ist die Interaktion der einzelnen Individuen zueinander und der Einfluss dessen auf die Fitnessfunktion. Bei einer komplexen Implementierung simulierter Interaktion spricht man auch häufig von künstlichem Leben (artificial life) oder co-evolution[10] [7] Da die Grenzen teilweise

fließend sind, ergeben sich oft Überschneidungen zwischen verschiedenen Disziplinen der EA. Die Klasse der genetischen Algorithmen wird dabei oft synonym zu EA verwendet. Aufgrund der angesprochenen Überschneidungen und dem Fakt geschuldet dass die genaue Repräsentation der genetischen Merkmale oftmals nicht aus dem Papern hervorgeht, wird auf eine explizite Unterscheidung in der Analyse der Experimente verzichtet.

2.4 Anwendungsgebiete

Die Einsatzgebiete EA sind vielfältig und nahezu unbegrenzt. Da die Rolle der Optimierung von Algorithmen einen großen Bereich in der Informatik einnimmt, ist der Bereich der Optimierung von NP-Problemen [16], NP steht dabei für nicht polynomiell und beschreibt in der theoretischen Informatik die Probleme zu denen bisher kein effizienter Algorithmus (Polyzeit) zur Berechnung bekannt ist, zu den wichtigsten Forschungsgebieten.[20] Ein weiteren wichtigen Bereich stellen außerdem die Simulation und Optimierung in Design-Prozessen dar. Dieser Bereich wird häufig unter dem Begriff des evolutionären Design (ED) geführt und behandelt unterschiedliche Forschungsgebiete. Vorrangig wird sich dabei mit der Optimierung und Evaluation von technischen Systemen in einer simulierten Umgebung beschäftigt [18]. Im Kapitel über Unerwartete Lösungsansätze werden drei Experimente aus den Bereichen Robotik, Simulation sowie Produktdesign vorgestellt und deren Implementierungen sowie Lösungsansätze der EA diskutiert.

3 Unerwartete Lösungsansätze

3.1 Antennendesign Mission ST-5

Eines der bekanntesten Einsätze EA ist das 2006 von der University of California Santa Cruz in Zusammenarbeit mit dem NASA Ames Research Center durchgeführte Experiment zur Entwicklung und Optimierung von Antennendesigns für die unter dem Programm ST-5 gestarteten Testsatelliten. Der implementierte EA startet mit einer initialen Population an zufälligen Antennendesigns, welche mithilfe einer Elektromagnetischen Simulation auf ihre Leistungsfähigkeit getestet wurden. Die einzelnen Lösungskandidaten der Simulation konnten aus 4 Antennenarmen bestehen, welche alle 0.4cm die Antenne jeweils um 5 Grad in eine beliebige Richtung im Raum verschieben konnte. Im Laufe des Projektes wurden die 4 Antennenarme auf einen einzigen reduziert, um weitere Parameter der Mission zu erfüllen. So durfte ein Antennendesign nur aus einem einzigen Arm bestehen, um der Rotation der Satelliten im Orbit Stand zu halten. Innerhalb von 5 Monaten konnte sowohl der EA implementiert als auch die Herstellung der Antennen fertig gestellt werden. Dabei sind die Ergebnisse des EA der Lösung eines menschlichen Designs in Sachen Robustheit und Effizienz deutlich überlegen. Die ausgewählten Lösungskandidaten erreichen eine komplexe Verzweigung des Antennenarms, die unmöglich in gleicher Zeit durch herkömmliche Designverfahren erreichbar gewesen wären. [11]

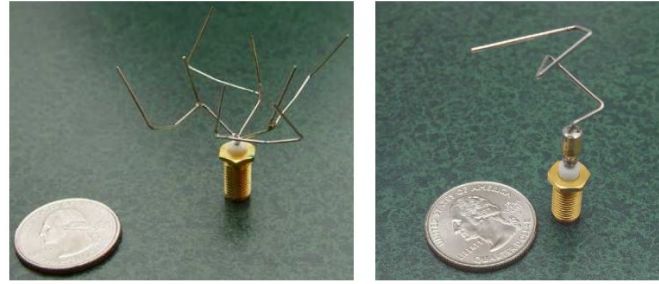


Abbildung 2: Lösungskandidaten ST-5 Experiment [11]

