

[9]

5. PK Schriftliche Ausarbeitung

Inwiefern sind genetische Algorithmen ein geeignetes Mittel zur Optimierung der Fahrpläne von Buslinien?

Untersuchungen mithilfe einer selbst entwickelten Simulation

Amon Benson & Luca Jungnickel

Inhaltsverzeichnis

1	Gliederung der Präsentation	1
2	Darstellung des Arbeitsprozesses	2
2.1	Begründung der Themenfindung	2
2.2	Einordnung in den fachwissenschaftlichen Zusammenhang	2
2.3	Konkrete Eingrenzung des Themas und Methodik	3
3	Arbeitsplan	7
4	Individuelle Reflexion	9
4.1	Amon	9
4.2	Luca	10
5	Literaturverzeichnis	12

1 Gliederung der Präsentation

- Optimierungsproblem
 - Erklärung unseres Optimierungsproblems (Luca)
 - Erklärung des Aufbaus unserer Simulation (Luca)
- Genetische Algorithmen
 - Genetische Algorithmen im Allgemeinen (Amon)
 - Anwendung des Genetischen Algorithmus auf die Simulation (Amon)
- Ergebnisse
 - Messreihe zur Effizienz des Genetischen Algorithmus (Luca)
 - Beantwortung der Leitfrage (Amon)

2 Darstellung des Arbeitsprozesses

2.1 Begründung der Themenfindung

Da wir uns sehr für Informatik interessieren, sind wir eines Tages auf das Themengebiet der Optimierungsverfahren gestoßen. Hier geht es darum, ein komplexes Problem mittels diverser Algorithmen zu optimieren. Eine mögliche Optimierungsmethode besteht in so genannten genetischen Algorithmen. Hier haben wir uns bei der Themenwahl von einigen interessanten Youtube-Videos inspirieren lassen, welche sich mit dieser Thematik auseinandersetzen. Eines der anschaulichsten Videos behandelt die Optimierung von künstlichen Kreaturen, welche sich mit einfachen Bewegungsfunktionen ausgestattet über eine möglichst weite Strecke bewegen sollten [5]. Dieses Video zeigt die Funktionsweise von genetischen Algorithmen sehr gut, jedoch gibt es nur einen sehr geringen Bezug zur Realität, da hier nur frei erfundene Kreaturen optimiert wurden. Unser Wunsch war es aber, mithilfe von genetischen Algorithmen etwas Realitätsnäheres zu optimieren.

Das Optimieren von Busrouten im Öffentlichen Nahverkehr in Städten war hier unserer Meinung nach gut geeignet, da sich dieses Problem mit nur wenigen Parametern beschreiben lässt. So ist das Simulieren einer Stadt mit Verkehrsaufkommen zwar komplex, jedoch kann eine Buslinie durch sehr wenige Variablen beschrieben werden, nämlich durch die Startzeit der Busse und der angefahrenen Bushaltestellen. Somit ist der Lösungsraum bei diesem komplexen Problem relativ klein, und so hatten wir den Gedanken, der genetische Algorithmus könne so gut eine sinnvolle Lösung finden.

Damit war unser Thema „Genetische Algorithmen zur Optimierung vom Öffentlichen Nahverkehr“ gefunden.

2.2 Einordnung in den fachwissenschaftlichen Zusammenhang

Lebewesen sind vollendete Problemlöser. In der Vielzahl der Aufgaben, die sie bewältigen, übertreffen die die besten Computerprogramme bei weitem - zur besonderen Frustration der Programmierer, die Monate oder gar Jahre harter geistiger Arbeit für einen Algorithmus aufwenden, während Organismen ihre Fähigkeiten durch den scheinbar ziellosen Mechanismus der Evolution erwerben.

Das Zitat von J. H. Holland [4] bezieht sich auf evolutionäre Algorithmen. Durch die zunehmende Leistungsfähigkeit von Rechnern sind diese ein neues Optimierungswerkzeug der Numerik geworden. Sie machen sich das Prinzip der biologischen Evolution zum Vorbild. Kleine Veränderungen von Parametern werden zufällig eingeführt. Durch eine Selektion werden die Veränderungen beibehalten, die zu eine Verbesserung bestimmter

Aufgaben führen. Über viele Schritte ergibt sich so ein verbessertes Verhalten, ohne dass das zugrundeliegende Prinzip notwendigerweise verstanden sein muss. Ein bekanntes Beispiel [3] ist die Optimierung von Flügelformen für Flugzeuge oder ähnliche Probleme in der Aerodynamik.

