

Berufsakademie Sachsen
Staatliche Studienakademie Leipzig

Optimierung der Suche von „Looping GIFs“ aus beliebigen Videodateien mit einem Evolutionären Algorithmus

Bachelorthesis
zur Erlangung der staatlichen Abschlussbezeichnung eines
BACHELOR OF SCIENCE (B. SC.)
in der Studienrichtung Informatik

Eingereicht von: Leon Kellerhals
Holtzendorffstraße 7
14057 Berlin
CS2012
5000152

Betreuer: Herr Dipl.-Ing. (FH) Clemens Daum
mobilcom-debitel GmbH
Ernst-Reuter-Platz 8
10587 Berlin

Leipzig, den 1.9.2015

Inhaltsverzeichnis

1	Einleitung	1
1.1	Test	1
2	Looping GIFs	2
3	Evolutionäre Algorithmen	3
3.1	Terminologie	5
3.2	Kodierung	5
3.2.1	Binäre Kodierung	6
3.2.2	Kodierung mit Fließkommazahlen	7
3.3	Fitnessfunktion	7
3.4	Selektion	8
3.4.1	Plus-Selektion	8
3.4.2	Komma-Selektion	8
3.4.3	Turnier-Selektion	8
3.4.4	Roulette-Selektion	8
3.5	Mutation	8
3.6	Elitismus	8
3.7	Terminierungskriterien	8
4	Schlussbemerkungen	9
	Quellenverzeichnis	12
	Literatur	12

Kapitel 1

Einleitung

Bla bla bla Einleitung

1.1 Test

Kapitel 2

Looping GIFs

Das Graphics Interchange Format (GIF) ist ein Bildformatstandard, der mit einer Farbpalette und einer verlustfreien Kompression arbeitet. Eine spezielle Eigenschaft dieses Bildformates ist, dass es mehrere Einzelbilder in einer Datei abspeichern kann, die von geeigneten Bildbeobachtungsprogrammen sowie den meisten Webbrowsern als Animationen interpretiert werden. Dabei wird die Animation meist in einer Endlosschleife abgespielt: Wird das Ende erreicht, so wird die Animation ohne Pause wieder von vorne abgespielt. Zwar wurden im GIF gespeicherte Bilder im Internet immer mehr durch das neue PNG-Format abgelöst, erlebte jedoch ein Wiederauferstehen in Bilder-Foren wie *tumblr* [Tumblr:1], *4chan* [4chan:1] oder *Imgur* [Imgur:1].

Eine spezielle Gattung unter den GIF-Animationen in Internetforen stellen die sogenannten „Looping GIFs“ dar. Diese Animationen haben die spezielle Eigenschaft, dass sich ihr erstes und letztes Bild so ähnlich sind, dass der Übergang beim Abspielen der Animation nicht auffällt. Diese Gattung erfreut sich großer Beliebtheit, da die Benutzer der Foren es als eine Herausforderung ansehen, den Schnitt in der Animation zu finden. Meist sind diese Looping GIFs etwa 1 bis 10 Sekunden lange Ausschnitte aus bekannten Videos oder Filmen. Diese Ausschnitte werden von Benutzern in den Foren meistens durch das mehrfache exakte Betrachten des Videomaterials mit dem Auge ermittelt - eine zeitaufwändige und anstrengende Arbeit. Jedoch handelt es sich hierbei um eine Aufgabe, die sehr gut durch einen Algorithmus übernommen werden kann. Ein unter dem Pseudonym „Zulko“ [Zulko:1] bekannter Entwickler hat sich dieser Aufgabe gewidmet und einen iterativen Algorithmus [Goldberg:1] zur Suche von Looping GIFs in Videodateien entworfen [Zulko:2].