3.2 Alternative Fortbewegung von Robotern bei Beschädigung

Ein weiteres interessantes Experiment im Forschungsgebiet der EA wird im Nature erschienen Artikel, “Robots that can adapt like animals”, beschrieben. Ziel war dabei die Anpassung eines 6-beinigen Roboters auf mögliche Schäden (zB. Ausfall eines Beines). Der Roboter sollte nachdem der Schaden festgestellt wurde selbstständig alternative Fortbewegungsmaßnahmen ergreifen. Das Vorgehen wurde dabei durch einen EA und der Roboter-Animations und Simulationssoftware Dart [9] durchgeführt. Die Lösungskandidaten wurden mithilfe eines MAP-Elite Algorithmus bestimmt - der Abseits der optimalen Lösung weitere lokale Maxima im Suchraum bestimmt [15]. Dadurch hatten die Forscher die Möglichkeit eine Vielzahl an unterschiedlichen Herangehensweisen zur Problemlösung durch den Roboter zu testen. Es stellte sich heraus, dass nicht unbedingt der Lösungsansatz mit dem höchsten Fitnesswert (optimierteste Lösung) zum gewünschten Erfolg führte. Einschränkungen in der Evolution der Fortbewegung eines Roboters wurde vorgenommen in dem der EA nur eine bestimmte Anzahl an Beine zu Fortbewegung verwenden durfte, um verschiedene Szenarien defekter Beinglieder zu simulieren. Die Resultate wurden ergänzend zum Artikel auch in Videoform veröffentlicht [4]. Eine besonders kreative Form der Fortbewegung konnte der Roboter unter der Beschränkung, dass keine Beine verwendet werden dürfen, entwickeln (siehe Abbildung) [14] [3].



Abbildung 3: Lösungskandidat: Beine dürfen Boden nicht berühren[3]

3.3 Simulation von virtuellen Kreaturen

Um das letzte hier vorgestellte Projekt näher zu beleuchten. Kommt man bei dem Titel Simulation von virtuellen Kreaturen nicht umhin die Arbeiten von Karls Sims zu erwähnen, welcher bereits 1994 virtuelle Kreaturen in verschiedenen Umgebung, an Land und im Wasser durch Einsatz eines EA das laufen, springen und schwimmen beibrachte. Dabei bestand ein Individuum aus Blöcken (Bones) und Verbindungen (Joints) die über ein kleines neuronales Netz angesteuert wurden die eine Bewegung des Blockwesens ermöglichten. Die Fitnessfunktion wurde dabei je nach gewünschten Verhalten (laufen, schwimmen, springen) immer wieder angepasst, wobei die Konfiguration des Neuronalen Netzes in Kombination mit den physischen Eigenschaften der Kreatur entscheidend war[19].

Auf exakt diesem Konzept basiert auch die Arbeit von Peter Krvcah der 2008 ein ähnliches Experiment in der Arbeit Towards Efficient Evolutionary Design of Autonomous Robots durchführte. Dabei beschäftigte sich Krvcah auch mit der Evolution des springens seiner virtuellen Blockgeschöpfe. Als Fitnessfunktion definierte er zunächst, dass ein Kreatur eine größere Chance der Reproduktion hat sofern ein Block in ihrem Körper möglichst weit auf der y-Achse vom Erdboden entfernt ist. Was sich zunächst nach einer plausible Fitnessfunktion anhört wurde von dem entwickelten EA jedoch schnell ausgetricks. Als Ergebniss erhielt Krvcah schlicht sehr, sehr große Kreaturen. Danach passte er die Fitnessfunktion erneut an sodass der Block der den Boden berührt möglichst weit auf der yAchse vom Erdboden entfernt sein musste. Doch auch nach der Anpassung beharte der EA darauf große Kreaturen zu schaffen, welche statt aktiv zu springen kreativ umfellen. [13]

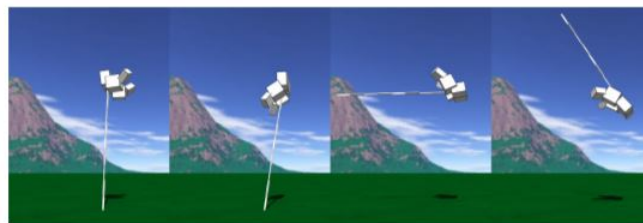


Abbildung 4: Lösungskandidat: Umfallen statt springen [13]

Auch hier zeigt sich dass EA nicht unbedingt immer das gewünschte Ergebnis liefern. Trotzdem sind Erkenntnisse dieser Art wichtig um die Simulationsumgebung zu verbessern und mögliche weitere Schlupflöcher zu schließen. Auch sind Experimente im Bereich Simulation von virtuellen Kreaturen essentiell wichtig für die Erforschung der Evolution auf der Erde [10].

