Optimierung einer simulierten Verkehrssituation mittels evolutionärer Algorithmen

Eine Projektarbeit im Fach Bionik von Philipp Meißner

Traffic Light tree: Lichtskulptur von Pierre Vivant in London

Inhalt

Ziel dieser Arbeit 2

Definition des Optimums und Anforderungen an die Simulation 4

Implementierung der Simulation 5

Implementierung des evolutionären Algorithmus 7

Abschließende Bewertung 10

Quellen 10

Abbildungsverzeichnis 10

Ziel dieser Arbeit

Die Situation erlebt praktisch jeder regelmäßig. Man quält sich durch den nachmittäglichen Innenstadtverkehr und muss an beinahe jeder Ampel halten. Es drängt sich das Gefühl auf, dass sie alle nur darauf programmiert wurden, einen persönlich aufzuhalten. Schaut man sich um, wird man meist eines Besseren belehrt, denn den anderen Verkehrsteilnehmern geht es selten besser. Noch schlimmer empfindet man es jedoch, wenn gar keine anderen Fahrzeuge zu sehen sind und man einsam an einer roten Ampel steht, die sich beharrlich weigert, grün zu werden. Man kommt dann leicht ins Grübeln, ob das denn wirklich alles nötig ist oder ob man die Ampeln nicht etwas besser schalten könnte, so dass der Verkehr insgesamt besser fließt.

Die besten Lösungen dafür lassen sich zweifellos mit intelligenten Ampeln erzielen, die Fahrzeuge erkennen und sich somit stets an die aktuelle Verkehrssituation anpassen können. Diese benötigen jedoch eigene Sensoren – meist Induktionsschleifen unter dem Fahrbahnbelag – deren Nachrüstung bei bestehenden Ampelanlagen häufig zu teuer ist. Besonders dem einzigen Fahrzeug an einer ansonsten leeren Kreuzung können solche Systeme im Idealfall schon beim Heranfahren den Weg freigeben. Im dichten Stadtverkehr, wenn praktisch alle Straßen stets gefüllt sind, helfen Sensoren allein jedoch auch nicht weiter und es kommt, wie auch bei den sensorlosen Ampeln, darauf an, dass die Schaltzeiten der Ampeln gut gewählt und vor allem gut aufeinander abgestimmt sind. Wenn man nun als frustrierter Autofahrer zu der Erkenntnis kommt, dass man die Ampeln, die einem die Fahrt verderben, doch sicher besser einstellen könnte, so hat man damit zwar vermutlich Recht, muss aber bei genauerer Analyse des Problems feststellen, dass das Umsetzen dieser Erkenntnis alles andere als trivial ist. Schon bei einem kleinen Straßennetz mit vielleicht zehn Ampeln hat man es mit einem System mit einer Vielzahl voneinander unabhängiger Parameter zu tun. Die Anzahl der Möglichkeiten ist astronomisch und durch logisches Überlegen kann man zwar offensichtliche Fehler vermeiden, jedoch nicht die optimale Lösung finden. Schon die Annahme einer optimalen Lösung ist ein Trugschluss, denn der Verkehr ändert sich ständig und eine Einstellung, die für eine Situation passend ist, kann in einer anderen ungeeignet sein. Dem kann man zwar entgegenwirken, indem man regelmäßige Tendenzen des Verkehrs mit einbezieht und die Einstellung über den Verlauf eines Tages variiert, grundsätzlich lösen lässt sich die Problematik dennoch nicht. Wer Ampelschaltungen optimieren will, hat dieselben Probleme wie Leute, die Fahrpläne entwerfen. Man kann es niemals allen recht machen, und was eine Anforderung erfüllt, kann einer anderen diametral entgegen stehen.

In der Praxis trifft man häufig auf derartige Aufgaben. Sie alle haben gemeinsam, dass man sie auf analytischem Wege nur sehr schwer oder sogar überhaupt nicht lösen kann. Häufig kommen dann Erfahrungswerte zum Einsatz, die sich in der Vergangenheit bewährt haben und die man durch stetiges Variieren einzelner Parameter und der Beobachtung der Auswirkungen erlangt hat. Sowohl das Problem als auch der Lösungsweg erinnern dabei an die Herausforderungen, mit denen Populationen von Lebewesen ständig konfrontiert sind und denen sie, wenn auch für das Individuum unbewusst, durch die natürliche Evolution begegnen. Die Umwelt stellt an Lebewesen eine Vielzahl von Anforderungen, die sich oft gegenseitig widersprechen und zu deren Erfüllung es zwar viele gute, aber keinen optimalen Weg gibt. Ein Ökosystem wird meist von einer Vielzahl von Lebewesen bewohnt, die alle individuelle Möglichkeiten gefunden haben, den Anforderungen, die das Ökosystem an sie stellt, zu entsprechen. Natürlich ist diese Formulierung etwas unwissenschaftlich, denn die Lebewesen suchen nicht aktiv nach Eigenschaften, die ihnen im täglichen Überlebenskampf hilfreich sind, sondern sie bilden sie über unzählige Generationen heraus, da sie alle den Gesetzen der Evolution unterliegen. Diese lassen sich im Groben auf zwei Schlagwörter reduzieren: Selektion und Mutation. Zufällige Veränderungen im Erbgut einzelner Individuen verschaffen diesen Eigenschaften, die sich von denen ihrer Artgenossen unterscheiden. Stellen diese neuen Eigenschaften im Hinblick auf die Anforderungen aus der Umwelt eine Verbesserung dar, so erhöht dies die Wahrscheinlichkeit, dass sie an nachfolgende Generationen weitergegeben werden. Diese Prinzipien führen zu einem kontinuierlichen Veränderungsprozess der dafür sorgt, dass sich eine Art immer besser an ihre Umweltbedingungen anpasst. Es gibt dabei einige zusätzliche Faktoren, die den Prozess beeinflussen und seine Effizienz steigern. So sorgt zum Beispiel die Sexualität zum einen für eine ständige Durchmischung des Erbguts und somit für eine Erhöhung der Variationsrate und zum anderen schafft sie einen zusätzlichen Selektionsfaktor, durch den vorteilhafte Eigenschaften vermehrt verbreitet werden. Eine umfassende Beschreibung aller bekannten Mechanismen der Evolution würde den Rahmen dieser Arbeit sprengen, jedoch werden im weiteren Verlauf noch einige praktische Beispiele folgen, die einzelne Aspekte veranschaulichen.