Kapitel 3

Evolutionäre Algorithmen

Im Jahre 1859 veröffentlichte Darwin sein Hauptwerk „On the Origin of Species“ („Über die Entstehung der Arten“) [1], in welchem er die Beobachtung formuliert, dass sich alle Lebewesen in langen Zeiträumen verändern und ihrer Umgebung anpassen. Dieser Prozess basiert laut Darwins These auf den Prozessen der *natürlichen Selektion*. Ernst Mayr fasste Darwins Evolutionstheorie folgendermaßen zusammen [2]:

1. Jede Art bringt so viele Nachkommen vor, dass die Population wachsen würde, wenn alle Nachkommen überlebten.
2. Die Populationsgröße einer Spezies ist langfristig konstant.
3. Ressourcen, die die Art für das Überleben benötigt, stehen nur begrenzt, aber über die Zeit in gleichbleibenden Mengen zur Verfügung.
4. Daraus folgt ein Kampf ums Überleben.
5. Die Individuen einer Population unterscheiden sich deutlich voneinander.
6. Die Unterschiede zwischen den Individuen beeinflussen ihre Fähigkeit in ihrer Umwelt zu überleben.
7. Individuen, die weniger gut an ihre Umwelt angepasst sind, haben eine geringere Überlebenschance und weniger Nachkommen. Individuen, die besser an ihre Umwelt angepasst sind, haben eine höhere Überlebenschance und mehr Nachkommen.
8. Die Eigenschaften der Individuen mit einer höheren Überlebenschance verbreiten sich in der Population. Die Eigenschaften der Individuen mit einer geringeren Überlebenschance werden seltener vererbt und fallen damit aus der Population heraus. Dieser Prozess nennt sich natürliche Selektion. Dieser langsam voranschreitende Vorgang führt dazu, dass sich Populationen von Lebewesen sich über lange Zeiteabschnitte an die Umwelt anpassen.

Im folgenden wird Darwins Evolutionstheorie anhand eines Beispiels aus [Rabbits:1] erläutert. Es existiere eine Population von Hasen. Die einzelnen Hasen der Population haben unterschiedliche Eigenschaften: Einige sind schnell, andere sind langsam, manche sind schlau oder dumm. Diese Eigenschaften beeinflussen ihre Fähigkeiten, vor Füchsen

(Man nehme an, dass diese auch existieren) zu fliehen. Dumme und langsame Hasen werden zu einer leichten Beute für Füchsen. Schnelle oder schlaue Hasen sind hingegen oft in der Lage, Füchsen zu entkommen, sei es, weil sie entweder schneller sind oder im richtigen Moment einen Haken schlagen, der ihnen einen großen Vorsprung verleiht. Da sie länger überleben oder (im Falle des schlaugen Hasen) besser Gefahr abschätzen können, sind sie öfter in der Lage, sich mit anderen Hasen zu paaren. Dumme Hasen merken hingegen unter Umständen nicht, dass ein Fuchs auf sie lauert und langsame Hasen entkommen dem Fuchs nicht. Jedoch können auch einige von ihnen überleben, sei es durch Glück.

Der Teil der Population, der überlebt hat, pflanzt sich nun fort. Dabei entsteht eine gute Mischung an Jungen: Langsame Hasen paaren sich mit schlaugen Hasen, schnelle Hasen mit schnellen Hasen, dumme Hasen mit schnellen Hasen, und so weiter. Bei manchen Jungen werden die Eigenschaften, die durch Kombination der Eigenschaften der Eltern entstanden sind, noch mutiert, wodurch ein besonders schlauer, dummer oder schneller Hase erzeugt wird. Die erzeugten Hasenjungen werden durchschnittlich schneller und schlauer sein als die der ursprünglichen Population, da mehr der schlaugen und schnellen Hasen der Elterngeneration überlebt haben und sich paaren konnten. Nach einigen Generationen wird dieser Effekt deutlich: Die gesamte Population der Hasen ist schneller und schlauer als die Hasen der ersten Generation. (Man könnte sich nun Sorgen um die Füchse machen, da die meisten Jagdversuche unerfolgreich sind. Jedoch untergehen Sie den gleichen Evolutionsprozess wie die Hasen und werden ebenfalls schneller und schlauer.)