4 Evaluation und versteckte Potenziale

4.1 Zusammenfassung und Fazit

In den dargestellten Versuchen ist zu erkennen, dass viele der in den Experimenten eingesetzten EA Lösungen gefunden haben, welche über den Erwartungen der jeweiligen Forschungsgruppen hinaus ging. Dabei hängt die vom EA gefundenen Lösungen stark von der Implementierungsart als auch von den gesetzten Rahmenbedingungen, in denen sich das Objekt evolutionär weiterentwickeln kann ab. Dabei können kleine Änderung der Parameter zu einer unerwartet kreativen Lösung führen. Eine Zusammenstellung weiterer Experimente findet sich in dem Paper “The Surprising Creativity of Digital Evolution” [14]. Dabei profitieren EA vor allem von Fortschritten im Bereich der Physikalischen Simulation. Je besser die Eigenschaften eines realen Objektes wie zB. Antennen oder Roboter modelliert werden können, desto besser performen die Lösungskandidaten auch wenn diese Tatsächlich gebaut und getestet werden.

4.2 Nachteile

Der Aufwand für die Modellierung eines Problems in einen EA ist hoch und das Aufstellen einer geeigneten Fitnessfunktion stellt Forscher vor eine schwierigen Herausforderung. Ein Beispiel ist die Anwendung der EA im Forschungsgebiet Artificial Life. Dort ist das Ziel eine möglichst realistische Abbildung der Welt, jedoch stößt man dabei schnell an die Grenzen des Machbaren. Schon die Berechnung der Physikalischen Gesetze und die Auswirkung auf die einzelnen Individuen ist so berechnungsintensiv das diese derzeit kaum realistisch abbildbar ist.[11] Ein weiterer offensichtlicher Nachteil ist die Tatsache dass es schwierig ist Festzustellen ob es sich bei der vom EA gefunden Lösung tatsächlich um die optimalste Lösung im Suchraum handelt, wobei dieser Nachteil auch auf alle anderen Optimierungsverfahren zutrifft und eher ein generelles Problem in der Lösungsfindung darstellt. Die generalisierte Struktur macht es einerseits einfach EA auf ein Problem anzuwenden dabei können die entwickelten Algorithmen aber kaum in Betrachtung der Effizienz mit speziell angepasste, mathematische Komplexe Algorithmen mithalten [10] [14].

4.3 Vorteile

EA sind im Gegensatz zu vielen anderen Lösungsverfahren wie zB. Neuronale Netzwerke keine Blackbox und auch abseits der Annäherung an die optimale Lösung findet sich ggf. eine interessanter Ansatz der im Experiment noch nicht berücksichtigt wurde.[15] Darüber hinaus liefern EA sowie auch andere Optimierungsalgorithmen wie zB. der Greedy-Algorithmus zu jedem Zeitpunkt eine Lösung. Die generalisierte Struktur hilft dabei den Algorithmus in vielen Bereichen einzusetzen. Die Möglichkeiten bei der Unterstützung von Produktdesigns sind vielfältig und sparen langfristig Kosten und Zeit in der Entwicklung.[11] Ein weiterer klarer Vor-

teil ist die Fähigkeit von EA Fehler in einer Simulationsumgebung zu finden, wobei die Anwendung dieser Umgebungen auf viele Forschungsbereiche Einfluss hat. Der jedoch wohl größte Vorteil ist das Systeme entwickelt werden können welche sich stetig weiter entwickeln und auch abseits der Vorgaben kreative Lösungsansätze finden können.