Erste Ansätze zur evolutionären Optimierung stammen bereits aus den 50er Jahren. [3]

Nach unserer Themenwahl haben wir festgestellt, dass tatsächlich auch schon für Probleme der Optimierung von Buslinien genetische Algorithmen verwendet wurden [10].

2.3 Konkrete Eingrenzung des Themas und Methodik

Um festzustellen, inwiefern genetische Algorithmen Fahrpläne von Buslinien optimieren können, hatten wir die Idee, eine eigene Simulation zu schreiben, welche wir dann mit unserem erstellten genetischen Algorithmus koppeln. Dadurch können wir so präzise wie möglich verstehen, wie ein genetischer Algorithmus funktioniert und sehr genaue Einstellungen und Veränderungen der Simulation vornehmen. Somit ist der Lerneffekt am höchsten.

In unserer Simulation haben wir das Programm *Eclipse* verwendet, mit dem Java-Programmierung durchgeführt werden kann. Diese Programm ist weit verbreitet und kostenlos erhältlich. Wir haben es auf normalen Windows-Notebooks eingerichtet. Die Simulation wurde auf der CPU durchgeführt. Die aktuelle Version unserer Applikation kann auf GitHub als Eclipse-Projekt heruntergeladen werden [2].

Nachdem wir uns ausführlich über die Funktionsweise von genetischen Algorithmen informiert hatten, mussten wir konkrete Einschränkungen und Überlegungen machen, wie komplex unsere Simulation und das gesamte Projekt werden soll. Somit haben wir als Erstes entschieden, wie die Simulation aufgebaut ist und wie der genetische Algorithmus die Simulation verwendet. So haben wir festgelegt, dass die Simulation einer Stadtfläche mithilfe einer gerasterten Karte dargestellt werden kann, wobei jede Kachel quadratisch ist. Eine Kachel kann entweder eine Gebäude oder eine Straße sein (s. Abb. 1).

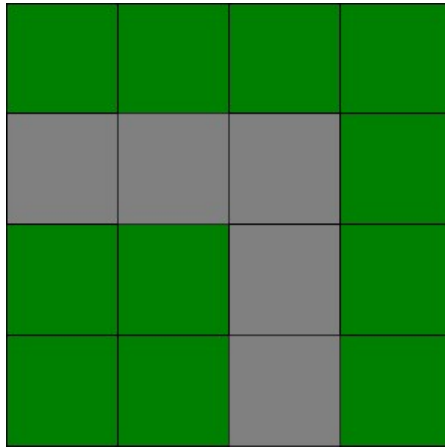


Abbildung 1: Darstellung der Stadtfläche durch Rasterkarte. Eine grüne Kachel repräsentiert ein Haus mit Verkehrsaufkommen, eine dunkelgraue Kachel ist eine Straße

Uns war von Anfang an bewusst, dass dies eine Einschränkung ist, da gewöhnliche Straßennetze nicht notwendigerweise rechteckig aufgebaut sind. Oft ist dies jedoch der Fall und wir mussten diese Einschränkung vornehmen, da es deutlich zu kompliziert gewesen wäre, ein realistisches Verkehrsnetz in die Simulation zu implementieren.

