Fachbereich Informatik

Studienarbeit

im Studiengang Informatik Master of Science

Benutzerschnittstellen für die interaktive Evolution

Betreuer: Dipl.-Inform. Ingo Boersch

28. Februar 2011

Stephan Dreyer

Matr.-Nr. 20062042

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe. Es wurden keine außer den im Literaturverzeichnis angegebenen Quellen und Hilfsmitteln verwendet. Die Arbeit hat in dieser oder ähnlicher Form noch keinem anderen Prüfungsausschuss vorgelegen.

Brandenburg, den 28. Februar 2011

Stephan Dreyer

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Motivation 1
1.2 Aufgabenstellung 2

2 Grundlagen **3**
2.1 Evolutionäre Algorithmen 3
2.2 Genetische Algorithmen 4
2.3 Die Fitnessfunktion 5
2.4 Interaktive Selektion 5
2.5 Arten der interaktiven Fitnesszuweisung 10
2.6 Anfassbare Benutzerschnittstellen 13

3 Methoden zur Verringerung der Benutzerermüdung **14**
3.1 Einführung 14
3.2 Methoden aus der Literatur 14
3.3 Vorschläge für neue Methoden 20
3.4 Taxonomie der Methoden 21

4 Benutzerschnittstellen **23**
4.1 Einführung 23
4.2 Klassische Modelle 23
4.3 Innovative Konzepte 25
4.4 Taxonomie der Modelle 29
4.5 Die Beispielanwendung 31

5 Fazit und Ausblick **32**

Literatur **33**

Abbildungsverzeichnis **35**

Tabellenverzeichnis **35**

Glossar **36**

A Bilder der Beispielanwendung **38**

1 Einleitung

1.1 Motivation

Der Bereich der künstlichen Intelligenz ist ein „Teilgebiet der Informatik, welches versucht, menschliche Vorgehensweisen der Problemlösung auf Computern nachzubilden, um auf diesem Wege neue oder effizientere Aufgabenlösungen zu erreichen.“[LC01] Ein Blick auf die biologische Evolution inspirierte zur Entwicklung von neuen Lösungen für altbekannte Probleme. Die Natur ist ein Meister der Optimierung, wodurch sich aus einzelligen Organismen komplexe Lebensformen entwickeln konnten, die optimal an ihren Lebensraum angepasst sind. Viele Tiere besitzen erstaunliche Eigenschaften, um beispielsweise an Futter zu gelangen oder selbst nicht zu einer leichten Beute zu werden.

Evolutionäre Algorithmen stellen ein schematisches, abstraktes Modell der biologischen Evolution dar. (vgl. [GKK04] S.33) Sie besitzen keine fest programmierten Problemlösungen, sondern streben danach, diese selbst zu finden.

Evolutionäre Algorithmen funktionieren wie folgt:

- Eine mögliche Problemlösung wird auf jeweils ein *Individuum* abgebildet.
- Es gibt eine ganze *Population* zufällig erzeugter Individuen.
- Die Individuen haben einen so genannten *Genotyp*.
- Individuen können ihren Genotyp vererben.
- Aus dem Genotyp der Individuen wird ein *Phänotyp* erzeugt.
- Die Güte der Lösung wird als *Fitness* bezeichnet. (bei Lebewesen ist eine möglichst hohe Überlebenswahrscheinlichkeit das Ziel)
- Die Fitness beeinflusst die Wahrscheinlichkeit der Reproduktion eines Individuums.

Bei der Fitness wird zwischen zwei Arten unterschieden. Die explizite Fitness ist ein numerischer Wert, während die implizite Fitness sich meist durch ihren Rang in der Population ergibt. Die Fitness eines Individuums kann automatisch durch eine *Fitnessfunktion* oder durch eine subjektive Bewertung durch den Benutzer bestimmt werden. Letzteres wird als *interaktive Fitness* bezeichnet. Bei einem evolutionären Algorithmus mit interaktiver Fitness übernimmt der Benutzer die Rolle des Steuermanns durch die Evolution. EA mit interaktiver Fitness sind notwendig, wenn es für eine Anwendung keine Fitnessfunktion gibt, die maschinell berechnet werden kann. Vorlieben, Intuition, Emotionen und psychologische Aspekte bestimmen die Richtung der Evolution. (vgl. [Tak01] S.1275) Der Mensch ersetzt hier die Fitnessfunktion und wird aus algorithmischer Sicht als Black Box angesehen. Ohne den Benutzer findet keine Fitnessbewertung und somit keine Evolution statt. Menschen sind anders als Maschinen nur begrenzt belastbar, wenn sie monotone Aufgaben ausführen müssen. Aus diesem Grund müssen evolutionäre Algorithmen mit interaktiver Fitness so entworfen werden, dass der Benutzer nicht zu schnell ermüdet. [Tak01]

In den letzten Jahren haben sich *Tangible User Interfaces* in ihrer Praxistauglichkeit stark weiterentwickelt. Smartphones, E-Book-Reader und Multi-Touch-Computer nutzen diese Eingabemethode bereits und erfreuen sich aufgrund der einfachen Bedienung einer großen Beliebtheit. Multi-Touch-

Systeme erweitern die Eingabemethoden herkömmlicher Touchscreens um Gesten zum Drehen und Skalieren von Objekten sowie der Möglichkeit, mit mehreren Benutzern gleichzeitig an einem Gerät arbeiten zu können. Außerdem kann der Benutzer durch Gesten wie Zeigen, Anfassen, Verschieben und Drehen mit einem Computerprogramm ähnlich interagieren, wie er es mit einem physischen Gegenstand tun würde. Dies ist auch für unerfahrene Benutzer eine Erleichterung bei der Bedienung von Programmen, die Lernkurve ist flach. Sollten sich TUI als eine modernere, bessere Form von Eingabegeräten erweisen, könnten diese auch für die interaktive Evolution Vorteile bieten.

1.2 Aufgabenstellung

Das Ziel dieser Arbeit ist, eine Übersicht über Anwendungsfälle, Probleme und Konzepte zur Umsetzung von interaktiven evolutionären Algorithmen zu erstellen. Dabei sollen reale und mögliche Anwendungsszenarien systematisiert werden. Es soll durch eine Literaturrecherche untersucht werden, welche Methoden existieren, um dem Problem der Benutzerermüdung entgegen zu wirken. Die Methoden sollen in Form einer Taxonomie nach ihren spezifischen Eigenschaften klassifiziert werden. Weiterhin soll untersucht werden, welche Modelle für Benutzerschnittstellen zur interaktiven Fitnessbestimmung existieren. Diese sollen nach ihren Eigenschaften und ihrer Eignung für die parallele Nutzung durch mehrere Benutzer aufgelistet werden. Dazu sollen die speziellen Vorteile und Probleme von Multi-Touch-Systemen untersucht und berücksichtigt werden. Anschließend soll prototypisch eine Software zur Demonstration der Vorteile von Multi-Touch-Systemen entwickelt werden.

2 Grundlagen

2.1 Evolutionäre Algorithmen

Ein in der Natur bei allen Lebewesen anzutreffendes Prinzip zur Optimierung von Parametern ist die Evolution. Sie optimiert verschiedene Parameter zur Erreichung einer möglichst hohen Überlebensfähigkeit und zur Erzeugung von Nachkommen. Dies wird in diesem Zusammenhang als Fitness bezeichnet. Lebewesen, die sehr gut an ihren Lebensraum angepasst sind und sich so gegenüber ihren Artgenossen durchsetzen können, erreichen eine hohe Fitness. Charles Darwin beschrieb dies in [Dar09] mit dem Ausdruck „Survival of the Fittest“. Daraus folgt, dass diese Individuen ebenfalls eine hohe Chance haben, sich zu reproduzieren. Im Gegensatz dazu ist es bei Lebewesen mit einer geringen Fitness unwahrscheinlicher, dass sie sich vermehren und sich ihr Genom über Generationen hinweg durchsetzen kann. Nach Charles Darwin[Dar09] sind vier Bedingungen für das Auftreten von Evolution erforderlich:

- Vermehrung von Individuen einer Population
- Veränderungen, die die Vermehrungsrate beeinflussen
- Vererbbarkeit der Veränderungen
- Begrenzte Ressourcen

Die Aufgabe evolutionärer Algorithmen besteht nun darin, diese Bedingungen auf technische Systeme zu übertragen. Im Laufe der zweiten Hälfte des zwanzigsten Jahrhunderts haben sich vier Unterarten der evolutionären Algorithmen durchgesetzt:

- Evolutionäre Programmierung (EP)
- Evolutionsstrategien (ES)
- *Genetische Algorithmen* (GA)
- *Genetisches Programmieren* (GP)

Alle vier Verfahren funktionieren nach demselben Schema: Lösungen für technische Probleme werden auf einzelne Individuen abgebildet. Ein Individuum kann beispielsweise ein Programm, ein Vektor von Parametern oder gar eine Baumstruktur sein. Für jedes Individuum muss es möglich sein, einen Fitnesswert zu ermitteln. Dieser kann berechenbar, durch eine subjektive Beurteilung oder implizit durch eine Überlebenswahrscheinlichkeit zu bestimmen sein. Außerdem müssen auf das Individuum genetische Operatoren (z.B. Mutation oder Crossover) anwendbar sein.

Der Ablauf genetischer Algorithmen lässt sich in zwei verschiedene Verfahren gliedern. Die *generationenbasierte Arbeitsweise* funktioniert so, dass jeweils die gesamte alte Generation durch die neuere ersetzt wird. Im Gegensatz dazu werden bei der *Steady-State-Arbeitsweise* nur einzelne Individuen der Population durch neue ersetzt. Dies erfordert eine Ersetzungsselektion. Ein Parameter dafür kann das Alter eines Individuums sein. In den meisten evolutionären Algorithmen werden die ältesten Individuen zuerst ersetzt. Es kann jedoch auch von Vorteil sein, das Individuum mit der höchsten Fitness unabhängig vom Alter vor einer Ersetzung zu schützen. [BHS07]

2.2 Genetische Algorithmen

Genetische Algorithmen wurden 1975 vom US-amerikanischen Informatiker John Holland entwickelt. Diese benötigen als Grundlage eine geeignete Codierung des zu optimierenden Sachverhaltes, oder zumindest der Parameter, die die Problemlösung charakterisieren. Diese einzelnen Parameter bilden den Genotyp des Individuums. Die zu optimierenden Größen werden häufig als ein einziger Bitstring dargestellt. [LC01]

Ein genetischer Algorithmus läuft wie folgt ab: Zunächst wird eine Population von Individuen zufällig erzeugt. Für jedes Individuum wird anschließend die Fitness bestimmt. Ausgehend von der Fitness werden zwei Individuen ausgewählt, daher eine *Selektion* getroffen. Zwischen diesen Individuen wird mit einer festgelegten Wahrscheinlichkeit P_c eine Kreuzung (Crossover) durchgeführt und mit einer ebenfalls definierten Wahrscheinlichkeit P_m eine Mutation durchgeführt. Eine Mutation bedeutet eine zufällige Abweichung von den Elternindividuen. Diese kann auf der Bitebene oder auf der Parameterebene stattfinden. Ein Crossover sieht meist so aus, dass zwei Elternteile an einem zufälligen Index i zerschnitten und die Endstücke getauscht werden (1-Punkt-Crossover). Beim 2-Punkt-Crossover werden beide Individuen an zwei Stellen zerschnitten und nur die Mittelstücke getauscht.[BHS07] Als Referenz wird oft der SGA[1] herangezogen. Dieser ist in Abbildung 1 dargestellt.

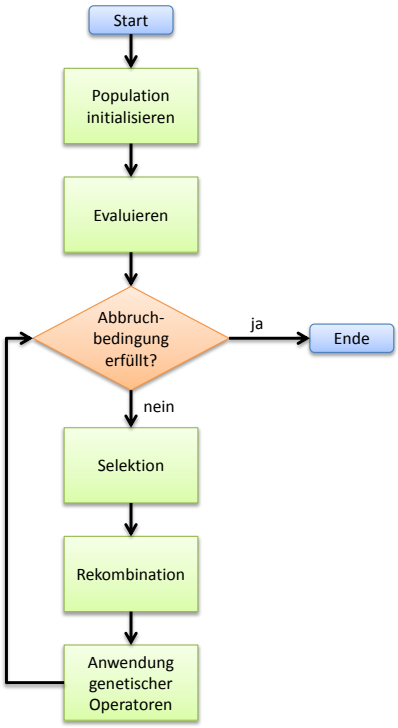


Abbildung 1: Einfacher genetischer Algorithmus

[1] SGA ist ein Akronym für Simple Genetic Algorithm, auf Deutsch: einfacher genetischer Algorithmus

2.3 Die Fitnessfunktion

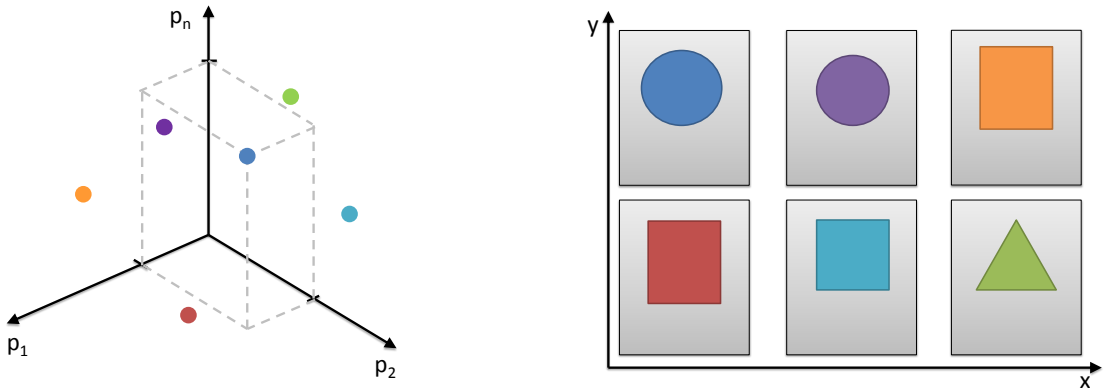
Die Fitness eines Individuums kann explizit, zum Beispiel als numerischer Wert, oder implizit durch einen Rang in der Population definiert werden. Bei genetischen Algorithmen ist es oft notwendig, einen expliziten Fitnesswert für die Individuen zu bestimmen. Wenn ein Fitnesswert existiert, ist es möglich, Individuen zu selektieren und so zu bestimmen, welche Individuen überleben und reproduziert werden dürfen. Die Fitnessfunktion ist daher eine Abbildung des Suchraums auf einen Fitnesswert. Dazu wird der Phänotyp des jeweiligen Individuums einer Bewertung unterzogen. Üblicherweise wird als Fitnesswert oft eine positive reelle Zahl verwendet, wobei S der Suchraum ist. Die Funktion 1 zeigt dies.

$$fit : S \rightarrow [0, \infty) \tag{1}$$

Wenn es Ziel des Algorithmus ist, das Maximum einer Funktion f zu finden, so ist das Chromosom c_1 genau dann eine bessere Lösung als Chromosom c_2, wenn gilt: $fit(c_1) > fit(c_2)$. Wird hingegen nach dem Minimum der Funktion f gesucht, so lässt sich dieser Fall auf die Maximierung zurückführen, wenn es eine Konstante $const \in \mathbb{R}$ gibt, für die gilt: $\forall c \in S : f(c) \leq const$. Die Fitnessfunktion wird nun anders definiert: $fit_2(c) = const - fit(c)$. fit_2 kann nur positive Werte annehmen soll durch den Algorithmus maximiert werden. [GKK04]

2.4 Interaktive Selektion

Ist es nicht möglich, eine Fitness für Individuen maschinell zu berechnen oder zu messen, so ist ein evolutionärer Algorithmus auf eine *Evaluation* durch den Menschen angewiesen. Der Mensch ist also Teil des Algorithmus, ohne ihn findet keine Bewertung der Güte von Lösungen und damit keine Optimierung statt. Formal ergeben sich durch die interaktive Fitnessbewertung zwei Funktionen. Eine Darstellungsfunktion $f_d : S_{para} \rightarrow S_{phys}$ ist eine Abbildung des Parameterraums S_{para} (siehe Abbildung 2(a)) auf einen physischen Raum S_{phys} (siehe Abbildung 2(b)), der vom Benutzer beispielsweise optisch wahrgenommen werden kann.