4.4 Abwägung des Einsatzes

Der Einsatz von EA hängt stark von dem Fokus und der präferierten Lösung ab. Wenn das Ziel eine einfache Optimierung sein soll, eignen sich Greedy-Algorithmen mit lokaler Suche in der Regel eher, da diese schneller zu programmieren sind und der Modellierungsaufwand deutlich geringer ist. [17] Bei Experimenten, in denen noch keine konkrete Lösung präferiert wird und nicht unbedingt nur die optimale Lösung als Ergebnis von Interesse ist, eignen sich EA besonders. Da sich EA mit unterschiedlichen Konfigurationen immer versuchen dem lokalen Maximum des Suchraums also einer optimalen Lösung annähern, dabei kann man aber durch Zurückverfolgen der Generationen immer auch die Zwischenschritte im Suchraum nachverfolgen. Ein Beispiel für die Nutzung dieser Eigenschaft ist der in Alternative Fortbewegung von Robotern bei Beschädigung erwähnte MAP-Elite Algorithmus [15] [6]. Abschließend kann man sagen dass der Einsatzbereich EA sehr vielseitig ist, aber die Büchse der Pandora ist es auch nicht. Eine Anwendung sollte demnach stets gegenüber den Alternativen abgewogen werden, da das Konzept ansich zwar einfach ist aber die Umsetzung am Ende trotzdem beliebig komplex werden kann.

5 Ausblick

Der Forschungsbereich der EA wächst stetig weiter, wobei die Grenzen zu anderen Disziplinen der Erforschung der künstlichen Intelligenz immer weiter verschwimmen. Die Verfügbarkeit von Rechenleistung, auf die immer einfach zugegriffen werden kann (Cloud-Computing), unterstützt die Entwicklung EA und verbreitet die Technologie weiter. In einem vom Unternehmen Sentient durchgeführten Interview mit führenden Vertretern des Forschungsgebiet der künstlichen Intelligenz, sehen viele großes Potenzial in der Anwendung von EA zur Entwicklung von kreativen, selbst lernenden Systemen. [2]

Literatur

- [1] Kategorien. [Online; accessed 03.12.2018], [https://en.wikipedia.org/wiki/Category:Optimization_algorithms_and_methods].
- [2] Sentient website. [Online; accessed 09.02.2019], [<https://www.prnewswire.com/news-releases/evolutionary-algorithms-the-next-big-thing-in-machine-learning-300696623.html>].
- [3] Danesh Tarapore Jean-Baptiste Mouret Antoine Cully, Jeff Clune. Robots that can adapt like animal. 2015. [Online; accessed 10.12.2018].
- [4] Danesh Tarapore Jean-Baptiste Mouret Antoine Cully, Jeff Clune. Robots that can adapt like animals, additional sources. <https://www.youtube.com/watch?v=T-c17RKh3uE&feature=youtu.be>, https://github.com/resibots/ite_v2, 2015.
- [5] Peter J. Bowler. *Evolution: The History of an Idea*. 2003.
- [6] Diego Browarnik and Pedro Gonnet. Kurs: Genetische Algorithmen. https://de.wikiversity.org/wiki/Kurs:Genetische_Algorithmen/Kapitel_7, 2007. [Online; accessed 03.12.2018].
- [7] Lars Bungum and Björn Gambäck. Evolutionary Algorithms in Natural Language Processing. 2010. [Online; accessed 29.10.2018].
- [8] Charles Darwin and Alfred Russel Wallace. *the origin of species*. 1859. [<https://bit.ly/2Eq76e3> | Auflage 6].
- [9] Lee et al. Dart: Dynamic animation and robotics toolkit. journal of open source software, 3(22), 500,. <https://doi.org/10.21105/joss.00500>, 2018.
- [10] D.B Fogel and Thomas Baeck. *Handbook of Evolutionary Computation*. 1997.
- [11] Gregory Hornby, Al Globus, Derek Linden, and Jason Lohn. Automated antenna design with evolutionary algorithms. 2006. [Online; accessed 03.12.2018].
- [12] De Jong Kenneth. *Evolutionary Computation: a unified approach*. MIT Press, 2002.
- [13] Peter Krvcah. Towards efficient evolutionary design of autonomous robots. In *International Conference on Evolvable Systems*, pages 153–164. Springer, 2008.
- [14] Joel Lehman and more. The Surprising Creativity of Digital Evolution. 2018. [Online; accessed 29.10.2018].

- [15] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *CoRR*, abs/1504.04909, 2015.
- [16] Stephan Olariu. *Handbook of bioinspired algorithms and applications*. Chapman and Hall, 2006.
- [17] David Poole and Alan Mackworth. *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press, Cambridge, UK, 2 edition, 2017.
- [18] Ingo Rechenberg and Manfred Eigen. *Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, 1973.
- [19] Karl Sims. Evolving virtual creatures. In *Evolving virtual creatures*. ACM, 1994.
- [20] Michael Sipser. *Introduction to the Theory of Computation*. International Thomson Publishing, 1st edition, 1996.