Eine weitere Einschränkung ist die vereinfachte Darstellung des Verkehrs. Wir haben uns dafür entschieden, dass Verkehrssystem so einfach wie möglich zu halten. Konkret ist damit gemeint, dass wir komplett auf Spuren, Ampeln, Einbahnstraßen o.ä. verzichtet haben. Außerdem behindern sich entgegenkommende Fahrzeuge nicht. Der einzige Parameter, welcher den Verkehr beeinflusst, ist die Geschwindigkeit der jeweiligen Straßenkachel. Ein Bus nimmt sofort den Geschwindigkeitswert der befahrenden Straße an. Damit lässt sich die Realität auch sehr gut nachbilden, da für eine Straße die in der Realität durchschnittlich gefahrene Geschwindigkeit als dauerhafter Geschwindigkeitswert der Simulation gelten kann. So können Straßen mit hohem Verkehrsaufkommen nachgebildet werden, indem die Geschwindigkeit der Straße in der Simulation geringer gesetzt wird. Auch sehr kleine Straßen, in denen baulich bedingt sehr langsam gefahren werden muss, können durch einen geringen Geschwindigkeitswert nachgestellt werden. Straßen, auf denen nur wenig Verkehr herrscht, jedoch sehr schnell gefahren werden darf, können durch einen hohen Geschwindigkeitswert in der Simulation eingefügt werden.

Diese Form der Vereinfachung hat jedoch auch eine Grenze. Wenn eine hohe Anzahl an Bussen regelmäßig über eine Straßenkachel fahren (zum Beispiel 10 Busse pro 1

simulierten Minute), dann würde sich in der Realität das Verkehrsaufkommen auf dieser Straße sehr erhöhen und somit würde die Durchschnittsgeschwindigkeit geringer werden. Da in der Simulation die Geschwindigkeit einer Straße aber immer konstant ist, wird nicht betrachtet, inwiefern die fahrenden Busse einen Einfluss auf die Verkehrsdichte haben. Jedoch ist dieses nichtlineare Problem unserer Meinung nach vernachlässigbar, da in der Realität die Verkehrsdichte durch Busse nicht wesentlich erhöht wird. Selten führen Busse zum „Verstopfen“ einer Straße, der Individualverkehr ist hier viel schwerwiegender.

Um die Komplexität unserer Simulation nicht noch weiter zu steigern, mussten wir uns zu Beginn das Problem der Wegfindung von Personen genau überlegen. In der Simulation werden nämlich zufällig Personen erzeugt, welche von einem zufälligen Start zu einem zufälligen Ziel gefahren werden wollen. Dafür muss jede Person ermitteln, mit welchen Buslinien sie fahren und an welchen Bushaltestellen sie umsteigen muss. Da unser Fokus auf die Optimierung der Buslinien mithilfe genetischen Algorithmen lag, haben wir uns entschieden, den Wegfindungsalgorithmus zu vereinfachen. Konkret heißt dies, dass der Wegfindungsalgorithmus als Route immer den von der Strecke her kürzesten Weg ermitteln wird, unabhängig davon, wie oft umgestiegen werden muss und wie sinnvoll dieser Weg eigentlich ist.

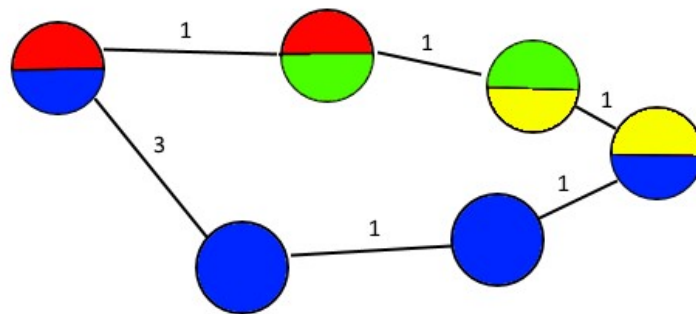


Abbildung 2: Skizze der vereinfachten Wegfindung. Die Kreise repräsentieren Stationen, die Farben verschiedene Buslinien. Ziel ist es, möglichst schnell vom Startpunkt (ganz linker Kreis) zum Zielpunkt (ganz rechter Kreis) zu gelangen. Die jeweilige Zahl zwischen den Kreisen gibt die zurückzulegende Strecke des Busses an, um zur nächsten Station zu fahren.