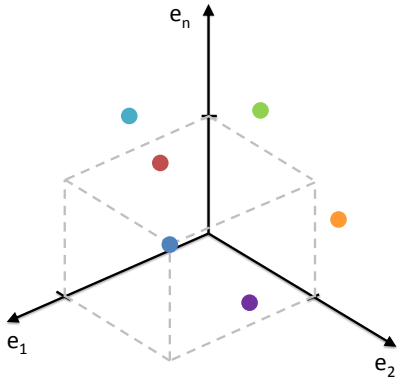


(a) Individuen im Parameterraum S_{para}

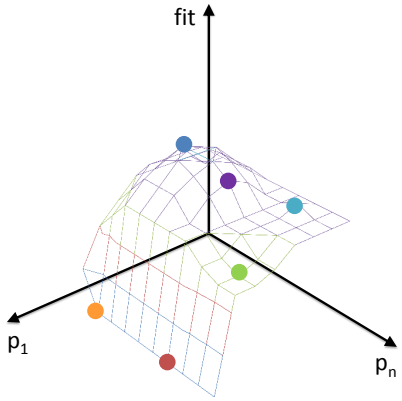
(b) Individuen im physischen Raum S_{phys}

Abbildung 2: Die Darstellungsfunktion f_d bildet Individuen vom Parameterraum in den physischen Raum ab

Eine subjektive Bewertungsfunktion $f_{sub} : S_{psych} \rightarrow S_{fit}$ ist eine Abbildung des psychischen Raums S_{psych} auf den Suchraum S_{such} mittels des Benutzers. Der psychische Raum S_{psych}, dargestellt in Abbildung 3(a), besteht aus subjektiven Eindrücken des Benutzers. Der Suchraum S_{such} ergibt sich aus dem Parameterraum S_{para}, erweitert um die Dimension der Fitness fit. Dieser ist in Abbildung 3(b) dargestellt. Das dort dargestellte Gitter, welches sich im Suchraum durch eine Verknüpfung der einzelnen Fitnesswerte ergibt, wird oft als Fitnesslandschaft bezeichnet. Die subjektive Fitness eines Individuums ist auch abhängig von den subjektiven Eindrücken der restlichen Individuen. Sie darf nicht als absoluter Wert betrachtet werden, denn oft ergeben erst in einer späteren Generation neue Ziele.



(a) Individuen im psychischen Raum S_{psych}



(b) Individuen im Suchraum S_{such}

Abbildung 3: Die subjektive Bewertungsfunktion f_{sub} bildet Individuen vom psychischen Raum in den Suchraum ab

In der Fachliteratur wird im Zusammenhang mit einer interaktiven Fitnessbewertung oft von einer eigenen Art evolutionärer Algorithmen gesprochen. In dieser Arbeit soll zwischen zwei Begriffen unterschieden werden:

- *Evolutionäre Algorithmen* (im Folgenden mit dem Akronym *EA* bezeichnet[2]) besitzen eine maschinell berechenbare Funktion zur Bestimmung der Fitness.
- *Interaktive evolutionäre Algorithmen* (im Folgenden mit dem Akronym *IEA* bezeichnet[3]) ersetzen die Fitnessfunktion durch eine menschliche Bewertung.

[2] In der englischsprachigen Literatur werden evolutionäre Algorithmen meist als „evolutionary computation“, kurz „EC“ bezeichnet

[3] In der englischsprachigen Literatur werden interaktive evolutionäre Algorithmen meist als „interactive evolutionary computation“, kurz „IEC“ bezeichnet

2.4.1 Anwendungsgebiete

IEA werden in vielen Gebieten der Wissenschaft, Kunst und Technik verwendet. Vor allem in Japan werden diese vielfältig eingesetzt. Das Miteinbeziehen von menschlichen Emotionen beim Produktdesign wird dort als Kansei Engineering bezeichnet [Nag99].

Beispiele für Anwendungen von IEA sind:

- Design
  - Modedesign

    Hier wurde ein IEA zur Assistenz beim Modedesign entwickelt. Durch Kombinationen von sechs verschiedenen Kleidungsteilen (z.B. Ärmel, Kragen, Rock und Gürtellinie) werden Kleidungsstücke zusammengestellt, die der Benutzer in jeder Generation bewerten muss. [LKC01]

  - Redaktionelles Design

    Im vorgestellten Beispiel wurde ein IEA verwendet, um ein Poster zu entwerfen. Die Parameter waren Layout, Schriftgröße und Farbe. Der Benutzer konnte aus zwölf Wörtern wählen, um seinen Eindruck zu beschreiben. Auf dieser Basis wurden die nächste Generation von Posterlayouts erzeugt. [Tak01] S.1280

  - Mikroelektromechanische Systeme

    Es wurde ein System entwickelt, welches Entwürfe für Inertialsensoren und RF-Filter mittels eines EA automatisch erzeugt. Der Benutzer kann hier sein Wissen und seine Erfahrung mit einbringen, indem er jede zehnte Generation manuell bewertet. [KZTA05]

  - Raumplanung in der Gebäudearchitektur

    Die Autoren entwickelten einen Raumplanungsalgorithmus, der aus Vorgaben (Anzahl der Räume) und Regeln (z.B. Reihenfolge in der die Räume wachsen, Form der Räume, Länge der Wände) Grundrisse von Räumen entwickelt. Der Benutzer bewertet diese anschließend. Anwendung findet dieses Beispiel bei der Planung von Räumen in Krankenhäusern. [IT09]

- Kunst
  - Graphische Kunst und Computeranimation

    Es wurde ein System zur Generierung von Fraktalen entworfen. Der Benutzer evaluiert jede Generation von Fraktalen und steuert so das Ergebnis. [CL01]

  - Beleuchtung in der Computergrafik

    Ein IEA steuert hier die Art, Beleuchtungsintensität und Position von Lichtquellen im dreidimensionalen Raum. Der Benutzer bewertet die daraus entstandenen Bilder. Besonders Anfänger konnten so deutlich schneller die gewünschte Beleuchtungssituation erzeugen. [HT00]

  - Generierung von Filtern für die Bildverarbeitung

    Es wurde ein System zur Generierung von Bildverarbeitungsfiltern mit Hilfe einer Filtersprache entworfen. Das Programm erzeugt durch genetische Programmierung

Filter, die auf Testbilder angewendet wurden. Der Benutzer kann die erzeugten Bilder dann bewerten. [RM07]

- Komposition von Musik

  Die Autoren entwickelten ein System zum evolutionären Komponieren von Musik. Dabei wurde genetisches Programmieren in Verbindung mit genetischen Algorithmen verwendet. Die Individuen des GA sind kurze rhythmische Muster. Die GP-Individuen stellen die zeitliche Anordnung der GA-Individuen dar. Die Benutzer hören sich jedes erzeugte Musikstück an und erhöhen oder verringern den jeweiligen Fitnesswert. [TI00]

- Technik
  - Generierung von Gesichtsbildern

    Ein IEC stellt Gesichtsbilder anhand von sechs verschiedenen Merkmalen zusammen. Diese sind Haare, Augenbrauen, Augen, Nase, Mund, Gesichtsform. Der Benutzer kann die erzeugten Bilder bewerten und so die Evolution beeinflussen. So können beispielsweise Phantombilder erzeugt werden. [Tak00]

  - Sprachverarbeitung und Sprachsynthese

    Es wurde prototypisch eine Software zur Verbesserung von Sprachaufnahmen implementiert. Dabei werden Filter verwendet, die sechs Parameter zur Optimierung anbieten. Dies sind die Verstärkungsfaktoren bei 125Hz, 250Hz, 500Hz, 1kHz, 2kHz und 4kHz. In jeder Generation werden dem Benutzer 20 Individuen zur Evaluation präsentiert. Dies sind die gefilterten Sprachaufnahmen, die der Benutzer sich anhören und bewerten muss. So wird die Qualität der Filter in jeder Generation verbessert. [HT00]

  - Anpassung der Parameter von Hörgeräten

    Digitale Hörgeräte müssen an den jeweiligen Benutzer angepasst werden, um ein optimales Hören zu ermöglichen. Dabei kann die Person, die das Hörgerät justiert, nicht einschätzen, wie gut der Benutzer damit hört. Bei der Anpassung durch einen IEA werden dem Benutzer Einstellungen präsentiert und dieser Bewertet jeweils die Güte der Einstellung. So können gute Parameter einfach gefunden werden. [TO00]

  - Abfrage von Mediendatenbanken

    Es wurde ein IEA zur Suche von Bildern in Datenbanken entwickelt. Der Benutzer gibt dabei eine textuelle Beschreibung des Bildes ein, das System gibt darauf hin Bilder aus, die der Benutzer auf ihre Eignung für die Suchanfrage hin bewertet. Der Algorithmus lernt so selbstständig, welche Bilder zu welchen Begriffen passen. [Tak01] S.1283

  - Bildverarbeitung in der Medizin

    Ein System zur Verbesserung von Bildern bei der Magnetresonanztomographie (MRT) wurde entworfen. Der IEA evolviert optische Filter, die Tumore erkennen und auf dem Bild markieren sollen. Der Benutzer kann die Tumore mit einem Zeichentablett nachzeichnen. Aus der Anzahl der Pixel, die vom Filter falsch markiert wurden, wird die Fitness bestimmt. So können die Filter optimiert werden. [Tak01] S.1284

- Steuerung und Robotik

  Ein IEA wurde eingesetzt, um die Bewegungen eines Roboterarms möglichst natürlich wirken zu lassen. Mit genetischer Programmierung werden Bewegungsprogramme erstellt, die der Benutzer anschließend auf ihre Natürlichkeit bewertet. [Tak01] S.1284

- Lebensmittelindustrie

  In Kooperation mit einem Kaffeeproduzenten in Berlin wurde ein IEC zur Mischung verschiedener Kaffeesorten entwickelt. Fünf Arten von Kaffeebohnen wurden gemischt, wobei das Mischungsverhältnis durch den evolutionären Algorithmus bestimmt wurde. Anschließend wurde von Kaffeetestern bewertet, wie nah der Geschmack an dem des Referenzkaffees lag. [Tak01] S.1285

- Wissenschaft
  - Geophysik

    Für Forschungen in der Geophysik wurde ein IEC angewendet. Die Wissenschaftler erstellten ein Modell zur Simulation der Mantelkonvektion der Erde. Die Startparameter für die Simulation konnten durch den IEC optimiert werden, um die gewünschten Ergebnisse zu erzielen. [Tak01] S.1285

  - Bildung

    Es wurde ein System für den Schreibunterricht entwickelt, mit dem Kinder lernen sollen, Geschichten zu verfassen. Das System zeigt 24 Bilder an und das Kind kann zweimal jeweils vier Bilder auswählen, aus dem eine Geschichte verfasst werden soll. Der evolutionäre Algorithmus erstellt mehrere Sätze, die sich aus vier Bildern zusammensetzen. Das Kind wählt davon in jeder Generation zwei bessere Sätze aus, die in die nächste Generation übernommen werden sollen. Dies wird so lange wiederholt, bis eine zufriedenstellende Geschichte geschrieben werden kann. [Tak01] S.1286

  - Soziale Systeme

    Ein System zur Optimierung von Studententeams wurde entwickelt. Die Teams (Chromosomen) von Studenten (einzelne Gene) wurden auf ihre Teamleistung (Fitness) hin in jedem Semester (Generation) bewertet. Im folgenden Semester wurden die Teams durch Tauschen von Studenten (Crossover) neu zusammen gestellt. [Tak01] S.1286

  - Data Mining

    Für den Bereich des Data Mining wurde genetisches Programmieren verwendet, um aus einer Menge von Parametern Funktionen zu generieren, die den jeweiligen Zielwert (im Data Mining „Klasse“ genannt) möglichst genau berechnen konnte. Dieser Algorithmus funktioniert auch ohne eine Bewertung durch den Menschen, jedoch wurde in jeder hundertsten Generation eine Evaluation durch den Benutzer durchgeführt, um die Konvergenz zu beschleunigen. [Tak01] S.1283

2.4.2 Anwendungsfälle

Wird für eine Optimierungsaufgabe ein evolutionärer Ansatz gewählt, so muss eine interaktive Fitnessbewertung erfolgen, wenn die Güte der Lösungen nicht eindeutig durch eine mathematische Funktion definiert werden kann. Existiert jedoch eine solche Funktion, so kann dennoch eine interaktive Fitnessbewertung eingesetzt werden, um die Konvergenz[4] des Algorithmus zu beschleunigen.

2.4.3 Probleme

IEA haben den Vorteil, dass der Benutzer seine Erfahrung, Vorlieben und die Fähigkeit, Zusammenhänge zu erkennen in den Optimierungsprozess mit einbringen kann. Hier liegt jedoch auch die größte Schwachstelle von IEA: der Benutzer muss am Prozess aktiv teilnehmen. Wird eine generationsbasierte Arbeitsweise gewählt, bei der alle Individuen einer Generation gleichzeitig grafisch dargestellt werden, so ist die Populationsgröße durch die Anzahl der gleichzeitig darstellbaren Individuen begrenzt. Bei zu vielen Individuen geht die Übersicht verloren und Details sind nicht mehr so gut zu erkennen. Wird die Population hingehen sequentiell dargestellt, oder sollen zeitdiskrete Medien wie Audio oder Video als Individuen dargestellt werden, so ist die Populationsgröße durch die Erinnerungskapazität des Menschen begrenzt. Die Anzahl der Generationen ist ebenfalls begrenzt, da der Benutzer bei einer monotonen Bewertungsaufgabe schnell ermüdet. Verschiedene Untersuchungen haben ergeben, dass diese Grenze bei etwa zehn bis 20 Generationen liegt. [Tak00] [Tak01] [HT00]

Die Ermüdung des Benutzers hängt nicht nur von der Anzahl der zu evaluierenden Individuen, sondern auch von der damit verbrachten Zeit ab. Dazu finden sich in der Literatur verschiedene Angaben. Gong, Yao und Yuan [GYY09] bezeichnen eine Zeit von unter zehn Minuten für die Evaluation als akzeptabel. Die Versuche in [LSA+06] haben ergeben, dass die Benutzer bei einer Evaluationsdauer von über einer Stunde erschöpft und frustriert waren.
Ein genereller Richtwert ist hier nicht zu finden und der jeweilige Grenzwert ist von der Art der Optimierung, der Anzahl an notwendigen Benutzerinteraktionen (z.B. Mausklicks), der Präsentationsform und der Benutzergruppe abhängig. Daher sollte dieser Wert im Einzelfall immer durch Versuche ermittelt werden.

Aufgrund des Problems der Benutzerermüdung ist der wichtigste Grundsatz bei der Entwicklung eines IEA, die Anzahl der zu bewertenden Generationen gering zu halten. Der Algorithmus muss daher so entworfen werden, dass er auch in wenigen Generationen eine gute Konvergenz zeigt. [Tak01]

2.5 Arten der interaktiven Fitnesszuweisung

Wie in Kapitel 2.3 bereits beschrieben wurde, benötigt jedes Individuum einen impliziten oder expliziten Fitnesswert, meist eine positive reelle Zahl. Die Selektionswahrscheinlichkeit hängt sehr stark von diesem Wert ab. Die Fitnesszuweisungen durch Benutzer lassen sich in verschiedene Arten unterteilen.