Evolutionäre Algorithmen (EA) kopieren das Verfahren der natürlichen Evolution, wobei das Verfahren stark vereinfacht wird. Im Gegenteil zu natürlicher Evolution verfolgen EA jedoch ein spezifisches *Ziel*. Das Ziel ist meistens ein zu lösendes Problem. Der EA erzeugt dafür eine *Population* von Lösungskandidaten. Jeder dieser Kandidaten oder *Individuen* enthält die nötige Information um ihren Lösungsversuch für das Problem zu repräsentieren. Diese Information wird in den meisten Fällen kodiert in *Genen* gespeichert. Mit einer vom Benutzer erstellten *Fitness-Funktion* kann für jedes Individuum der *Fitness-Wert* - die Qualität des Individuums für das gegebene Problem - bestimmt werden. Für den Paarungsprozess werden zunächst *Eltern* ausgewählt, die in den Paarungs-Pool aufgenommen werden (dieser Schritt nennt sich *Eltern-Selektion*). Aus dem Paarungs-Pool werden zwei Eltern ausgewählt, um *rekombiniert* zu werden. Bei der *Rekombination* wird ein Kind-Individuum erzeugt, das auch einen Lösungskandidat für das Problem repräsentiert. Die Kinder-Individuen stellen die Individuen der *nächsten Generation* dar. Einige Individuen der neuen Generation werden zufällig ausgewählt, um *mutiert* zu werden. In diesem Schritt werden einige Gene des Individuums zufällig geändert, um eine zusätzliche Vielfalt in der Population zu erreichen. Die meisten evolutionären Algorithmen arbeiten mit einer *festen Populationsgröße*. Daher werden im Schritt der *Umwelt-Selektion* die Individuen mit den besten Fitness-Werten aus der Eltern- und der Kind-Generation ausgewählt. Alle anderen Individuen werden entfernt.

Mit der Zeit haben sich verschiedene Methoden für die Implementierung eines EA ent-

wickelt, um bestimmte Probleme besser lösen zu können. Nur die generelle Struktur des Algorithmus ist meistens die gleiche.

Im folgenden werden die Terminologie von evolutionären Algorithmen sowie die verschiedenen Verfahren in den Abschnitten von EA vorgestellt.

3.1 Terminologie

TODO

3.2 Kodierung

Um ein Optimierungsproblem mit einem EA zu lösen, wird meist eine Übersetzung des Suchraums des Problems benötigt. Diese Übersetzung gleicht dem Genotyp-Phänotyp-Modell aus der Genetik. Der Wert, der im Individuum gespeichert wird, ist der Genotyp. Damit lässt sich mit Hilfe der Kodierung der Phänotyp berechnen. Der Phänotyp stellt einen Lösungskandidaten für das Optimierungsproblem bzw. einen Wert aus dem Suchraum des Optimierungsproblem dar. Es muss also ein Alphabet erstellt werden, das durch den Kodierungsprozess einen Wert im Suchraum des Optimierungsproblems darstellt.

In den frühen zeiten der evolutionären Algorithmen wurde die Genetik der Natur so weit berücksichtigt, dass man

Bei der Wahl der Kodierung sollten einige Grundregeln beachtet werden. Eine davon ist das *Prinzip des minimalen Alphabets* von Goldberg [**Goldberg:1**]:

Der Nutzer sollte das kleinstmögliche Alphabet nutzen, das dem natürlichen Ausdruck des Problems entgegenkommt.

Es soll also weder ein zu großes, noch ein zu kleines Alphabet verwendet werden. Implementiert der Nutzer ein zu großes Alphabet, so kann viel Rechenzeit beim Durchforsten zu kleiner und unwichtiger Teilbereiche des Alphabets verloren werden. Ist das Alphabet hingegen zu klein, so kann es sein, dass das Alphabet bestimmte Bereiche des Problems nicht abdeckt. Das ist insbesondere dann für den EA hinderlich, wenn das zu findende Optimum nicht durch das Alphabet dargestellt werden kann.

Ein weiteres Kriterium für die gute Wahl einer Kodierung ist, dass die Entwicklung von Lösungen möglichst natürlich durch das Alphabet unterstützt wird. Damit wird erreicht, dass nah aneinander liegende Lösungen im Problem auch in ihrer kodierten Form im Individuum Ähnlichkeiten aufweisen. Auch kann dadurch der Rechenaufwand der Kodierung klein gehalten werden.

3.2.1 Binäre Kodierung

In den Anfängen von EA wurden fast ausschließlich binäre Kodierungen verwendet [Gerdes:1].

Bei der binären Kodierung wird der Lösungsraum des Problems durch einen binären String $b = \mathbb{B}^k = 0, 1^k$ repräsentiert. Mit mehr oder weniger Aufwand kann jedes Problem mit einer binären Kodierung repräsentiert werden.