Die Abbildung 2 veranschaulicht das Problem. Der implementierte Wegfindungsalgorithmus wird als Strecke zwischen dem linken und dem rechten Punkt die obere Route wählen, da die Strecke hier insgesamt 3 beträgt. Der untere Weg hat eine Strecke von 5,

jedoch kann die Person das Ziel mit nur einem Bus erreichen. In der oberen Route muss die Person von der roten zur grünen zur gelben Buslinie umsteigen, in der Realität ist diese Strecke natürlich deutlich schlechter und man würde die blaue Buslinie favorisieren. Da diese genaue Art der Wegfindung jedoch so nicht implementiert wurde, werden in der Simulation teilweise komplizierte Routen gefunden. In einer Erweiterung des Algorithmus könnte eine Favorisierung von Strecken mit wenigen Umstiegen zum Beispiel durch einen Gewichtungsfaktor eingebaut werden.

Der von uns programmierte Algorithmus basiert auf der Quelle [3]. Teilweise konnten Java-Routinen aus [3], [6] und [8]. In diesen Quellen finden sich auch Dokumentationen, auf die hier nicht im Detail eingegangen werden kann.

In Bezug auf den genetischen Algorithmus entschieden wir uns dazu, die Chromosomen einzelner Individuen in Abschnitte zu unterteilen. Das erlaubte uns nun, die Wertebereiche einzelner Abschnitte unabhängig voneinander festzulegen; Der Abschnitt, der die Stationen kodiert, kann so booleanische Werte für die Zustände „Station“ und „keine Station“ enthalten, während die Abschnitte, die die Busabfahrtszeiten festlegen, als Ganzzahlen interpretiert werden. Auch haben wir auf die Implementierung unterschiedlicher Mutations- und Crossover-Methoden [3] verzichtet und verwenden in jeder Generation der Einfachheit halber ein und dieselbe Methode.

Der Hauptteil der Arbeit für die Präsentation war die Programmierung und der Test des Algorithmus. In der Präsentation wird dieser motiviert und erklärt. Zusätzlich wird die Simulation live präsentiert und anschließend diskutiert.

3 Arbeitsplan

Im Folgenden wird der Arbeitsplan in kompakter Form als Tabelle dargestellt.

Datum	Amon	Luca
April 2017	Hier kam zum ersten Mal die Idee auf, genetische Algorithmen als Hauptbestandteil unserer Präsentationsprüfung zu verwenden. Wenig später einigten wir und dann auch schon auf das jetzige Thema: Die Planung von Busrouten mithilfe eines genetischen Algorithmus. Auch ein erstes Treffen mit dem Fachlehrer in Bezug auf die Realisierbarkeit dieses Themas fand im April statt.	
21. April 2017	Erstes Treffen, um die Aufteilung der Programmieraufgaben zu besprechen. Außerdem richteten wir eine Git-Repository ein, mithilfe derer wir unser Projekt gemeinsam bearbeiten und synchronisieren konnten.	
Mai bis August 2017	Graphische Einarbeitung der Simulation. Anzeige der Tile-basierten Stadt, der Events, Personen und aktiven Buslinien.	Programmieren der Simulation. Gegen Ende August funktionierten die Wegfindungsalgorithmen noch nicht, wir verwendeten also fixe Buslinien. Das Transportieren von Personen war allerdings schon möglich.
2. September 2017	Zusammenführen der Ergebnisse.	
20. September 2017	Treffen mit dem betreuenden Fachlehrer, um den Fortschritt unseres Projektes zu besprechen.	
27. September 2017	Treffen mit dem betreuenden Fachlehrer. Besprechen der vorläufigen Gliederung und des Inhalts unseres Vortrags.	
4. November 2017	Besuch der Code-Night am Hasso Plattner Institut. Ziel war es, eine Nacht lang gemeinsam mit anderen Informatik interessierten Schülern an einem Projekt zu arbeiten. Dies half uns auch in Bezug auf unsere Präsentationsprüfung, da man einiges über die Arbeitsteilung im Team und die Nutzung von GitHub [2] für eine gemeinsame Projektverwaltung lernte. In der Code Night hat unser Team den <i>ersten Platz</i> erzielt.	