[4] Konvergenz bezeichnet in diesem Zusammenhang die Annäherung an das gesuchte Optimum [GKK04] S.56

2.5.1 Direkte Zuweisung

Eine Art der Fitnesszuweisung ist die direkte Zuweisung eines diskreten Fitnesswerts. Abbildung 4 zeigt an dem Beispiel aus [LKC01], wie die Benutzerschnittstelle dafür aussehen kann. Jedes Bild repräsentiert jeweils einen Phänotyp. Der Schieberegler unter den Bildern ermöglicht es, einen Fitnesswert zuzuweisen. In der Ausgangsstellung steht der Regler in der Mitte. Wird er dort belassen, drückt der Benutzer aus, dass er dem Phänotyp gegenüber neutral eingestellt ist. Wird der Regler nach links gezogen, so bedeutet dies, dass der Benutzer eine Abneigung gegenüber dem Phänotyp hat. Analog dazu bedeutet die Stellung rechts, dass er Gefallen daran findet. Der Schieberegler kann auf jede beliebige Stellung zwischen dem Minimum und dem Maximum verschoben werden und beeinflusst so direkt den Fitnesswert des Individuums.

Abbildung 4: Benutzerschnittstelle für eine direkte Zuweisung einer diskreten Fitness durch den Benutzer (aus [LKC01])

2.5.2 Gefallen und Abneigung

Bei dieser Art der Zuweisung muss der Benutzer sich nicht mit diskreten Fitnesswerten befassen und drückt lediglich binär seinen subjektiven Eindruck des Individuums aus: Gefallen (positiv) oder Abneigung (negativ). Der Rang des Individuums in der Population wird dadurch in eine positive oder negative Richtung verschoben [KZTA05] S.5668. Abbildung 5 zeigt ein Beispiel für eine solche Benutzerschnittstelle. Hier wählt der Benutzer bei jedem Individuum zwischen befriedigend (satisfactory) und unbefriedigend (unsatisfactory) aus.

Abbildung 5: Benutzerschnittstelle, durch die der Benutzer Gefallen oder Abneigung ausdrücken kann (aus [GYY09])

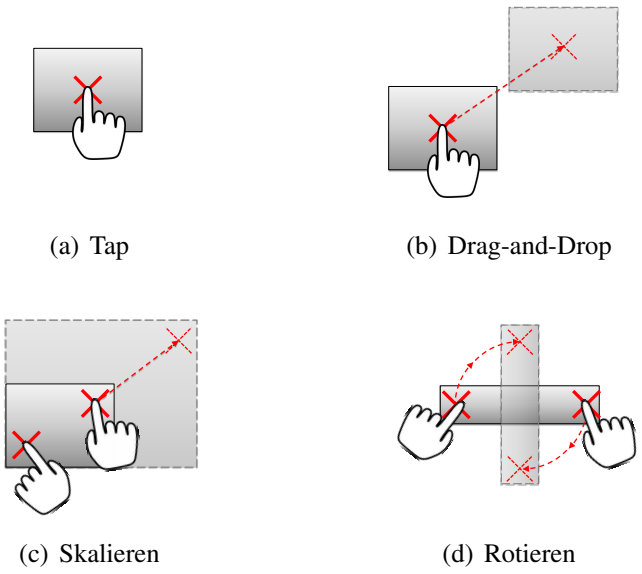
2.5.3 Mehrkriterielle Bewertung

Optimierungsaufgaben haben oft mehrere Kriterien, nach denen optimiert werden soll. Werden dem Benutzer Individuen präsentiert, von denen in seiner Gesamtheit kein Individuum besser ist, so kann er keine differenzierte Bewertung durchführen. Hier kann dennoch eine Bewertung erfolgen, wenn die Benutzerschnittstelle es dem Benutzer ermöglicht, nicht das Individuum als Ganzes, sondern nach einzelnen Kriterien zu bewerten. Diese Kriterien können einzelne Parameter repräsentieren oder auf die Parameter abbildbar sein. Der Benutzer wählt das Kriterium aus, welches am besten den Anforderungen genügt und dieses Kriterium wird dann weiter vererbt.

2.6 Anfassbare Benutzerschnittstellen

Benutzerschnittstellen, die durch Handberührungen oder Handbewegungen Interaktionen auslösen, werden als anfassbare Benutzerschnittstellen bezeichnet. Beispiele dafür sind die bei Notebooks üblichen Touchpads, Touchscreens oder Multi-Touch-Tables. Bei Touchscreens kann zwischen Single-Touch (ein Berührungspunkt) und Multi-Touch (mehrere Berührungspunkte) unterschieden werden. Multi-Touch-Displays haben den Vorteil, dass sie die Arbeit mit mehreren Benutzern parallel, sowie die Verwendung von Gesten ermöglichen. Folgende Gesten bilden die Grundlage zur Bedienung von Multi-Touch-Systemen [Sch10]:

- **Tap** ist ein einfaches Antippen eines Objekts mit dem Finger (siehe Abbildung 6(a)). Diese Geste wird meist zur Auswahl von Objekten verwendet.
- **Drag-and-Drop** verschiebt Objekte, indem der Benutzer diese antippt und den Finger in die gewünschte Richtung verschiebt, während er den Bildschirm berührt. (siehe Abbildung 6(b))
- **Skalieren** verkleinert oder vergrößert Objekte, indem zwei Finger auf den Bildschirm gelegt und aufeinander zu oder voneinander weg bewegt werden. (siehe Abbildung 6(c))
- **Rotieren** dreht Bilder um die eigene Achse, indem zwei Finger auf den Bildschirm gelegt und in einer Kreisbewegung um den Mittelpunkt bewegt werden. (siehe Abbildung 6(d))



(a) Tap (b) Drag-and-Drop

(c) Skalieren (d) Rotieren

Abbildung 6: Gesten zur Bedienung von Multi-Touch-Systemen (nach [Sch10])

Multi-Touch-Tische sind horizontal ausgerichtete Multi-Touch-Displays. Sie ermöglichen mehreren Benutzern das parallele Arbeiten, ähnlich einem Projektteam, das an einem Tisch sitzt und gemeinsam Entwürfe und Dokumente betrachtet und bearbeitet. An die Entwicklung von Anwendungen für Multi-Touch-Tische werden besondere Anforderungen gestellt. Bei der Verwendung von Text ist dabei zu beachten, dass dieser entweder aus mehreren Richtungen gelesen werden kann oder drehbar ist. Analog zu Dokumenten auf einem Tisch können Objekte auf einem Multi-Touch-Tisch oft zu Benutzern hingezogen und gedreht werden.

3 Methoden zur Verringerung der Benutzerermüdung

3.1 Einführung

Die Ermüdung des Benutzers ist das größte Problem von IEA, wie in Kapitel 2.4.3 beschrieben wurde. Wie dem entgegen gewirkt werden kann, ist Gegenstand zahlreicher Forschungen. Die meisten daraus entwickelten Methoden haben das Ziel, die Anzahl der zur Evaluation notwendigen Benutzerinteraktionen zu minimieren. In diesem Kapitel werden einige Lösungskonzepte zusammengefasst.

3.2 Methoden aus der Literatur

3.2.1 Einfachheit der Benutzerschnittstelle

Es gibt verschiedene Aussagen darüber, wie eine Benutzerschnittstelle gestaltet werden sollte, um effizient zu sein. Im Folgenden eine Übersicht der wesentlichen Konzepte:

- Die Anzahl der dem Benutzer zur Bewertung präsentierten Individuen sollte gering gehalten werden (vgl. [LC99]). Wird ein Individuum evaluiert, welches mehrfach in der Population enthalten ist, so kann jedem dieser Individuen die Fitness des evaluierten Individuums zugewiesen werden. Eine Mehrfachevaluation des gleichen Individuums in einer Generation sollte vermieden werden. Ebenso sollte es vermieden werden, dem Benutzer ungültige Lösungen zur Evaluation zu präsentieren. Dazu ist eine Validierungsfunktion mit entsprechenden Regeln notwendig.
- Die Benutzerschnittstelle sollte einfach zu bedienen sein, so dass für die Bewertung eines Individuums möglichst wenig Interaktionen vom Benutzer ausgeführt werden müssen. (vgl. [GYY09])
- Die visuelle Darstellung der Individuen sollte möglichst nah an der darzustellenden Information liegen. Dem Benutzer sollte jedoch nur der Phänotyp dargestellt werden, der Genotyp bleibt verborgen. (vgl. [LSA+06])
- Es sollten nur die Informationen dargestellt werden, die für die Bewertung der Individuen relevant sind. (vgl. [LSA+06])
- Die grafische Ausgabe sollte ansprechend, aber gemäß dem KISS-Prinzip[5] einfach und übersichtlich sein.
- Der Prozess der Evaluation sollte möglichst abwechslungsreich gestaltet werden und dem Benutzer Vergnügen bereiten. Bei einer monotonen Fließbandarbeit ermüdet der Benutzer sehr schnell.
- Das Ziel der Evolution sollte jederzeit für den Benutzer klar erkennbar sein. (vgl. [LSA+06]) Daher kann es hilfreich sein, während der gesamten Evaluation ein Elite-Individuum als Referenz anzuzeigen. [Tak01]
- Die Anzahl der möglichen Bewertungsstufen sollte gering gehalten werden. Findet eine direkte Zuweisung diskreter Fitnesswerte (wie in Kapitel 2.5.1) statt, so muss darauf geachtet

[5] Das KISS-Prinzip (Keep it simple and stupid) besagt, ein System so einfach wie möglich zu halten.

werden, dem Benutzer während der Evaluation nicht zu viele Werte anzubieten. Menschen sind beispielsweise nicht in der Lage, exakt zwischen 62 und 63 Punkten zu unterscheiden, wenn die Benutzerschnittstelle 100 Stufen zur Bewertung zur Verfügung stellt. Da während der Selektion aber $63 > 62$ gilt, obwohl der Benutzer vielleicht das gleiche meint, ist diese Einteilung suboptimal. [Tak01] beschreibt diesen Effekt als Quantisierungsrauschen.

[Tak01] zeigt auf, dass eine Einteilung in fünf bis sieben Stufen sinnvoll ist, um das Quantisierungsrauschen zu minimieren. Je weniger Stufen die Benutzerschnittstelle anbietet, umso weniger ermüdet der Benutzer. Allerdings nimmt die Effizienz des EA mit der Verringerung der Fitnessabstufungen ab. Ein weiteres Konzept ist der paarweise Vergleich. Dem Benutzer werden jeweils nur zwei Individuen gleichzeitig angezeigt. Der Benutzer wählt das Individuum aus, welches ihm besser gefällt. Der Rang des gewählten Individuums verbessert sich, während das andere Individuum im Rang absteigt.

3.2.2 Clusterbildung

In [LC99] wird eine Methode beschrieben, welche die Anzahl der zu evaluierenden Individuen verringert, während die Populationsgröße unverändert bleibt. Dies wird möglich, indem die Population in sogenannte Cluster geteilt wird. Ein Cluster ist hier eine Menge von Individuen, die vom Genotyp ähnlich sind. Aus jedem Cluster wird ein Individuum ausgewählt, welches den gesamten Cluster repräsentiert (siehe Abbildung 7). Dieses wird anschließend dem Benutzer zur Evaluation präsentiert. Der Fitnesswert der restlichen Individuen im Cluster wird anschließend in Abhängigkeit des Abstands zum repräsentativem Individuum zugewiesen. Der Abstand wird mit Gleichung 2 berechnet (euklidischer Abstand).

$$dist(x) = \sqrt{\sum_{i=1}^{n} (c_i - x_i)^2} \tag{2}$$

Als Clustering-Funktion wurde im Beispiel der k-Means-Algorithmus verwendet. Ein einfacher genetischer Algorithmus mit Clustering ist vereinfacht in Abbildung 8 dargestellt.

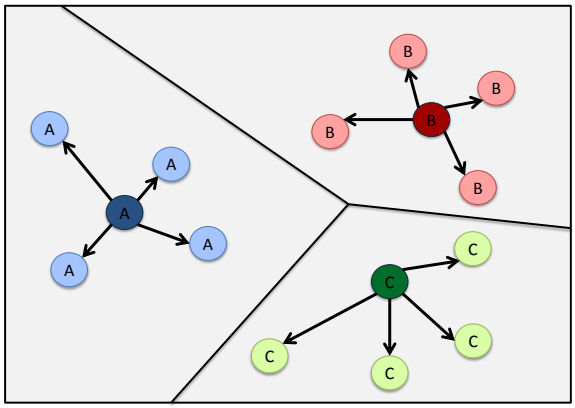


Abbildung 7: Population mit Clustern (nach [LC99])

In Versuchen konnte bestätigt werden, dass die Methode der Clusterbildung zu einer Verringerung der Anzahl von Generationen bei vergleichbarer Konvergenz führen kann. Wird beispielsweise

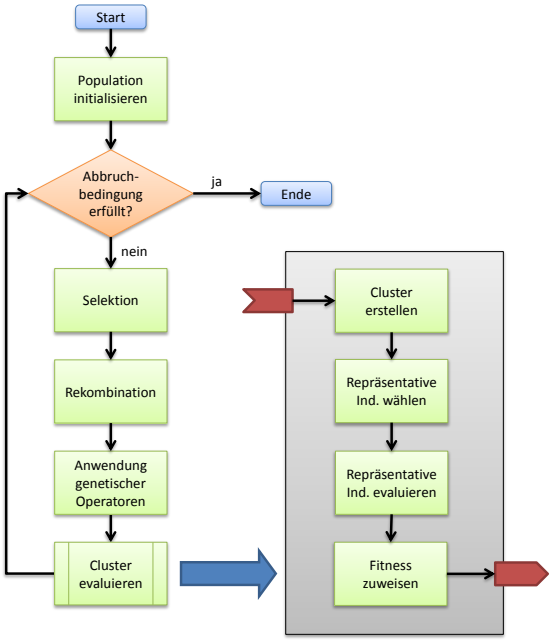


Abbildung 8: Einfacher IEA mit Clustering (nach [LC99])

eine Populationsgröße von 100 bei zehn Generationen gewählt, ergibt sich als Gesamtanzahl von Individuen $100 \cdot 10 = 1000$. Wird hier die Methode der Clusterbildung mit einer Clustergröße von zehn angewendet, müssen lediglich $\frac{100}{10} \cdot 10 = 100$ Individuen evaluiert werden. Voraussetzung für die Anwendung der Methode ist, dass ähnliche Genotypen auch ähnliche Phänotypen erzeugen.

3.2.3 Fixieren von Merkmalen

Der Autor Hideyuki Takagi stellt in [Tak00] eine Methode vor, bei der der Suchraum durch einen aktiven Eingriff des Benutzers reduziert wird. Entdeckt der Benutzer ein Merkmal, das ihm gefällt und das daher nicht weiter evolviert werden soll, so kann er dieses fixieren. Als Beispiel wird die Generierung von Gesichtsbildern aus sechs einzelnen Merkmalen vorgestellt: Haare, Augenbrauen, Augen, Nase, Mund, Gesichtsform. Jedes Merkmal kann 30 Zustände annehmen. Der Suchraum hat somit eine Größe von $30^6 = 729$ Millionen. Wenn der Benutzer nun ein Merkmal, beispielsweise die Augen, fixiert, weil sie den gesuchten Augen entsprechen, werden diese vom genetischen Algorithmus nicht weiter verändert. Der Suchraum wird dadurch auf $30^5 = 24.3$ Millionen reduziert.

Versuche haben ergeben, dass durch die Methode die zur Generierung des gesuchten Gesichts benötigte Anzahl an Generationen gegenüber dem herkömmlichen IEA auf 70% reduziert werden konnte. Die Methode birgt jedoch zwei Risiken. Zum einen erfordert die zusätzliche Möglichkeit des Fixierens von Merkmalen mehr Konzentration beim Benutzer, denn zum Fixieren muss er zusätzliche Buttons anklicken. Zum anderen legt sich der Benutzer durch das Fixieren eines Merkmals auf dieses fest. Potenziell bessere Lösungen werden dem Benutzer nicht mehr präsentiert. Die Methode eignet sich hingegen gut, wenn es ein absolutes Optimum für Merkmale gibt. Sobald dieses Optimum erreicht ist, kann der Benutzer dieses fixieren und beschleunigt so die Konvergenz des Algorithmus.