Im folgenden wird beispielhaft ein Problem vorgestellt, das sich sehr leicht mit binärer Kodierung repräsentieren lässt. Man nehme an, man hat einen Raum mit n Lampen $L = \{l_1, l_2, \dots, l_{n-1}, l_n\}$ und n Schaltern $S = \{s_1, s_2, \dots, s_{n-1}, s_n\}$. Jeder Schalter s_i kann an oder aus sein und bedient damit, ob die Lampe l_i an oder aus ist. Ziel ist, die Helligkeit an einer bestimmten Stelle des Raumes durch das An- und Ausschalten der Schalter auf einen gegebenen Wert zu optimieren. Für die Repräsentation des Problems lässt sich offensichtlich eine binäre Kodierung verwenden: Man nehme einen binären String der Länge n . Die i -te 0 (oder 1) stellt dar, dass der Schalter s_i an (oder aus) ist.

Eine weitere Verwendung der binären Kodierung ist die Darstellung vorzeichenloser Integer-Werte. In 3.1 wird dargestellt, wie die Kodierung eines vorzeichenlosen Integers x mit einer Länge von l Bit in einen binären String A erfolgen kann. Werden für das Problem Integer-Werte mit Vorzeichen benötigt, so kann ein weiteres Bit als Vorzeichenbit verwendet werden.

$$A = A_1 A_2 \dots A_{n-1} A_n \in \mathbb{B}$$

$$x = \sum_{i=1}^n A_i \cdot 2^{n-i} \quad (3.1)$$

Für viele Probleme werden mehrere Parameter benötigt. Damit eine Folge von Parametern binär kodiert werden kann, muss für jeden Parameter eine feste Länge bekannt sein. Liegt dies vor, können die einzelnen kodierten Werte im binären String aneinandergereiht werden. 3.2 zeigt auf, wie die Kodierung von m vorzeichenlosen Integer-Werten mit je n Bit erfolgen kann.

$$A = A_1 A_2 \dots A_{m \cdot n-1} A_{m \cdot n} \in \mathbb{B}$$

$$x_k = \sum_{i=1}^n A_{n(k-1)+i} \cdot 2^{n-i} \quad (3.2)$$

Werden die Schalter im Beispiel oben durch Dimmer mit k Stufen ausgetauscht, so lässt sich die in 3.2 dargestellte Kodierung verwenden. m ist dabei die Anzahl der Dimmer bzw. Lampen. Die Länge – also die Anzahl der Bits – der einzelnen vorzeichenlosen Integerwerte n muss so gewählt werden, dass alle Dimmerstufen k dargestellt werden können ($k \leq 2^n$).

3.2.2 Kodierung mit Fließkommazahlen

Man nehme an, die Schalter aus dem Beispiel im Abschnitt 3.2.1 werden durch stufenlose Dimmer ausgetauscht. Die Dimmer können also jeden reellen Zahlenwert zwischen 0 – zugehörige Lampe ist aus – und 1 – zugehörige Lampe leuchtet mit maximaler Helligkeit – annehmen. Werden die Schalter aus dem Beispiel 3.2.1 durch stufenlose Dimmer ausgetauscht,

3.3 Fitnessfunktion

Die Fitnessfunktion oder Zielfunktion bestimmt, wie „fit“ bzw. wie gut ein Individuum ist und beeinflusst damit direkt die Überlebens- und Fortpflanzungschancen der Individuen. Es handelt sich dabei um eine Abbildung, die jedem Element des Suchraums des Problems eine reelle Zahl als Bewertung zuweist. Dabei kann selbst entschieden werden, ob das fitteste Individuum den größten oder den kleinsten Fitnesswert haben soll, also, ob der Wert der Fitnessfunktion maximiert oder minimiert werden soll. Für Selektionsverfahren, die die Selektionswahrscheinlichkeit anhand der Fitness der Individuen bestimmen, stellen negative Fitnesswerte ein Problem dar, da auch die Selektionswahrscheinlichkeit negativ wird. Daher ist es üblich, die Fitnessfunktion so zu konstruieren, dass sie auf alle positive reelle Zahlen abbildet:

$$f : \Omega \rightarrow \mathbb{R} \quad (3.3)$$

Die Fitness eines Individuums ist meistens relativ. Das bedeutet, die Fitness einzelnen Individuums sagt nichts über seine tatsächliche Qualität aus. Mit der Fitness lassen sich lediglich die Individuen unter sich vergleichen.