14. November 2017	Letztes Treffen mit dem betreuenden Fachlehrer vor Abgabe des Exposés.	
7. Januar 2018	Entwicklung eines genetischen Algorithmus in Java, unabhängig von unserer Simulation.	Implementierung der Wegfindungsalgorithmen in die Simulation.
20. Januar 2018	Einarbeiten des genetischen Algorithmus in die Simulation.	Fertigstellung der Simulation, Erstellung des "Blueprint-Converters". Dieser wandelt einen Chromosomensatz (bestehend aus Ganzzahlen) in einen Plan zum Erstellen der Buslinien um. Dieser Plan kann dann in der Stadt simuliert werden.
28. Januar 2018	Anzeigen des genetischen Stammbaums. Alle Individuen mit Verbindungen zu ihren Eltern werden in einem separaten Fenster angezeigt, aktueller Fortschritt des genetischen Algorithmus kann simuliert werden.	Beheben der letzten Fehler in den Wegfindungsalgorithmen der Simulation.
6. Februar 2018	Implementierung einer Fitness-Funktion. Diese wertet eine Simulation aus und gibt dem Individuum seinen Fitness Wert. Außerdem führten wir erste erfolgreiche Tests des gesamten Programms durch.	
11. Februar 2018	Auswertung weiterer Versuchsreihen, erste Tests an einer nachgebildeten Karten von Manhattan.	
15. Februar 2018	Letztes Treffen mit dem betreuenden Fachlehrer vor Abgabe der schriftlichen Ausarbeitung.	

4 Individuelle Reflexion

4.1 Amon

Die Arbeit an unsere Präsentationsprüfung lieferte mir einen umfangreichen Einblick in die Welt der evolutionären Algorithmen.

Schon vor der Themenwahl hatte ich bereits von genetischen Algorithmen gehört, besonders hilfreich waren dort einige Videos auf YouTube [5], die einen sehr leicht verständlichen, aber nur groben Überblick über dieses Gebiet geben.

So fiel die Entscheidung für ein Themenfeld der Präsentationsprüfung auf genetische Algorithmen. Die erste schwierige Frage war dann die Einigung auf ein konkretes Thema. Meine Idee war es, genetische Algorithmen mit Musik zu verknüpfen, d. h. zum Beispiel eine Software zu entwickeln, die automatisch Akkorde oder Melodien generiert. Auch wenn das sicherlich eine nette Spielerei gewesen wäre, fehlt hier doch etwas der Bezug zur realen Welt bzw. ein sinnvoller Anwendungszweck. Luca brachte dann den Vorschlag, die Infrastruktur einer Stadt genetisch generieren zu lassen.

Relativ schnell wurde uns bewusst, dass wir dieses Thema extrem eingrenzen mussten. Somit begrenzten wir die Simulation auf Buslinien. Danach überlegten wir uns, wie eine Simulation aufgebaut sein müsste und entwarfen Pläne für das Busroutensystem und die Wegfindung der Personen. Die Aufteilung und Zusammenarbeit gelang vollkommen problemlos. Luca beschäftigte sich zunächst hauptsächlich mit der Programmierung der Simulation, während ich mich um deren graphische Implementierung und später um die Programmierung der genetischen Algorithmen kümmerte. Dieses Prinzip der Trennung von „Core-Anwendung“ und Grafik stammt aus unserer Quelle über objektorientierte Programmierung von Balzert [1], welche mir auch unabhängig von unserer Präsentationsprüfung sehr bei der strukturierten und organisierten Programmierung half.

Ich besorgte mir Literatur zur Programmierung von genetischen Algorithmen [8, 6] und programmierte einige Beispiele aus [6] nach, bis ich dann Anfang Januar 2018 begann, den genetischen Algorithmus für unser Projekt zu erstellen.

Es beruhigt mich auf jeden Fall, dass unser Projekt letzten Endes an kleineren Städten sehr erfolgreiche Ergebnisse liefert. Es ist jedoch etwas Schade, dass wir am Schluss nur wenig Zeit zum „Experimentieren“ hatten. Auch musste stellenweise mehr Wert auf Schnelligkeit zur Erreichung eines konkreten Ergebnisses als auf höchste Sorgfalt (Kommetierung, Eleganz der Umsetzung) gelegt werden.

4.2 Luca

Die Arbeitsteilung zwischen Amon und mir hat hervorragend funktioniert. Hier hatten wir bereits Erfahrung, da wir gemeinsam die MSA Präsentation gemacht haben und schon dort die Zusammenarbeit sehr gut funktioniert hat.