3.2.4 Direkte Manipulation

Bei der in [LKC01] vorgestellten Methode greift der Benutzer ebenfalls direkt in die Evolution ein. Als Beispiel führen die Autoren hier ein System zum evolutionären Modedesign an. Dem Benutzer werden auf dem Bildschirm pro Generation acht Entwürfe präsentiert. Der Benutzer kann hier jederzeit eine aktive Rolle im Designprozess einnehmen und Merkmale festlegen. Wünscht er beispielsweise lange Ärmel, so kann er diese, unabhängig davon was gerade angezeigt wird, über ein Kontextmenü auswählen. Nachfolgende Generationen von Kleidern werden nur noch über lange Ärmel verfügen. Der Benutzer definiert also, welche Eigenschaft der Phänotyp besitzen soll und verändert damit den Genotyp. Er beobachtet nicht nur den Verlauf der Evolution und gibt seine Bewertungen ab, sondern kann aktiv daran teilnehmen. Dieser Prozess gestaltet sich weniger monoton und sollte den Benutzer nicht so schnell ermüden lassen.

Die Autoren führten Versuche mit dem System für Modedesign mit zehn und 20 Generationen durch. Bei dem Versuch mit zehn Generationen waren durch die Möglichkeit der direkten Manipulation gegenüber dem normalen IEA durchschnittlich 6.57 Generationen weniger erforderlich, bis ein geeigneter Modeentwurf gefunden wurde. Dabei muss erwähnt werden, dass Benutzer bereits in der ersten Generation zu einem zufriedenstellenden Ergebnis kommen können, wenn sie die direkte Manipulation benutzen. Da in diesem Fall der evolutionäre Ansatz nicht mehr benutzt wird, kann die Eignung zur Vermeidung von Benutzerermüdung nicht objektiv bewertet werden.

3.2.5 Automatische Fitnessdiskretisierung

In [GYY09] wird eine Methode vorgeschlagen, welche dem in Kapitel 2.5.2 beschriebenen Prinzip von Gefallen und Abneigung diskrete Größen hinzufügt. Dazu wurde untersucht, welchen subjektiven Eindruck der Benutzer vom Individuum hat und wie lange er benötigt, um sein Gefallen oder seine Abneigung durch Anklicken des entsprechenden Buttons auszudrücken. Die Autoren trafen daraufhin folgende Annahmen: Der Benutzer braucht nur eine sehr kurze Zeit, um das Individuum positiv zu bewerten, welches er am meisten bevorzugt. Analog dazu benötigt er nur eine sehr kurze Zeit, um das Individuum negativ zu bewerten, welches ihm am wenigsten gefällt. Je länger der Benutzer benötigt, um ein bestimmtes Individuum positiv oder negativ zu bewerten, desto geringer ist sein Gefallen oder seine Ablehnung. Die Autoren erstellten eine Funktion, die eine Abbildung des Zeitraums auf den Fitnessraum darstellt. Diese kann in [GYY09] nachgelesen werden.

Nach einer Auswertung der durchgeführten Versuche konnte keine eindeutige Aussage darüber getroffen werden, ob die Konvergenz des IEA durch den Einsatz der Methode beschleunigt werden kann, da die Ergebnisse unterschiedlich ausfielen. Die Autoren betonen, dass weitere Untersuchungen darüber notwendig seien, wie die Methode verbessert werden kann. Weiterhin muss berücksichtigt werden, dass der Benutzer die Evaluation unterbrechen könnte, beispielsweise wenn er einen Telefonanruf bekommt oder eine Pause macht. In diesem Fall kann die Evaluationszeit in dieser Generation nicht mehr zur Fitnessberechnung verwendet werden.

3.2.6 Prognose der Fitnesswerte

In [Tak01] wird beschrieben, dass dem Benutzer Arbeit abgenommen werden kann, indem Fitnesswerte vorhergesagt werden. Die Idee bei dieser Methode ist, dass eine große Population verwendet werden kann, für die Individuen ein Fitnesswert vorberechnet wird und nur die besten Individuen

dem Benutzer zur Evaluation präsentiert werden. Die Cluster-Methode in Kapitel 3.2.2 verwendet einen ähnlichen Ansatz, beschränkt sich jedoch auf eine einfache Zuweisungsfunktion. Die Vorlieben des Benutzers können jedoch auch „gelernt“ werden, indem während der Evaluation neuronale Netze, Support Vector Machines oder Fuzzy-Regler trainiert werden [KYZ+06]. Die Autoren erhoffen sich, dass durch eine Fitnessprognose nur noch jede *n*-te Generation vom Benutzer evaluiert werden muss.

Die Versuche in [KYZ+06] haben ergeben, dass die Fuzzy-Regler in durchschnittlich 77% der Fälle die Vorlieben des Benutzers vorhersagen konnten. Der menschliche Aufwand konnte so im Durchschnitt um 51% verringert werden. Auch bei den anderen maschinellen Lernmethoden, die untersucht wurden, ergaben sich Verbesserungen der Leistung des IEA. Die Methode kann weiterhin verbessert werden, wenn der Prognose eine Prognosesicherheit hinzugefügt wird. Es müssen dann nur Individuen mit einer geringen Prognosesicherheit dem Benutzer zur Evaluation präsentiert werden.

3.2.7 Visualisierung des Suchraums

Die Autoren Hayashida und Takagi stellen in [HT00] eine Methode vor, die sie als „Visualized IEC“, also visualisierter interaktiver evolutionärer Algorithmus (im Folgenden mit VIEA abgekürzt) bezeichnen. Evolutionäre Algorithmen suchen in einem *n*-dimensionalen Suchraum nach möglichen Optima[6]. Sollen beispielsweise vier Parameter optimiert werden, so ergibt sich ein vierdimensionaler Suchraum. Die Idee der Methode ist, dem Benutzer die Möglichkeit zu geben, selbst Individuen im Suchraum zu erzeugen, wo sie eine gute Lösung vermuten. Der Bildschirm des Computers bietet allerdings nur zwei Dimensionen zur Darstellung an. Um den *n*-dimensionalen Suchraum auf einen zweidimensionalen Darstellungsraum abzubilden, verwenden die Autoren Selbstorganisierende Karten[7]. In Abbildung 9 ist der Unterschied zwischen einem herkömmlichen IEA und dem VIEA dargestellt. Der Benutzer hat hier die Möglichkeit, direkt in die Evolution einzugreifen.

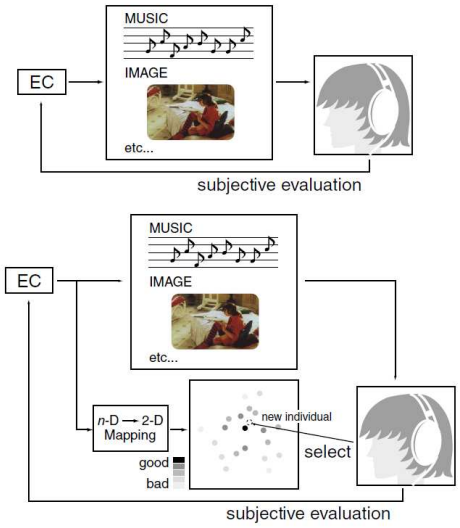


Abbildung 9: IEA (oben) und VIEA (unten) gegenübergestellt (aus [HT00])

[6] Anmerkung des Autors: EA suchen kein absolutes Optimum, sondern eine oder mehrere gute Lösungen

[7] Selbstorganisierende Karten bilden mittels neuronaler Netze *n*-dimensionale Merkmalsräume auf eine zweidimensionale Karte ab. Punkte mit einer geringen Distanz auf der Karte haben somit auch ähnliche Merkmale [BHS07]

Abbildung 10 zeigt eine Benutzerschnittstelle zur Sprachverarbeitung mit Hilfe des VIEA. Der Benutzer hört sich in jeder Generation 20 Sprachproben, also die Phänotypen des Algorithmus, an und kann diese in fünf Stufen bewerten. Durch einen Mausklick auf die Selbstorganisierende Karte rechts im Bild werden an der entsprechenden Stelle im Parameterraum neue Individuen erzeugt.

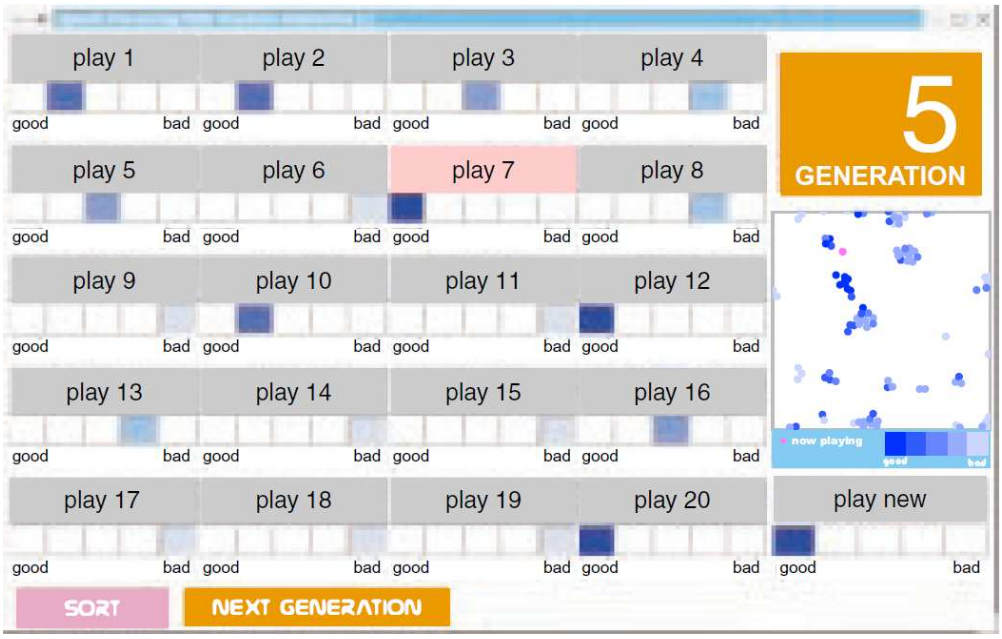


Abbildung 10: Benutzerschnittstelle für einen VIEA zur Sprachverarbeitung (aus [HT00])

Die Versuche in [HT00] haben ergeben, dass die Konvergenz des VIEA mit einer Populationsgröße von 20 der eines herkömmlichen IEA mit Generationsgrößen von 100 und 1000 entsprach. Daraus resultierend erwarten die Autoren, dass der Algorithmus fünfmal schneller als ein IEA konvergiert. Weiterhin wurde bei Versuchen bestätigt, dass der VIEA bei drei und bei fünf Merkmalsdimensionen ähnlich konvergierte, während die Leistung eines IEA stark von der Anzahl der Dimensionen abhängt. Die Methode hat allerdings den großen Nachteil, dass die Selbstorganisierende Karte in jeder Generation trainiert werden muss. Der Versuch ergab, dass das Trainieren einer Selbstorganisierenden Karte mit $200 \cdot 200 = 40000$ Punkten zu lange dauert, um eine flüssige Benutzerinteraktion zu ermöglichen. Des Weiteren ist die Bedienung nicht intuitiv, denn der Benutzer benötigt Kenntnisse über die Funktion der Selbstorganisierenden Karte.

3.2.8 Parallele Bewertung

Der Autor Robert Fischer beschreibt in [Fis08] eine Methode, bei der mehrere Benutzer parallel die Evaluation durchführen. Dabei bewertet jeder Benutzer nach anderen Optimierungskriterien. Als Beispiel wird hier eine Anwendung zur Generierung von CSS-Layouts angeführt, bei dem ein Benutzer nur die Farbgebung, ein zweiter nur die Anordnung und ein dritter Benutzer nur die Schriftgestaltung bewertet. Für die einzelnen Benutzer bedeutet dies eine Verkleinerung des Suchraums und damit auch eine Verringerung der Ermüdung. Die Methode ermöglicht eine Mehrzieloptimierung. Denkbar wäre, dass die so erzeugten Populationen nach der Evaluation durch eine Rekombinationsregel, die besagt dass ein Individuum nur mit einem Individuum aus einer anderen Population gekreuzt werden darf, zusammengeführt werden. Dieser Schritt sollte ohne Mutation erfolgen.

Diese Methode existiert bisher nur als Konzept, daher wären Versuche notwendig, um Aussagen darüber treffen zu können, ob die Methode gut funktioniert. Robert Fischer betont, dass die Personen des Evaluationsteams die Lösungsparameter, beispielsweise Farbkontraste, möglichst homogen bewerten sollten. Sind die Vorstellungen des Optimums hingegen sehr unterschiedlich, würde sich diese Methode eher nachteilig auswirken.

3.3 Vorschläge für neue Methoden

3.3.1 Optimierung der Parameter des IEA

Evolutionäre Algorithmen besitzen eine Reihe von Parametern, die die Konvergenz maßgeblich beeinflussen können. Wird beispielsweise ein genetischer Algorithmus mit interaktiver Fitness gewählt, so bietet dieser folgende Parameter an: die Populationsgröße, die Kreuzungswahrscheinlichkeit P_c, die Mutationswahrscheinlichkeit P_m, die maximale Abweichung der Chromosomen bei einer Mutation, die Art der Selektion, spezifische Selektionsparameter (z.B. bei Turnierselektion die Turniergröße), die Art der Mutation, spezielle Regeln (z.B. ob die Elitestrategie verwendet werden soll), und viele weitere. Diese Parameter sind schwierig festzulegen und können oft nur durch Versuche ermittelt werden. Es bietet sich jedoch auch an, sie durch einen genetischen Algorithmus zu optimieren. Das könnte wie folgt ablaufen: Die Parameter des IEA werden bei jedem Durchlauf durch ein Individuum des GA repräsentiert. Die Anzahl der evaluierten IEA-Individuen, die benötigt werden um eine gute Lösung zu finden, bestimmen die Fitness der GA-Individuen. Bricht der Benutzer die Evaluation vorzeitig ab, ist die Fitness minimal. Die GA-Population sollte beim Beenden des IEA persistent gespeichert werden, damit die Ergebnisse nicht verloren gehen.

Eine solche Optimierung könnte gute Parameter für eine schnelle Konvergenz des IEA finden. Gleichzeitig wäre sie robust gegenüber der Varianz zwischen verschiedenen Benutzern. Dies ist jedoch nur ein Konzept, welches durch praktische Versuche belegt werden muss.

3.3.2 Verwendung anfassbarer Benutzerschnittstellen

Während sich herkömmliche Benutzerschnittstellen wie Tastatur und Maus in den letzten Jahren kaum noch verändert haben, sind neue Benutzerschnittstellen (siehe Kapitel 2.6) in ihrer Technik und Anwendung zur Reife gelangt. Multi-Touch-Bildschirme sind inzwischen in vielen Mobiltelefonen und tragbaren Computern zu finden. Die hohen Verkaufszahlen lassen vermuten, dass die neue Art der Benutzerinteraktion von Kunden gut angenommen wird. Ein Grund dafür könnte darin liegen, dass hier die Eingabe direkt am Ort der Darstellung von Menüs, Buttons, Bildern und weiteren Objekten stattfindet. Diese lassen sich „anfassen“ und können durch Berührungen verschoben, skaliert und gedreht werden. Diese Art der Eingabe ist der natürlichen Interaktion von Menschen mit ihrer Umgebung viel ähnlicher als die Bedienung unnatürlicher Objekte wie Maus oder Tastatur. In Kapitel 3.2.1 ist beschrieben, dass eine Benutzerschnittstelle für interaktive evolutionäre Algorithmen möglichst einfach gehalten sein sollte.