Man betrachte folgendes beispielhafte Aufgabe: Es soll die Nullstelle der Funktion $g : \mathbb{R}^n \rightarrow \mathbb{R}$ durch einen EA bestimmt werden. Dafür ist es nötig,

Mit der Fitnessfunktion lassen sich viele Probleme beschreiben. Beispielsweise ist es auch möglich, die Nullstelle einer Funktion zu suchen.

Der EA ist nur in der Lage, das globale Minimum oder Maximum einer Funktion zu suchen. Um das gegebene Problem P also zu optimieren, muss eine Funktion definiert werden, die das Problem in ein Minimierungs- oder Maximierungsproblem Die Fitnessfunktion oder Zielfunktion bestimmt wie „fit“ bzw. wie gut ein Individuum ist und beeinflusst damit direkt die Überlebens- und Fortpflanzungschancen der Individuen. Die Fitness eines Individuums wird dabei meist als reelle Zahl angegeben.

Die Fitnessfunktion wird dabei so gewählt, dass das zu optimierende Problem in ein Minimierungs- oder Maximierungsproblem umgewandelt wird. Die Mechanismen des EA sind nur in der Lage, das globale Minimum oder das globale Maximum der Fitnessfunktion zu finden. Die Fitnessfunktion ist also die einzige Verbindung des Algorithmus zu dem Problem.

3.4 Selektion

Stub

3.4.1 Plus-Selektion

Stub

3.4.2 Komma-Selektion

Stub

3.4.3 Turnier-Selektion

Stub

3.4.4 Roulette-Selektion

Stub

3.5 Mutation

Stub

3.6 Elitismus

Stub

3.7 Terminierungskriterien

Stub

Kapitel 4

Schlussbemerkungen

- ☐ **Titelseite:** Länge des Titels (Zeilenumbrüche), Name, Studiengang, Datum.
- ☐ **Erklärung:** vollständig Unterschrift.
- ☐ **Inhaltsverzeichnis:** balanzierte Struktur, Tiefe, Länge der Überschriften.
- ☐ **Kurzfassung/Abstract:** präzise Zusammenfassung, Länge, gleiche Inhalte.
- ☐ **Überschriften:** Länge, Stil, Aussagekraft.
- ☐ **Typographie:** sauberes Schriftbild, keine „manuellen“ Abstände zwischen Absätzen oder Einrückungen, keine überlangen Zeilen, Hervorhebungen, Schriftgröße, Platzierung von Fußnoten.
- ☐ **Punktuation:** Binde- und Gedankenstriche richtig gesetzt, Abstände nach Punkten (vor allem nach Abkürzungen).
- ☐ **Abbildungen:** Qualität der Grafiken und Bilder, Schriftgröße und -typ in Abbildungen, Platzierung von Abbildungen und Tabellen, Captions. Sind *alle* Abbildungen (und Tabellen) im Text referenziert?
- ☐ **Gleichungen/Formeln:** mathem. Elemente auch im Fließtext richtig gesetzt, explizite Gleichungen richtig verwendet, Verwendung von mathem. Symbolen.
- ☐ **Quellenangaben:** Zitate richtig referenziert, Seiten- oder Kapitelangaben.
- ☐ **Literaturverzeichnis:** mehrfach zitierte Quellen nur einmal angeführt, Art der Publikation muss in jedem Fall klar sein, konsistente Einträge, Online-Quellen (URLs) sauber angeführt.
- ☐ **Sonstiges:** ungültige Querverweise (??), Anhang, Papiergröße der PDF-Datei (A4 = 8.27×11.69 Zoll), Druckgröße und -qualität.

Abkürzungsverzeichnis

EA Evolutionärer Algorithmus

Abbildungsverzeichnis

Quellenverzeichnis

Literatur

- [1] Charles Darwin. *On the Origin of Species*. 1859. URL: <http://www.gutenberg.org/etext/15707> (siehe S. 3).
- [2] Ernst Mayr. *The Growth of Biological Thought*. Harvard University Press, 1982 (siehe S. 3).

Selbstständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form weder veröffentlicht, noch einer anderen Prüfungsbehörde vorgelegt.

Leipzig, am 1. September 2015

Leon Kellerhals