Zu Beginn haben Amon und ich das Thema sinnvoll eingeschränkt. So hatten wir zuerst die Idee, eine ganze Infrastruktur in einer Stadt zu optimieren, also zum Beispiel auch zu optimieren, wo Straßen gebaut werden, Siedlungen geschaffen werden oder ähnliches. Schnell haben wir jedoch gemerkt, dass dies zu komplex wird und wir uns auf eine einfachere Optimierung beschränken sollten. So haben wir uns schließlich geeinigt, "nur" die Buslinien in einer Stadt mithilfe einer selbstentwickelten Simulation zu optimieren.

In der Simulation habe ich zunächst die „Core“-Anwendung programmiert, also die rechnerische Grundlage ohne eine grafische Darstellung. Amon hat sich zuerst auf die Programmierung der Grafik und später auf die Programmierung des genetischen Algorithmus fokussiert. Dabei mussten viele Entscheidungen, welche die Funktion der Simulation betreffen, gefällt werden. So habe ich mich unter anderem entschieden, die Wegfindungsalgorithmen in der Simulation zu vereinfachen, da der Fokus unserer Arbeit die Optimierung mithilfe genetischer Algorithmen ist, und nicht die der Wegfindungsalgorithmen (siehe Abschnitt 2.3). Abschließend betrachtet war das eine sehr gute Entscheidung, da der Auswirkungen der Vereinfachung verkraftbar sind und wir uns so auf die gründliche Implementierung des genetischen Algorithmus konzentrieren konnten.

Die Quellenwahl sowie Recherche ist mir sehr leicht gefallen, so hatte ich bereits Erfahrung in der Einbindung von sog. Third-party-libraries, also der Verwendung von fremden Bibliotheken in der Programmierung. Dadurch konnte ich mich auf die eigentliche Programmierung konzentrieren. Außerdem gibt es zu der Thematik der genetischen Algorithmen sehr gute und ausführliche Literatur, welche zwar eigentlich auf Universitätsniveau ist, jedoch in den meisten Punkten auch verständlich ist. Ein Problem für mich war, die beschriebene Mathematik in meiner Hauptquelle [3] zu verstehen. Diese ist auf Universitätsniveau und so sehr verschieden zu der Art von Mathematik, welche man in der Schule kennenlernt. Nichtsdestotrotz konnte mir diese Quelle einen sehr tiefen Einblick in dieses Thema geben und war so für mich Grundlage in der Entwicklung des genetischen Algorithmus.

Abschließend lässt sich die Planung und Umsetzung des Projektes als sehr erfolgreich betrachten. So haben wir es geschafft, dieses komplexe Projekt umzusetzen und ein

funktionierendes Ergebnis vorzeigen zu können. Zusätzlich haben wir ein unserer Meinung interessantes Thema ausgesucht, an dem wir Spaß hatten, zu arbeiten. Dadurch haben wir uns intensiv mit dem Themengebiet auseinandergesetzt und sehr viel gelernt. Auch für die Zukunft hat mir dieses Thema einiges gebracht, da ich voraussichtlich technische Informatik studieren möchte und es so sehr sinnvoll ist, bereits einen Einblick in ein Themengebiet der Informatik zu erhalten. Ich habe ausführlich gelernt, wie genetische Algorithmen funktionieren und wie man diese mit einfachen Methoden effektiv gestalten kann, wie große Softwareprojekte zu strukturieren sind und wie man gemeinsam an einer Software arbeitet und dies koordiniert. Auch wenn letzteres nicht explizit etwas mit unserer Fragestellung zu tun hat, ist auch dieses Wissen in der Informatik sehr wichtig.

5 Literaturverzeichnis

- [1] Balzert, Helmut. *Java: Objektorientiert programmieren: Vom objektorientierten Analysemodell bis zum objektorientierten Programm*. Dortmund: W3L AG, 2014. (Quelle von Luca)

Dieses Buch war für den Erfolg unseres Projektes essentiell. Durch bestimmte objektorientierte Designmethoden konnten wir das Projekt erst sinnvoll strukturieren und uns so schließlich auf die Entwicklung der Simulation und der Implementierung der genetischen Algorithmen fokussieren. Auch wenn dieses Buch auf den ersten Blick nichts mit unserem Themengebiet zu tun hat, haben wir auf bestimmte Designmethoden zurückgegriffen und waren nur so in der Lage, die Simulation und den genetischen Algorithmus zu entwickeln.