Die Verwendung anfassbarer Benutzerschnittstellen könnte Potenziale bieten, um die Benutzerermüdung bei IEA zu verringern und den Evaluationsvorgang mehrbenutzerfähig zu machen.

3.4 Taxonomie der Methoden

Anhand der Eigenschaften der Methoden zur Verringerung der Benutzerermüdung wurde die Taxonomie in Tabelle 3.4 aufgestellt. Hier wurde zusammengefasst, um welche Art von Methode es sich jeweils handelt und welche Anforderungen an den IEA gestellt werden. Bei der Manipulation greift der Benutzer direkt in die Evolution ein, statt nur fertige Lösungen zu bewerten. Teilbewertung bedeutet, dass der Benutzer nicht mehr die gesamte Population bewerten muss. Der Begriff Fitnesstransformation beschreibt, dass statt einer Zuweisung der Fitness durch den Benutzer hier von einem anderen Raum in den Fitnessraum transformiert wird. Des Weiteren nehmen einige Methoden eine Optimierung der Benutzerschnittstelle oder des evolutionären Algorithmus vor. Außerdem wurden die Anforderungen der Methoden mit in die Tabelle aufgenommen. Verschiedene Methoden stellen besondere Anforderungen an die Darstellung von Individuen, die Art der Fitnesseingabe oder benötigen zusätzliche Benutzereingaben. Viele Methoden erfordern auch eine Anpassung des evolutionären Algorithmus, beispielsweise bei Selektion und Fitnesszuweisung.

Ein bewertender Vergleich der Methoden konnte nicht durchgeführt werden, da die Versuchsbedingungen der zu Grunde liegenden Studien nicht einheitlich waren. Einige Methoden wurden bisher noch nicht durch wissenschaftliche Versuche belegt.

Nr.	Methode	Quellen	Art der Methode					Besondere Anforderungen an			
						Optimierung					
			Manipulation	Teil-bewertung	Fitness-transformation	Benutzer-schnittstelle	EA	Darstellung	Eingabe der Fitness	zusätzliche Eingaben	Algorithmus
1	Einfachheit der Benutzerschnittstelle	[Tak01][LSA+06] [LC99][GYY09]	—	—	—	×	—	×	×	—	—
2	Clusterbildung	[LC99]	—	×	—	—	—	—	—	—	×
3	Fixieren von Merkmalen	[Tak00]	×	—	—	—	—	—	—	×	×
4	Direkte Manipulation	[LKC01]	×	—	—	—	—	—	—	×	×
5	Automatische Fitnessdiskretisierung	[GYY09]	—	—	×	—	—	—	×	—	—
6	Prognose der Fitnesswerte	[KYZ+06]	—	×	—	—	—	—	—	—	×
7	Visualisierung des Suchraums	[HT00]	×	—	—	—	—	×	—	×	×
8	Parallele Bewertung	[Fis08]	—	×	—	—	—	×	×	—	×
9	Optimierung der Parameter des IEA		—	—	—	—	×	—	—	—	×
10	Verwendung anfassbarer Benutzerschnittstellen		—	×	—	×	—	×	×	—	—

×Eigenschaft vorhanden

—Eigenschaft nicht vorhanden

Tabelle 1: Taxonomie der Methoden zur Verringerung der Benutzerermüdung

4 Benutzerschnittstellen

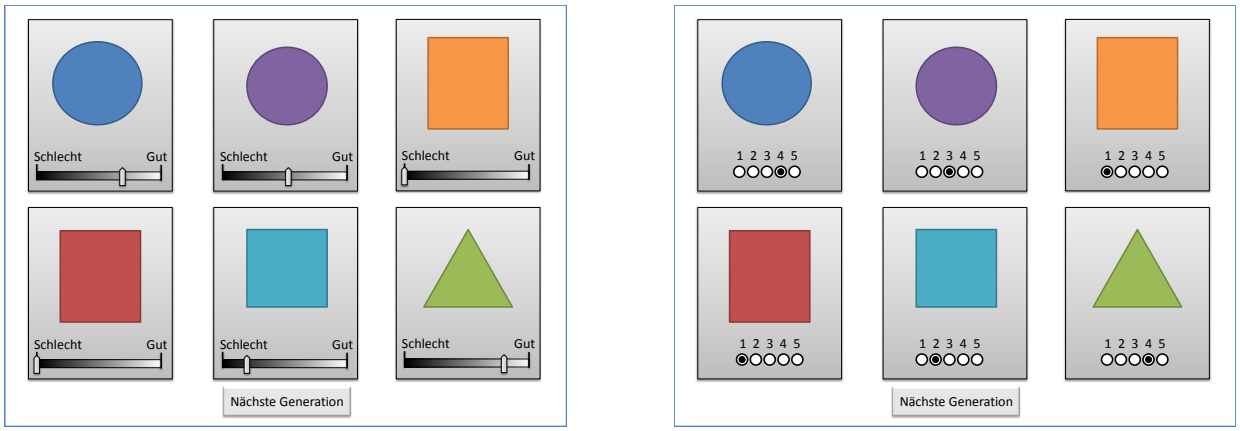
4.1 Einführung

In Kapitel 3.3.2 wurde bereits die Idee formuliert, Multi-Touch-Schnittstellen zur Bedienung von IEA zu verwenden. Die Eingabe kann hier mit Berührungen und Gesten erfolgen. Die grafische Ausgabe kann ebenfalls einfacher und natürlicher gestaltet werden. Statt Menüs oder Textfelder können hier Objekte verwendet werden, mit denen interagiert werden kann. Der Eindruck der Natürlichkeit kann noch verstärkt werden, wenn diese Objekte miteinander kollidieren können und sich mit einer physikalischen Trägheit bewegen. Bei Mobiltelefonen reagieren beispielsweise Listen mit einer gewissen Trägheit, wenn man durch diese scrollt. Doch auch bei diesem „natürlicheren“ Eingabekonzept muss darauf geachtet werden, dass der Benutzer nicht überstrapaziert wird. Die Anzahl der Interaktionen zum Erreichen eines Ziels sollte gering gehalten werden. Außerdem dürfen Berührungsfelder nicht zu klein gestaltet sein, denn die Auflösung von Touchscreens ist geringer als die von modernen Computermäusen.

4.2 Klassische Modelle

4.2.1 Kachelmodell

Viele Anwendungen von IEA wurden mit klassischen Eingabeschnittstellen wie Tastatur und Maus umgesetzt. Diese Konzepte stammen meist aus den Bereichen Forschung und Technik. Dort wird oft auf eine präzise und sterile Form von Benutzerschnittstellen Wert gelegt. Dies ist optisch weniger ansprechend und führt schnell zur Ermüdung des Benutzers. Die in Kapitel 3.2.1 beschriebenen Forderungen nach einer einfachen und abwechslungsreichen Benutzerschnittstelle werden selten erfüllt. Oft wird eine Benutzerschnittstelle verwendet, die einen Auszug oder die gesamte aktuelle Population in Form von Kacheln auf dem Bildschirm darstellt.



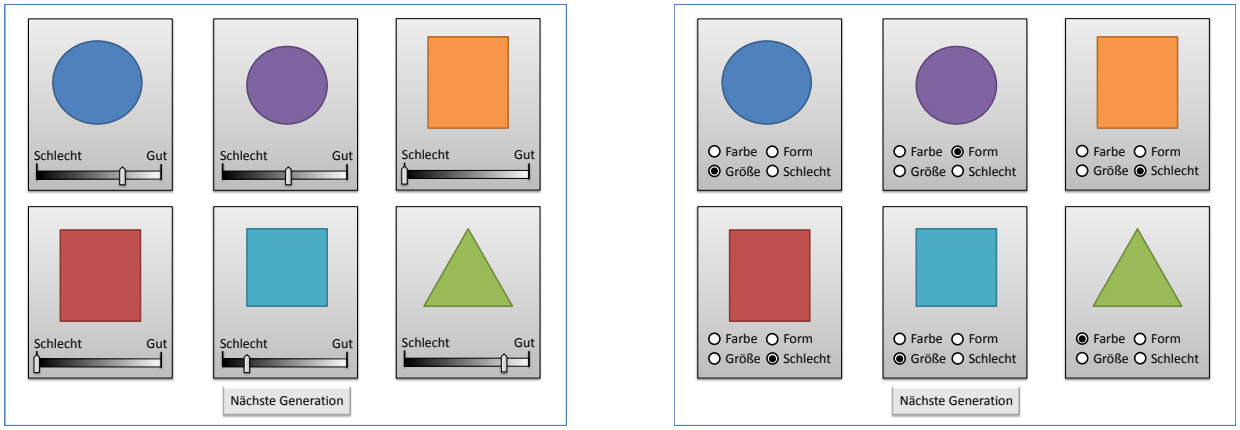
(a) Eingabe kontinuierlicher Werte per Schieberegler

(b) Bewertung mit einer festen Anzahl von Werten

Abbildung 11: Direkte Fitnessbewertung bei herkömmlichen Benutzerschnittstellen

Abbildung 11 zeigt zwei Varianten der direkten Zuweisung, die in Kapitel 2.5.1 eingeführt wurde. Abbildung 11(a) zeigt eine Schnittstelle, die dem Benutzer ermöglicht, den Individuen stufenlos einen Fitnesswert zuzuweisen. Wie in Kapitel 3.2.1 beschrieben wurde, ist eine hohe Anzahl von

Bewertungsabstufungen jedoch kontraproduktiv. Abbildung 11(b) zeigt die selbe Schnittstelle mit einem diskreten, fünfstufigen Wertebereich. Der Benutzer kann die Individuen auf einer Skala von eins (schlecht) bis fünf (gut) bewerten. Diese geringer aufgelöste Einteilung bedeutet schon eine Entlastung des Benutzers. Bei diesem generationsbasierenden IEA bestätigt der Benutzer durch das Anklicken des Buttons „Nächste Generation“, dass er mit der Bewertung fertig ist. Daraufhin wird eine neue Generation erzeugt und dem Benutzer präsentiert.



(a) Eingabe von Gefallen oder Ablehnung

(b) Auswahl des besten Kriteriums

Abbildung 12: Indirekte Fitnessbewertung bei herkömmlichen Benutzerschnittstellen

Abbildung 12 zeigt zwei weitere Formen der Benutzerschnittstelle, bei denen der Benutzer Fitnesswerte nicht direkt zuweisen muss. Bei der in Abbildung 12(a) gezeigten Schnittstelle kann der Benutzer wie in Kapitel 2.5.2 beschrieben Gefallen oder Abneigung bezüglich der Individuen ausdrücken. In Abbildung 12(b) ist eine Schnittstelle dargestellt, die die Bewertung nach verschiedenen Kriterien ermöglicht. Gefällt dem Benutzer beispielsweise die Farbe des Objekts, so kann er dieses Kriterium anklicken. Leider ist eine Bewertung mehrerer Kriterien gleichzeitig hier nicht möglich.

4.2.2 Eignung für Mehrbenutzer-Umgebungen

Die vier vorgestellten Benutzerschnittstellen eignen sich durch die verwendeten Bedienelemente, die eine hohe Eingabeauflösung erfordern, nicht für einen Einsatz an Multi-Touch-Bildschirmen. Eine gleichzeitige Evaluation durch mehrere Benutzer an einem Gerät ist ebenfalls nicht möglich, da der Text der Bedienelemente nur aus einer Richtung lesbar ist.

4.2.3 Paarweiser Vergleich

Die Benutzerschnittstelle in Abbildung 13 ist minimal gehalten, so dass der Benutzer bei der Bewertung nichts über mögliche Wertebereiche und Attribute wissen muss und wenig Eingaben tätigen muss. Um auszudrücken, welches der beiden Individuen ihm besser gefällt, genügt ein Mausklick. Der Benutzer erhöht so den Rang des Individuums. Sind beide Individuen subjektiv gleichrangig, kann der Benutzer ohne Abgabe einer Bewertung auf den Button „weiter“ klicken. Bei diesen Konzept wäre durch die Art der Eingabe eine Mehrbenutzer-Fähigkeit gegeben, jedoch ist sie unsinnig, da sich die Benutzer bei der Bewertung abwechseln müssten. Dieses Konzept erfüllt zwar die Anforderungen an die Einfachheit der Benutzerschnittstelle und ist durch die großen Interaktionsbereiche auch für anfassbare Benutzerschnittstellen geeignet, führt jedoch zu anderen

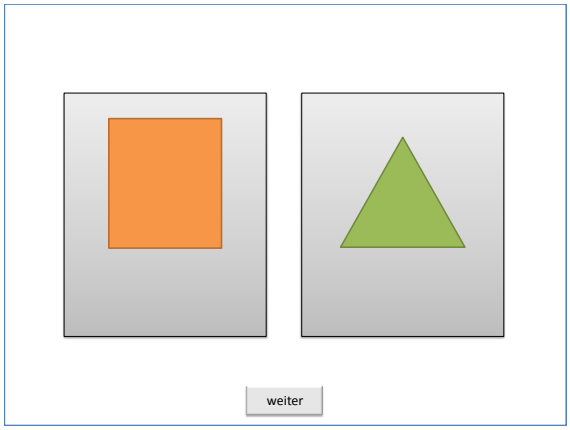


Abbildung 13: Indirekte Bewertung durch Auswahl des besseren Individuums

Problemen. Die Konvergenz dieses IEA ist gering. Da immer nur zwei Individuen zur Auswahl stehen, müssen sehr viele Evaluationsschritte durchgeführt werden. Nur wenn alle Individuen bewertet wurden, kann eine Rangordnung erstellt werden. Um die Relationen der Individuen untereinander bestimmen zu können, müssen einzelne Individuen dem Benutzer mehrfach präsentiert werden.

4.3 Innovative Konzepte

4.3.1 Attributpyramide

Robert Fischer beschreibt in [Fis08] das Konzept einer Attributpyramide. Das Modell ist in Abbildung 14 dargestellt. Unten im Bild ist die Attributpyramide zu sehen, auf der bis zu drei Parameter dargestellt werden können. Die Bilder der Individuen (Phänotypen) fahren von links nach rechts wie auf einem Laufband über den Bildschirm. Wenn diese den Bildschirm rechts verlassen, ohne dass der Benutzer eingegriffen hat, wird ihnen die Bewertung „Neutral“ gegeben. Der Benutzer kann per Drag-and-Drop die Bilder auf ein Attribut der Pyramide ziehen, wie der gestrichelte Pfeil andeutet. Das entsprechende Attribut wird dann positiv bewertet. Wird das Bild auf einen Kreis auf der Grenzlinie zwischen zwei Attributen gezogen, so werden die beiden angrenzenden Attribute positiv bewertet. Legt der Benutzer das Bild in der Mitte der Pyramide ab, so bewertet er alle drei Attribute positiv. Das Modell ermöglicht bei drei Attributen acht Bewertungsabstufungen.

Dieses Modell kann als Strom-Modell klassifiziert werden. Ein Strom von Individuen zieht am Benutzer vorbei und dieser nimmt sich diejenigen heraus, die er als interessant erachtet. Strom-Modelle eignen sich gut für eine Steady-State-Arbeitsweise, können jedoch auch generationenbasiert implementiert werden. Sie eignen sich ebenfalls für die parallele Bewertung durch mehrere Benutzer an einem Multi-Touch-System.

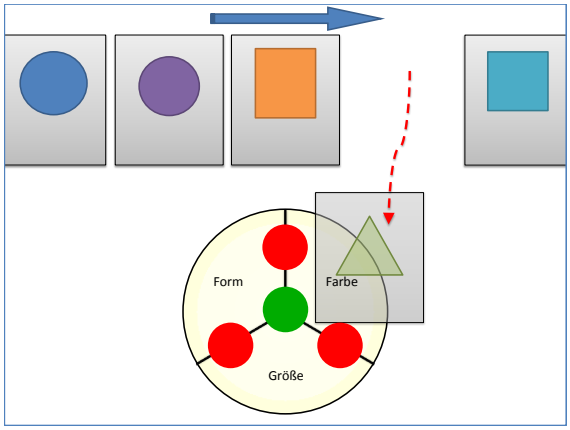


Abbildung 14: Bewertung nach Kriterien mittels einer Attributpyramide (nach [Fis08])

4.3.2 Strom-Modell mit Attributzonen

Robert Fischer stellt in [Fis08] ein weiteres Konzept für eine mehrkriterielle Bewertung vor, welches in Abbildung 15 dargestellt ist. Bei diesem Modell handelt es sich ebenfalls um ein Strom-Modell, bei dem Bilder von Individuen von links nach rechts über den Bildschirm fahren. Für die Bewertung der einzelnen Attribute stehen Zonen zur Verfügung, auf denen die Bilder per Drag-and-Drop abgelegt werden können. Der Benutzer hat hier auch die Möglichkeit, das ganze Individuum mit allen Attributen positiv oder negativ zu bewerten. Dazu muss das Bild auf der Plus- oder Minuszone abgelegt werden. Verlässt das Bild den rechten Bildschirmrand, wird das Individuum neutral bewertet. So ergeben sich bei drei Attributen sechs Abstufungen der Bewertung.

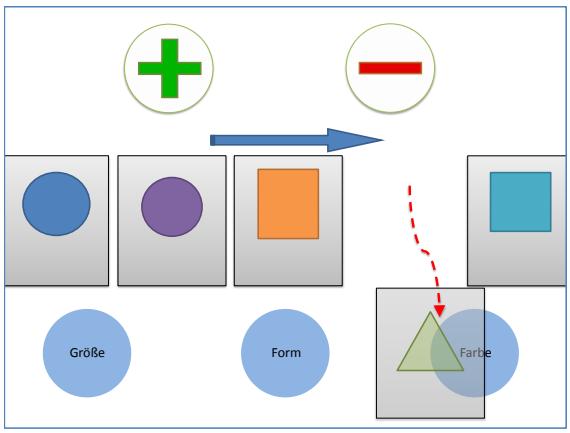


Abbildung 15: Bewertung nach Kriterien mittels Attributzonen (nach [Fis08])

4.3.3 Drei-Klassen-Modell

Eine weitere Variante des Strom-Modells ist das Drei-Klassen-Modell, welches ebenfalls in [Fis08] vorgestellt wird. Bei diesem Modell werden nicht einzelne Attribute, sondern nur ganze Individuen bewertet. Wie in Abbildung 16 zu sehen ist, erfolgt die Bewertung nach dem Prinzip von Gefallen und Abneigung. Die Bilder von Individuen fahren auf einem Laufband über den Bildschirm und können auf die Pluszone für eine positive, oder auf die Minuszone für eine negative Bewertung

gezogen werden. Auch hier gilt: verlassen die Bilder den rechten Bildschirmrand, erfolgt eine neutrale Bewertung. Dieses Modell bietet wenige Bewertungsabstufungen, ist intuitiv bedienbar und eignet sich auch für die gleichzeitige Evaluation durch mehrere Benutzer.

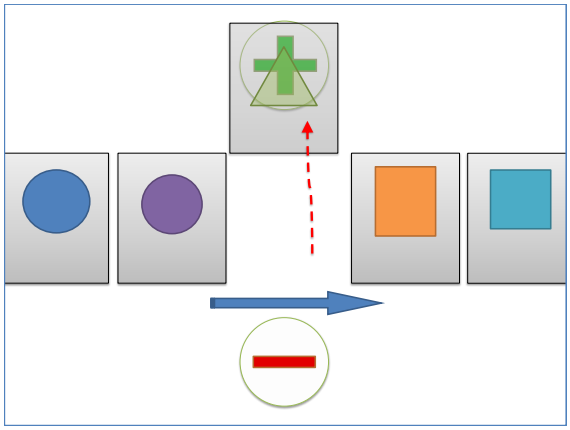


Abbildung 16: Dreistufige Bewertung mit dem Drei-Klassen-Modell (nach [Fis08])

4.3.4 Schauer-Modell

Das Schauer-Modell ist ein weiteres Konzept von Robert Fischer [Fis08]. Der Begriff bezieht sich auf den Regenschauer, ein örtlich begrenzter, kurz andauernder Niederschlag. Ähnlich einem Gärtner, der bestimmte Blumenbeete bevorzugt behandelt und nur diese gießt, bestimmt der Benutzer hier, welche Bereiche der Population von Individuen gegossen werden, also eine höhere Fitness erhalten. Der Kreis mit der gestrichelten Linie in Abbildung 17 stellt den Schauer dar. Wird dieser über die Bilder der Individuen bewegt, findet für diese eine Fitnesserhöhung statt. Die Individuen, die nicht gegossen werden, unterliegen einer Negativselektion. Das Alter ist dafür ein wichtiges Attribut. Mit zunehmendem Alter verschwinden die nicht gegossenen Individuen, sie „vertrocknen“. In der Abbildung sind die älteren Individuen durch ihre geringere Größe zu erkennen. Individuen, deren Alter einen vordefinierten Grenzwert übersteigen, werden aus der Population entfernt. Dadurch bekommt die restliche Population die Gelegenheit, sich auszubreiten, indem eine Selektion und anschließend eine Reproduktion durchgeführt wird. Das neue Individuum wird in der Nähe der Eltern in die Population eingefügt.

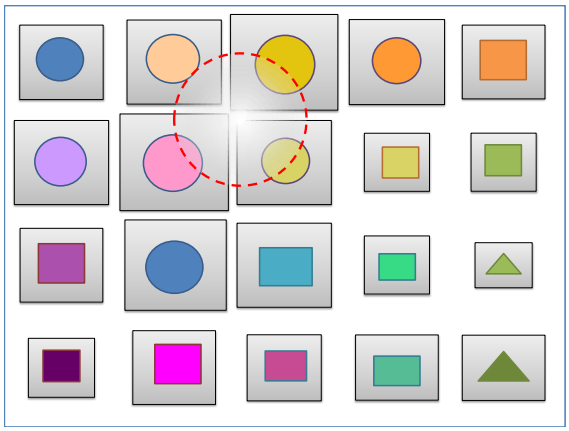


Abbildung 17: Bewertung einer geordneten Population mit dem Schauermodell (nach [Fis08])

Algorithmisch verlangt das Schauer-Modell nach einer Steady-State-Arbeitsweise. Es eignet sich vermutlich für eine parallele Evaluation durch mehrere Benutzer an Multi-Touch-Systemen. Hier wären mehrere verschiebbare Schauer denkbar. Eine Bedingung für das Modell ist, dass die Population geordnet ist. Um den Abstand der Individuen im Parameterraum zu bestimmen, eignet sich die Abstandsfunktion aus Kapitel 3.2.2.

4.3.5 Billardtisch-Modell

Dieses Modell ist eine abgewandelte Form des Schauermodells. Die Idee dahinter besteht darin, eine grafische Ordnung der Population indirekt zu erzeugen. Dies sieht so aus: die Phänotypen der Individuen liegen zunächst ungeordnet auf einem Tisch. Der Benutzer kann ein Elternindividuum zur Reproduktion auswählen, indem er es anklickt. Das zweite Elternindividuum wird nun mit einer berechneten Wahrscheinlichkeit selektiert. Diese Wahrscheinlichkeit ist abhängig vom Abstand der Individuen im Darstellungsraum, also auf dem Tisch. Die Abhängigkeit soll in Abbildung 18 verdeutlicht werden. Das Kreuz stellt die Position des Mausklicks dar. Zur Bestimmung der Selektionswahrscheinlichkeit wird hier die Gauß-Funktion verwendet. Im Bild sind die Werte der Funktion durch den Farbverlauf verdeutlicht: große Funktionswerte sind dunkler, kleine Funktionswerte sind heller dargestellt. Der gestrichelte Kreis stellt den Grenzwert der Funktion dar, daher findet die Selektion des zweiten Elternindividuums nur innerhalb dieses Kreises statt. Sobald ein zweites Individuum selektiert wurde, wird ein Crossover (optional mit Mutation) beider Individuen durchgeführt und das neue Individuum zwischen den Eltern eingefügt. Ist nicht ausreichend Platz vorhanden, so findet durch die Kollision eine physikalische Verdrängung statt. Da die Population eine maximale Größe nicht überschreiten sollte, wird das Alter der Individuen zur Negativselektion verwendet. Bei jeder Reproduktion wird das Alter aller Individuen erhöht. Individuen, deren Alter einen bestimmten Wert überschreitet, werden aus der Population entfernt.

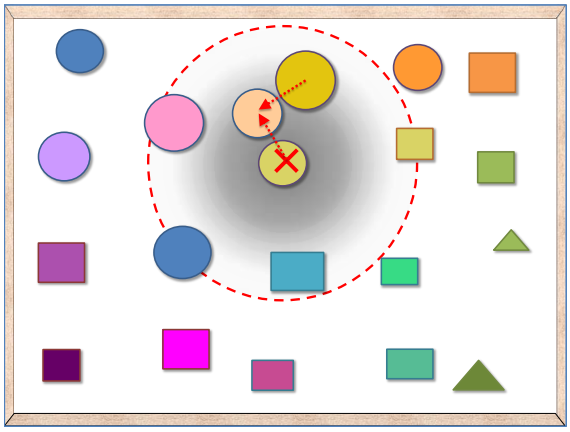


Abbildung 18: Direkte Auswahl zu reproduzierender Individuen mit dem Billardtisch-Modell

Dieses Modell verlagert Selektion, implizite Fitness und das Ordnen der Population in den Darstellungsraum. Die Evolution ist somit für den Benutzer transparenter als bei anderen Modellen. Zur Umsetzung des Modells eignet sich eine Physik-Engine. Eine parallele Evaluation durch mehrere Benutzer an einem Multi-Touch-System ist möglich, da die Individuen nur angetippt werden müssen.

4.3.6 Titanic-Modell

Das Titanic-Modell greift die Metapher des sinkenden Schiffes auf. In Abbildung 19 ist dies dargestellt. Die Individuen schwimmen auf dem Wasser. Einzelne Individuen werden negativ bewertet, indem sie durch Gewichte zum Sinken gebracht werden. Das bedeutet algorithmisch, dass die Individuen bei ihrer Erzeugung eine bestimmte Fitness besitzen. Die Belastung des Individuums mit Gewichten bedeutet eine Verringerung der Fitness. Das Alter der Individuen kann in die Fitnessbestimmung mit eingebracht werden, indem die Fitness in Abhängigkeit von der Lebensdauer verringert wird. Das Individuum versinkt allmählich. Sobald die Fitness einen Grenzwert unterschreitet, werden in der Population Selektion und Reproduktion durchgeführt und das alte Individuum wird durch ein neues ersetzt.

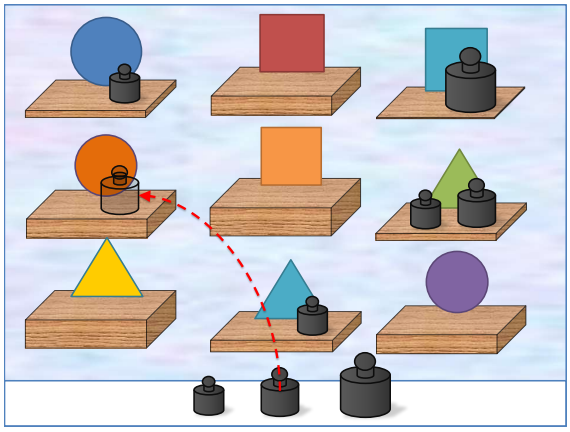


Abbildung 19: Negativbewertung durch das „Versenken“ schlechter Individuen beim Titanicmodell

4.4 Taxonomie der Modelle

Durch eine Analyse der hier aufgeführten Modelle und ihrer Eigenschaften wurde zusammenfassend die Taxonomie in Tabelle 4.4 aufgestellt. Die Art der Bewertung gliedert sich in mehrkriteriell (siehe Kapitel 2.5.3), diskrete Fitness (Kapitel 2.5.1) und Gefallen und Abneigung (Kapitel 2.5.2). Bei der Bewertung der einzelnen Individuen wird hier unterschieden, ob der Benutzer seine Auswahl positiv (die Selektionswahrscheinlichkeit angeklickter Individuen wird erhöht) oder negativ (die Selektionswahrscheinlichkeit angeklickter Individuen wird verringert) treffen kann. Bei der Art der Eingabe wird unterschieden, ob ein einfacher Klick oder eine Drag-and-Drop-Geste ausgeführt werden muss. Des Weiteren ist in der Tabelle erkennbar, ob sich das jeweilige Modell für eine generationsbasierte Arbeitsweise oder eine Steady-State-Arbeitsweise eignet. Außerdem werden weitere Eigenschaften gelistet: Die Mehrbenutzerfähigkeit, die Anforderungen an die Auflösung der Eingabe und die Anzahl notwendiger Interaktionen zur Bewertung einzelner Individuen. Die jeweiligen Angaben beziehen sich nur auf die vorgestellten Modelle, bei abgewandelten Modellen könnten die Eigenschaften variieren.

Nr.	Modell	Art der Bewertung			Selektion		Art der Eingabe		mögliche Arbeitsweise		Weitere Eigenschaften		
		mehr-kriteriell	diskrete Fitness	Gefallen und Abneigung	positiv	negativ	Klick	Drag-and-Drop	generations-basiert	Steady-State	mehr-benutzer-fähig	Auflösung der Eingabe	Interaktionen zur Bewertung
1	Kachelmodell mit kontinuierlichen Werten	—	×	—	×	×	×	—	×	—	—	hoch	viele
2	Kachelmodell mit fünf Stufen	—	×	—	×	×	×	—	×	—	—	hoch	viele
3	Kachelmodell für Gefallen und Abneigung	—	—	×	×	×	×	—	×	—	—	hoch	viele
4	Mehrkriterielles Kachelmodell	×	—	—	×	×	×	—	×	—	—	hoch	viele
5	Paarweiser Vergleich	—	—	×	×	—	×	—	×	—	—	gering	wenige
6	Attributpyramide	×	—	—	×	—	—	×	×	×	×	mittel	wenige
7	Strom-Modell mit Attributzonen	×	—	×	×	×	—	×	×	×	×	gering	wenige
8	Drei-Klassen-Modell	—	—	×	×	×	—	×	×	×	×	gering	wenige
9	Schauer-Modell	—	—	×	×	—	—	×	—	×	×	gering	wenige
10	Billardtisch-Modell	—	—	×	×	—	×	—	—	×	×	mittel	wenige
11	Titanic-Modell	—	×	—	—	×	×	—	—	×	×	mittel	wenige

×Eigenschaft vorhanden

—Eigenschaft nicht vorhanden

Tabelle 2: Taxonomie der Modelle zur Fitnessbewertung

4.5 Die Beispielanwendung

Im Rahmen der Arbeit wurde der Prototyp einer Anwendung zur Evaluation an Multi-Touch-Systemen entwickelt. Dieser basiert auf der jMonkeyEngine[8], einer in Java implementierten 3D-Engine mit Physiksimulation. In der Beispielanwendung wurden zwei Modelle implementiert. Abbildung 20 in Anhang A zeigt das Auswahlmenü der Modelle.