- [2] Benson, Amon und Jungnickel, Luca. *Generic Traffic Simulator*. 2017. URL: <https://github.com/amonbenson/GenericTrafficSimulator>. [abgerufen am 11.03.2018] (Quelle von beiden)

Link zur Github-Repository unseres Projektes.

- [3] Gerdes, Ingrid, Klawonn, Frank und Kruse, Rudolf. *Evolutionäre Algorithmen: Genetische Algorithmen - Strategien Und Optimierungsverfahren - Beispielanwendungen*. Wiesbaden: Friedr. Vieweg Sohn Verlag, 2004. (Quelle von Luca)

Diese Quelle konnte mir einen sehr tiefen und gründlichen Einblick in die genetischen Algorithmen geben. Ein Problem dieses Buches war, dass die Mathematik auf Universitätsniveau beschrieben und so teilweise schwierig nachzuvollziehen war. Jedoch war dieses Buch meine wichtigste Quelle zum Verständnis von genetischen Algorithmen. Zusätzlich konnten viele vorgestellte Designmethoden implementiert beziehungsweise abgewandelt eingesetzt werden.

- [4] Holland, J. H. "Genetic Algorithms". In: *Scientific American* (1992). (Quelle von Amon)

- [5] Huang, Cary K. *Evolution Simulator (Part 1/4)*. 2015. URL: https://www.youtube.com/watch?v=G0Fws_hhZs8. [abgerufen am 08.03.2018] (Quelle von Amon)

In diesem Video erfuhr ich erstmalig über die Existenz und den grundlegenden Aufbau von genetischen Algorithmen. Da die Funktionsweise hier zwar relativ simpel, aber sehr anschaulich präsentiert wird, ist sie es definitiv wert, im Literaturverzeichnis erwähnt zu werden.

- [6] Jacobson, Lee und Kanber, Burak. *Genetic Algorithms in Java Basics*. New York: Apress, 2015. (Quelle von Amon)

Dieses Buch bietet zwei große Vorteile; Erstens wird die Idee eines genetischen Algorithmus, die in [3] sehr ausführlich beschrieben wird, sinnvoll und verkürzt zusammengefasst. Weiterhin enthält diese Quelle Code-Beispiele verschiedener genetischer Algorithmen in Java. Damit bot das Buch eine hervorragende Grundlage für die Implementierung unseres eigenen genetischen Algorithmus.

- [7] Patel, Amit. *Introduction to A**. 2014. URL: <https://www.redblobgames.com/pathfinding/a-star/introduction.html>. [abgerufen am 03.01.2018] (Quelle von Luca)

Diese Onlinequelle hat mir für die Entwicklung der Simulation sehr geholfen. So musste ich einen Wegfindungsalgorithmus implementieren, damit Personen die richtige Buslinie zum Ziel verwenden und auch umsteigen können. Die Quelle hat mir einen gründlichen Einblick in die Funktionsweise verschiedener Wegfindungsalgorithmen gegeben. Schließlich habe ich den A*-Algorithmus mithilfe der third-party-library „jGraph“ implementiert.

- [8] Selzam, Bianca. *Genetische Algorithmen*. Techn. Ber. TU Dortmund, 2006. (Quelle von Amon)

Diese Quelle verwendete ich, um mir einen ersten grundlegenden Überblick über

die Bestandteile eines genetischen Algorithmus zu verschaffen. Sie beschreibt die Bezüge von genetischen Algorithmen zur biologischen Evolution.

- [9] *Titelbild.* URL: `darwinian-medicine.com/wp-content/uploads/2016/11/evolution.jpg`. (Quelle von Amon)

- [10] Yu, Bin u. a. “Parallel genetic algorithm in bus route headway optimization”. In: *Applied Soft Computing, Volume 11, Issue 8* (2011). (Quelle von Luca)