Das erste implementiere Modell ist das Drei-Klassen-Modell, dargestellt in Abbildung 21. Hier fahren Bilder von Individuen, die im Beispiel als einfache geometrische Körper mit verschiedenen Farben umgesetzt wurden, über den Bildschirm und der Benutzer kann diese zur Bewertung auf die Plus- oder Minuszonen ziehen, wie in Kapitel 4.3.3 bereits beschrieben wurde. In der Beispielanwendung funktionierte dies gut, jedoch wurde deutlich, dass die Konvergenz des Algorithmus unzureichend ist und noch optimiert werden muss. Es wurde hier keine der in Kapitel 3 vorgestellten Methoden zur Fitnessprognose implementiert. Des Weiteren zeigte sich, dass die Drag-and-Drop-Geste als primäre Form der Eingabe eine schlechte Ergonomie hat, da der Benutzer während der gesamten Geste Druck auf das Display ausüben muss. Dies ist als Vermutung anzusehen, daher sind noch weitere Untersuchungen erforderlich. Abbildung 22 wurde zu einem Zeitpunkt aufgenommen, als die Konvergenz schon erkennbar war. Hier ist zu sehen, dass die Population bereits viele orangene Tori enthält, nachdem diese vorher positiv selektiert wurden. Abbildung 23 zeigt die Funktionalität der dreidimensionalen Ansicht der Individuen. Der Benutzer kann die Kamera hier um den Körper rotieren und diesen aus allen Richtungen betrachten.

Das zweite Modell, welches implementiert wurde, ist das in Kapitel 4.3.5 vorgestellte Billardtisch-Modell. Die Individuen werden hier wie beim ersten Modell als einfache geometrische Körper mit verschiedenen Farben repräsentiert. Abbildung 24 zeigt die Benutzerschnittstelle nach der Initialisierung der Population. In Abbildung 25 ist die Selektion des ersten Individuums durch einen Mausklick dargestellt. Das zweite Individuum wird hier noch zufällig ausgewählt, eine Selektion in Abhängigkeit des Abstands wurde in der Beispielanwendung noch nicht implementiert. Der hell aufleuchtende Kubus unten rechts im Bild ist das Kind-Individuum der beiden Eltern, die ebenfalls hell aufleuchten. Die vorgeschlagene Negativselektion von Individuen ist ebenfalls noch nicht implementiert. In Abbildung 26 ist zu sehen, dass dadurch die Population bei jeder Reproduktion wächst. Die Ergonomie dieses Modells ist deutlich besser als beim ersten Beispielmodell. Die Primäre Geste zur Eingabe ist hier ein einfaches Antippen mit dem Finger oder ein Klick mit der Maus. In Anhang A sind alle Bildschirmfotos der Anwendung zu finden.

Da bisher keine Versuche mit belegten Ergebnissen durchgeführt wurden, sind die Angaben über die Eignung und Konvergenz der Anwendung als hypothetisch zu betrachten und erfordern noch weitere Untersuchungen.

[8] Nähere Informationen sind unter `http://jmonkeyengine.com/` zu finden

5 Fazit und Ausblick

Interaktive evolutionäre Algorithmen bieten in vielen Anwendungsbereichen die Möglichkeit, Optimierungen durchzuführen und dabei die Erfahrungen, Eindrücke, Vorlieben, Emotionen, Intuition und das Verständnis des Benutzers mit einfließen zu lassen. Jedoch führt ein langwieriger Optimierungsprozess zu einer Ermüdung des Benutzers. In der Fachliteratur und vom Autor dieser Arbeit werden dazu verschiedene Konzepte vorgeschlagen. Diese in Kapitel 3 vorgestellten Methoden lassen sich grob gruppieren. Mehrere Methoden haben den Ansatz, den Benutzer nicht die gesamte Population bewerten zu lassen. Dies wird entweder durch eine Vorhersage der Fitnesswerte oder eine Aufteilung der Bewertungsaufgabe auf mehrere Benutzer ermöglicht. Einige Methoden ermöglichen es dem Benutzer, direkt in die Evolution einzugreifen, um diese zu beschleunigen. Eine Methode hat das Ziel, die Fitnesswerte nicht direkt aus der Bewertung, sondern aus dem Verhalten des Benutzers während der Bewertung zu berechnen. Eine weitere Methode führt eine Optimierung der Parameter des evolutionären Algorithmus durch. Die Optimierung der Benutzerschnittstelle ist das Ziel der letzten beiden Methoden. Diese erfordern keine Modifikation des Algorithmus und können mit weiteren Methoden kombiniert werden. Die in der Literatur vorgeschlagenen Methoden sind in vielen Fällen mit Versuchsergebnissen belegt. Leider waren die Versuchsbedingungen sehr unterschiedlich, so dass diese nicht verglichen werden konnten. In weiterführenden Studien könnten entsprechende Beispielanwendungen zur Durchführung dieser Versuche implementiert werden. Für einen objektiven Vergleich wären Versuche unter gleichen Bedingungen bei einer Anwendung mit einer großen Anzahl von Benutzern notwendig. Um allgemeinere Aussagen treffen zu können, müssten die Methoden in mehreren verschiedenen Anwendungen getestet werden. Verschiedene Anwendungen erfordern oft verschiedene Arten von Fitness, Darstellung und Eingabe.

In Kapitel 4 wurden verschiedene Modelle für Benutzerschnittstellen vorgestellt, in einer Taxonomie zusammengefasst und verglichen. Diese lassen sich in statische Modelle, die oft nach einer generationsbasierten Arbeitsweise verlangen, und dynamische Modelle, die meistens eine Steady-State-Arbeitsweise des EA bedingen, unterteilen. Die Eigenschaften, die dort den Modellen zugeordnet wurden, beziehen sich streng auf die beschriebenen Modelle. Variationen der Modelle sind jedoch auch denkbar. So könnte die Eingabe beim Schauer-Modell aus Kapitel 4.3.4 verändert werden, so dass der Benutzer den Schauer durch einen einfachen Klick statt einer Drag-and-Drop-Geste verschieben kann. Dabei würde jedoch die Mehrbenutzer-Fähigkeit verloren gehen, da bei einem Klick nicht zugeordnet werden kann, welcher Schauer verschoben werden soll. In weiterführenden Arbeiten könnten die Modelle als Multi-Touch-Anwendungen implementiert und mit den Methoden aus Kapitel 3 kombiniert werden. Dann könnten diese in Versuchen mit Testpersonen mit den klassischen Eingabemodellen in Hinsicht auf die Benutzerermüdung und Konvergenz verglichen werden.

Die entwickelte Beispielanwendung befindet sich noch in einem frühen Stadium und kann nicht als vollständige Anwendung gesehen werden. Sie könnte in Zukunft weiter entwickelt werden um anschließend belegbare Versuche mit Testpersonen durchzuführen. Zur Verbesserung der Konvergenz könnten hier Methoden zur Fitnessprognose implementiert und getestet werden. Auch die Implementierung weiterer Eingabemodelle wäre denkbar.

Literatur

[BHS07] Boersch, Ingo, Heinsohn, Jochen und Socher, Rolf: *Wissensverarbeitung: Eine Einführung in die Künstliche Intelligenz für Informatiker und Ingenieure*. Spektrum Akademischer Verlag, 2007, ISBN 3827418445. 3, 4, 18, 36

[CL01] Chapuis, Jonathan und Lutton, Evelyne: *ArtiE-Fract: Interactive Evolution of Fractals*. 4th International Conference on Generative Art, Dezember 2001. 7

[Dar09] Darwin, Charles: *On the Origin of Species by Means of Natural Selection*. BiblioLife, LLC, 2009, ISBN 1116871319. 3

[Fis08] Fischer, Robert: *Entwicklung einer Applikation zur Layoutoptimierung von Webseiten mit evolutionären Algorithmen und interaktiver Fitness*. Bachelorarbeit, Fachhochschule Brandenburg, 18.08.2008. 19, 22, 25, 26, 27, 35

[GKK04] Gerdes, Ingrid, Klawonn, Frank und Kruse, Rudolf: *Evolutionäre Algorithmen: Genetische Algorithmen - Strategien und Optimierungsverfahren - Beispielanwendungen*. Vieweg, 2004, ISBN 3528055707. 1, 5, 10, 36

[GYY09] Gong, Dunwei, Yao, Xin und Yuan, Jie: *Interactive Genetic Algorithms with Individual Fitness not Assigned by Human*. Journal of Universal Computer Science, 15(13):2446–2462, 2009. 10, 12, 14, 17, 22, 35

[HT00] Hayashida, Norimasa und Takagi, Hideyuki: *Visualized IEC - Interactive Evolutionary Computation with Multidimensional Data Visualization*. IEEE International Conference on Industrial Electronics, Control and Instrumentation (IECON2000), Seiten 2738–2743, 27. Oktober 2000. 7, 8, 10, 18, 19, 22, 35

[IT09] Inoue, Makoto und Takagi, Hideyuki: *EMO-based Architectural Room Floor Planning*. IEEE International Conference on Systems, Man and Cybernetics (SMC2009), Seiten 524–529, Oktober 2009. 7

[KYZ+06] Kamalian, Raffi, Yeh, Eric, Zhang, Ying, Agogino, Alice M. und Takagi, Hideyuki: *Reducing Human Fatigue in Interactive Evolutionary Computation through Fuzzy Systems and Machine Learning Systems*. 2006 IEEE International Conference on Fuzzy Systems, Seiten 3295–3301, Juli 2006. 18, 22

[KZTA05] Kamalian, Raffi, Zhang, Ying, Takagi, Hideyuki und Agogino, Alice M.: *Reduced Human Fatigue Interactive Evolutionary Computation for Micromachine Design*. 4th International Conference on Machine Learning and Cybernetics (ICMLC 2005), Seiten 5666–5671, August 2005. 7, 11

[LC99] Lee, Joo Young und Cho, Sung Bae: *Sparse Fitness Evaluation for Reducing User Burden*. 1999 IEEE International Fuzzy Systems Conference Proceedings, Seiten 998–1003, August 1999. 14, 15, 16, 22, 35

[LC01] Lämmel, Uwe und Cleve, Jürgen: *Lehr- und Übungsbuch Künstliche Intelligenz*. Hanser, Carl, GmbH & Co., 2001, ISBN 3446214216. 1, 4

[LKC01] Lee, Jong Ha, Kim, Hee Su und Cho, Sung Bae: *Accelerating Evolution by Direct Manipulation for Interface Fashion Design*. 4th International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2001), Seiten 343–347, 2001. 7, 11, 17, 22, 35

[LSA+06] Llorà, Xavier, Sastry, Kumara, Alías, Francesc, Goldberg, David E. und Welge, Micheal: *Analyzing Active Interactive Genetic Algorithms using Visual Analytics*, Februar 2006. 10, 14, 22

[Nag99] Nagamachi, Mitsuo: *Kansei engineering and new product development: A special seminar on Kansei engineering by Mitsuo Nagamachi, Professor Emeritus, Hiroshima University ; the 11th Symposium on QFD, Novi, Mich., USA, June 12, 1999*. QFD Institute, [Novi, Mich, 1999. http://www.worldcat.org/oclc/53160528. 7

[RM07] Romero, Juan und Machado, Penousal: *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Springer, Dordrecht, 2007, ISBN 3540728775. http://www.worldcat.org/oclc/437206461. 8

[Sch10] Schmidt, Sebastian: *Konzeption und Implementierung eines Multitouch-Informationssystems*. Bachelorarbeit, Fachhochschule Brandenburg, 22.9.2010. 13, 35

[Tak00] Takagi, Hideyuki: *Active User Intervention in an EC Search*. International Conference on Information Sciences (JCIS2000), Seiten 995–998, März 2000. 8, 10, 16, 22

[Tak01] Takagi, Hideyuki: *Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation*. Proceedings of the IEEE, 89(9):1275–1296, September 2001. 1, 7, 8, 9, 10, 14, 15, 17, 22

[TI00] Tokui, N. und Iba, H.: *Music Composition with Interactive Evolutionary Computation*. 3rd International Conference on Generative Art, Seiten 215–226, 2000. 8

[TO00] Takagi, Hideyuki und Ohsaki, Miho: *IEC Fitting: New framework of hearing aid fitting based on computational intelligence technology and user's preference for hearing*. Poster Session PB9, Int. Hearing Aid Research Conf. (IHCON2000), Seiten 49–50, August 2000. 8

Abbildungsverzeichnis

1 Einfacher genetischer Algorithmus 4
2 Die Darstellungsfunktion f_d bildet Individuen vom Parameterraum in den physischen Raum ab 5
3 Die subjektive Bewertungsfunktion f_{sub} bildet Individuen vom psychischen Raum in den Suchraum ab 6
4 Benutzerschnittstelle für eine direkte Zuweisung einer diskreten Fitness durch den Benutzer (aus [LKC01]) 11
5 Benutzerschnittstelle, durch die der Benutzer Gefallen oder Abneigung ausdrücken kann (aus [GYY09]) 12
6 Gesten zur Bedienung von Multi-Touch-Systemen (nach [Sch10]) 13
7 Population mit Clustern (nach [LC99]) 15
8 Einfacher IEA mit Clustering (nach [LC99]) 16
9 IEA (oben) und VIEA (unten) gegenübergestellt (aus [HT00]) 18
10 Benutzerschnittstelle für einen VIEA zur Sprachverarbeitung (aus [HT00]) 19
11 Direkte Fitnessbewertung bei herkömmlichen Benutzerschnittstellen 23
12 Indirekte Fitnessbewertung bei herkömmlichen Benutzerschnittstellen 24
13 Indirekte Bewertung durch Auswahl des besseren Individuums 25
14 Bewertung nach Kriterien mittels einer Attributpyramide (nach [Fis08]) 26
15 Bewertung nach Kriterien mittels Attributzonen (nach [Fis08]) 26
16 Dreistufige Bewertung mit dem Drei-Klassen-Modell (nach [Fis08]) 27
17 Bewertung einer geordneten Population mit dem Schauermodell (nach [Fis08]) . . 27
18 Direkte Auswahl zu reproduzierender Individuen mit dem Billardtisch-Modell . . . 28
19 Negativbewertung durch das „Versenken“ schlechter Individuen beim Titanicmodell 29
20 Das Auswahlmenü für die Beispielmodelle 38
21 Fitnesszuweisung beim Drei-Klassen-Modell 38
22 Konvergenz in Richtung orangener Tori 39
23 Dreidimensionale Ansicht der Individuen 39
24 Initialpopulation des Billardtisch-Modells 40
25 Durchgeführtes Crossover zwischen den beiden hell aufleuchtenden Individuen . . 40
26 Ohne Negativselektion wird der Tisch immer voller 41

Tabellenverzeichnis

1 Taxonomie der Methoden zur Verringerung der Benutzerermüdung 22
2 Taxonomie der Modelle zur Fitnessbewertung 30

Glossar

EA
: siehe *evolutionärer Algorithmus*.

Evaluation
: Evaluation, auch Bewertung genannt, ist die Zuweisung von Fitnesswerten durch einen Benutzer.

evolutionärer Algorithmus
: Evolutionäre Algorithmen (kurz: EA) übertragen die natürliche Evolution auf technische Systeme. Sie stellen eine Möglichkeit zur Optimierung von Problemlösungen dar (vgl. [BHS07]).

Fitness
: Die Fitness beschreibt die Güte des Phänotyps und beeinflusst die Reproduktionswahrscheinlichkeit des Individuums. Sie kann als explizite Fitness (numerischer Wert) oder implizite Fitness (Rang in der Population) auftreten.

Fitnessfunktion
: Die Fitnessfunktion entspricht einer Zielfunktion bei Optimierungsproblemen. Sie bildet die Grundlage genetischer Algorithmen und stellt eine Abbildung dar, die jedem Individuum eine üblicherweise positive reelle Zahl als Bewertung zuweist. In der Literatur werden für die Fitnessfunktion oft auch die Synonyme Zielfunktion, Evaluierungsfunktion oder Bewertungsfunktion verwendet (vgl. [GKK04]).

generationenbasierte Arbeitsweise
: Arbeitsweise eines genetischen Algorithmus, bei der für jede Generation von Individuen Selektion, Anwendung genetischer Operatoren, Evaluation und Erzeugung einer neuen Generation durchgeführt werden. Die alte Generation wird anschließend verworfen. Den Gegensatz dazu bildet die *Steady-State-Arbeitsweise*.

genetischer Algorithmus
: Eine Form von Evolutionären Algorithmen, bei der der Genotyp eines Individuums als ein Vektor von Parametern oder als einziger Bitstring dargestellt wird. (vgl. [BHS07]) Der Terminus „Genetischer Algorithmus“ wird oft mit GA abgekürzt.

genetisches Programmieren
: Genetisches Programmieren (kurz: GP) ist ein *evolutionärer Algorithmus* zur Entwicklung von Funktionen oder Programmbäumen.

Genotyp
: Darstellung eines Individuums im Parameterraum, in der Literatur auch manchmal als Chromosom bezeichnet. Entspricht den Genen eines Lebewesens und kodiert die vererbbaren Informationen.

IEA
: siehe *interaktiver evolutionärer Algorithmus*.

Individuum
: Ein Individuum ist eine mögliche Problemlösung in der Optimierung mit Evolutionären Algorithmen.

interaktive Fitness
: Fitnesswert, der durch den Benutzer zugewiesen wird.

interaktiver evolutionärer Algorithmus
: *Evolutionärer Algorithmus* mit *interaktiver Fitness*.

Phänotyp
: Ausprägung der Eigenschaften, die durch den Genotyp des Individuums kodiert sind, ähnlich dem Körper eines Lebewesens.

Population
: Eine Population ist die Menge aller Individuen zu einem bestimmten Zeitpunkt.

Selektion
: Die Selektion bestimmt, welche Individuen fortgepflanzt oder ersetzt werden. Es handelt sich um eine Abbildung einer Individuenmenge I_1 auf eine andere Individuenmenge I_2. I_2 ist eine Teilmenge von I_1.

Steady-State-Arbeitsweise
: „Steady-State“, frei übersetzt „stabiler Zustand“, bedeutet, dass während des evolutionären Algorithmus immer nur einzelne Individuen ersetzt werden. Meist wird das Alter eines Individuums bei der Ersetzungsselektion mit berücksichtigt. Den Gegensatz dazu bildet die *generationenbasierte Arbeitsweise*.

Tangible User Interfaces
: Auf Deutsch: „anfassbare Benutzerschnittstellen“ ist ein Oberbegriff für Eingabegeräte, bei denen digitale Informationen als physikalische Objekte dargestellt werden, beispielsweise Touchscreens. Tangible User Interfaces wird oft mit TUI abgekürzt.

A Bilder der Beispielanwendung

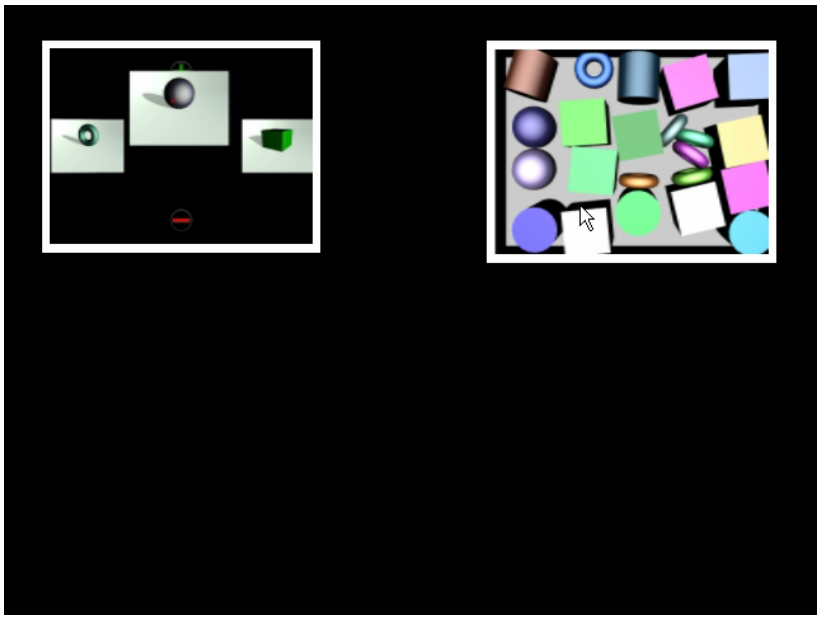


Abbildung 20: Das Auswahlmenü für die Beispielmodelle

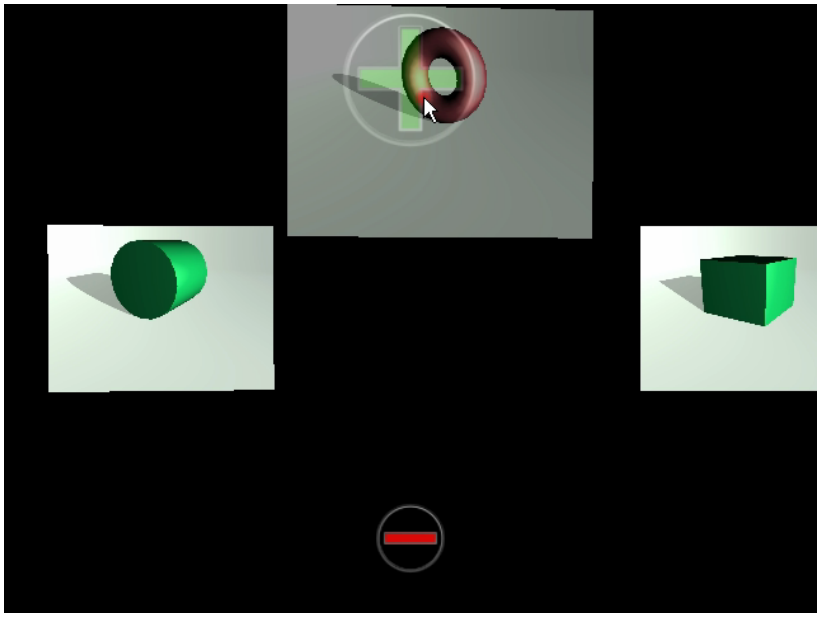


Abbildung 21: Fitnesszuweisung beim Drei-Klassen-Modell

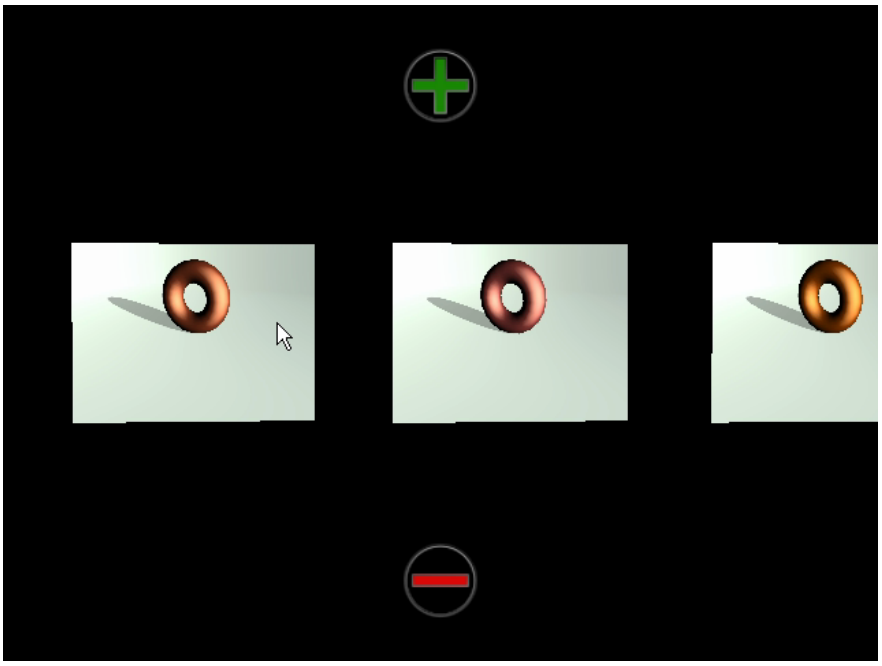


Abbildung 22: Konvergenz in Richtung orangener Tori

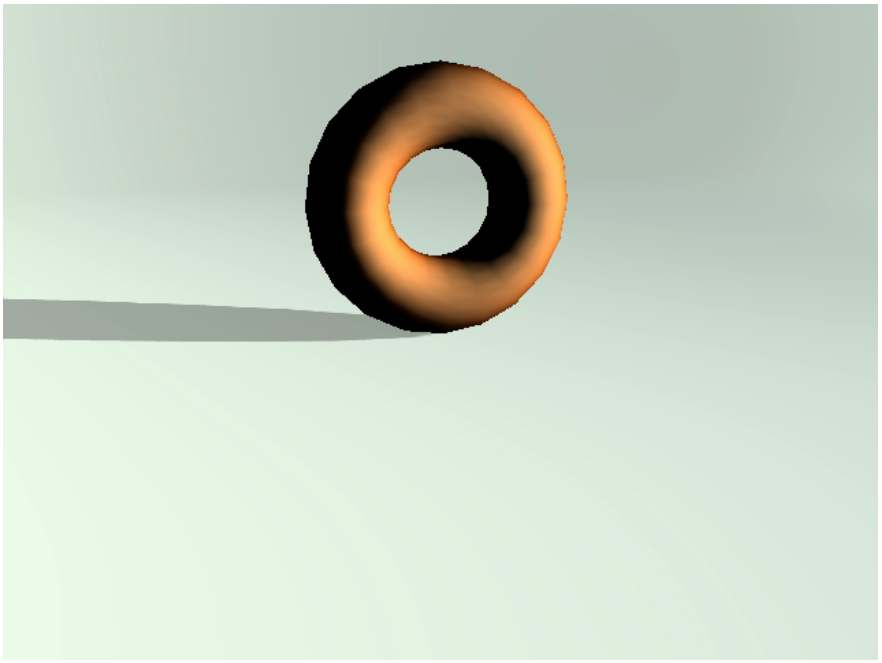


Abbildung 23: Dreidimensionale Ansicht der Individuen

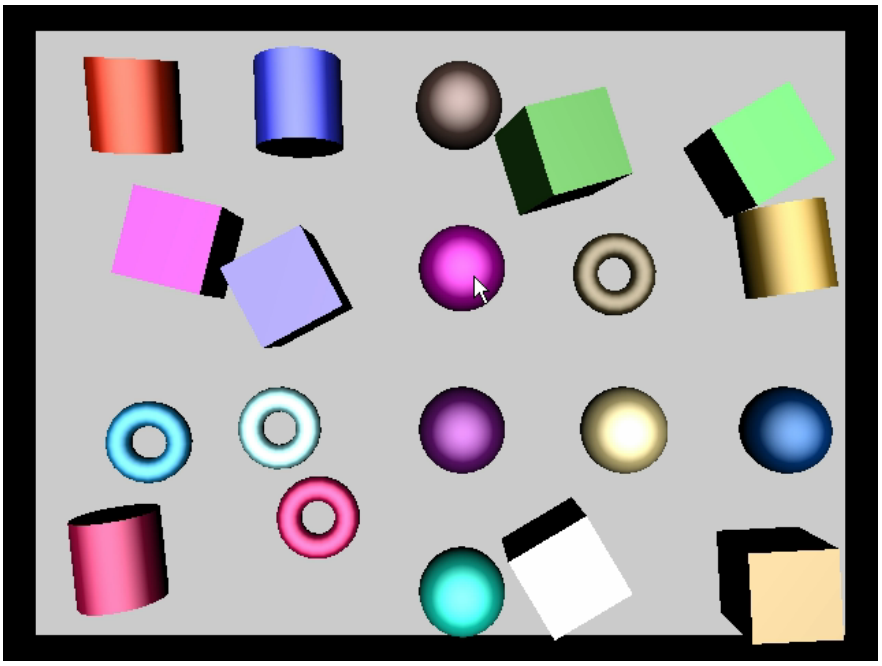


Abbildung 24: Initialpopulation des Billardtisch-Modells

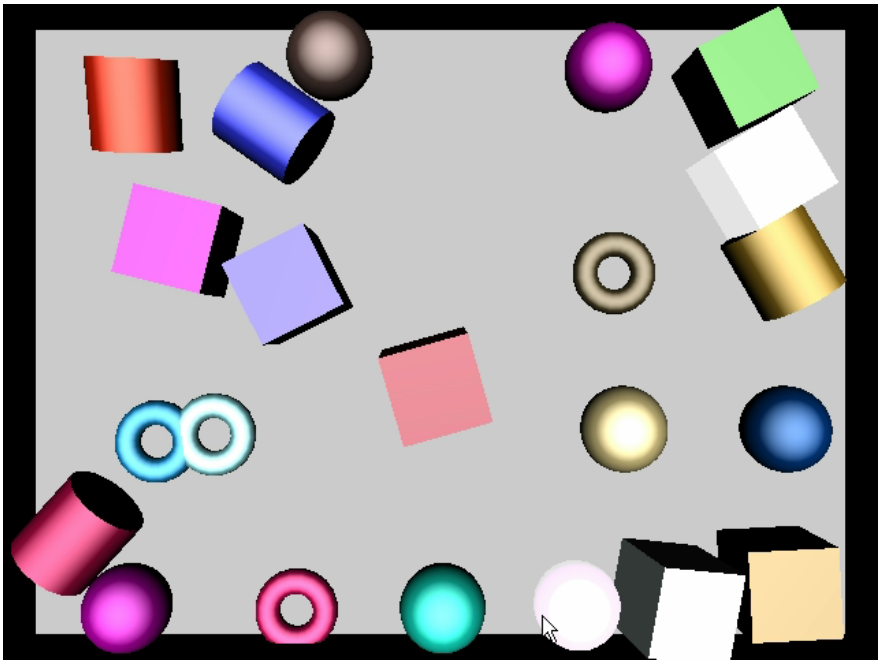


Abbildung 25: Durchgeführtes Crossover zwischen den beiden hell aufleuchtenden Individuen

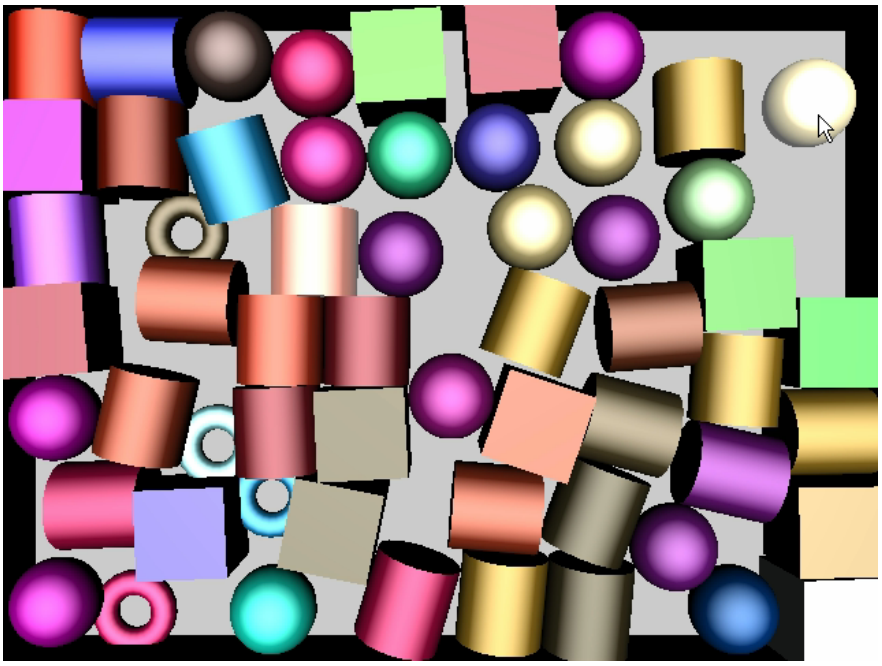


Abbildung 26: Ohne Negativselektion wird der Tisch immer voller