Die Parallelen zwischen den zuvor beschriebenen technischen Problemstellungen und den Herausforderungen des Lebens sind so groß, dass sich die Idee quasi aufdrängt, für die Lösung der Probleme die Mechanismen der Evolution zu nutzen. Dies soll in dieser Arbeit anhand der anfangs beschriebenen Ampelproblematik durchgeführt werden. Dafür wird ein Simulationsprogramm geschrieben, das eine beispielhafte Verkehrssituation mit einigen Kreuzungen und Ampeln widerspiegeln und bewerten soll. Anschließend sollen evolutionäre Algorithmen – also Algorithmen, die die Prinzipien der Evolution anwenden – dafür benutzt werden, um die Ampelschaltungen zu optimieren und den Verkehrsfluss zu verbessern. Das Programm soll in C++ unter Nutzung der Klassenbibliothek Qt geschrieben werden. Qt dient dabei zur Realisierung einer grafischen Benutzeroberfläche (GUI) und stellt zudem etliche Funktionen zur einfachen Implementierung effizienter Softwarearchitektur bereit. Da jedoch der bionische Aspekt der Arbeit im Fokus stehen soll, wird auf programmiertechnische Probleme nur am Rande eingegangen.

Definition des Optimums und Anforderungen an die Simulation

Die Evolution ist ein stetiger Prozess, der eine kontinuierliche Optimierung der Eigenschaften der Lebewesen zur Folge hat. Jedoch ist das Optimierungsziel dabei nicht immer offensichtlich und besteht meist in einer Erfüllung zahlreicher Anforderungen. Schnelligkeit, Stärke und gute Flugeigenschaften können dabei ebenso wichtig sein wie Energieeffizienz oder gute Tarnung. Meist bestimmen dabei die jeweiligen Umweltbedingungen die Prioritäten der einzelnen Faktoren. Da die Evolution jedoch kein intelligent gesteuerter Prozess, sondern lediglich das Ergebnis des Zusammenspiels verschiedener Wirkmechanismen ist, führt sie auch gelegentlich zu Ergebnissen, die einem logisch denkenden Beobachter sehr unvernünftig vorkommen. Der Evolutionsbiologe Richard Dawkins zeigt dies in seinem 1986 erschienenen Buch „Der blinde Uhrmacher“ sehr anschaulich am Beispiel des Hahnenschweif-Widah. Die Männchen dieses in Afrika lebenden, etwa spatzengroßen Vogels bilden in der Paarungszeit Schwanzfedern aus, die eine Länge von mehr als 45 cm erreichen können. Dass diese Federn den Vogel erheblich behindern, seine Flugfähigkeit beeinträchtigen und ihn somit zur leichten Beute für Raubtiere machen, ist offensichtlich. Dawkins erklärt das Phänomen mit der Wirkungsweise der sexuellen Selektion. Da die Weibchen dieser Gattung Männchen mit langen Schwanzfedern bevorzugen, hat sich dieses Merkmal im Laufe der Zeit so extrem ausgebildet. Eine positive Rückkopplung ist dabei ausschlaggebend, denn die besonders langschwänzigen Männchen tragen nicht nur die Gene für das Ausbilden eines langen Schwanzes in sich, welche sie an ihre männlichen Nachkommen weitergeben, sie tragen zugleich auch das Erbgut, das ihre weiblichen Nachkommen dazu bringt, eben diese langschwänzigen Männchen zu bevorzugen. Diese Gene haben sie von ihren Müttern bekommen, die ja ebenfalls langschwänzige Männchen bevorzugt haben müssen, denn sonst hätten sie sich nicht mit dem vermutlich langschwänzigen Vater gepaart. Da der Selektionsdruck durch Beutetiere mit steigender Schwanzlänge wächst, stellt sich irgendwann ein Gleichgewicht zwischen diesen beiden konträren Anforderungen ein. Dieses Gleichgewicht, man könnte auch sagen Optimum, liegt jedoch weit ab von dem, was ein Beobachter, der diesen Mechanismus nicht kennt, als sinnvoll erachten würde. Das Beispiel zeigt zum einen, wie der Prozess der Evolution manchmal einen einmal eingeschlagenen Holzweg bis zum Ende beschreitet und es macht zum anderen klar, dass bloßes Kopieren von der Natur nicht immer klug ist. Der Bioniker, der technische Probleme mit Lösungen aus der Natur angehen will, sollte also stets darauf achten, dass das Vorbild, das er sich sucht, auch tatsächlich auf die Erfüllung der gewünschten Anforderungen optimiert ist.

Abbildung 1: männlicher Hahnenschweif-Widah

Bei der Verkehrssimulation können wir die Anforderungen selbst festlegen. Das Beispiel macht jedoch deutlich, dass eine simulierte Evolution nur dann das gewünschte Ergebnis liefern kann, wenn dieses anhand der verwendeten Algorithmen klar definiert ist. Dies bedeutet, dass die Verkehrssimulation nur dann praxisrelevante Ergebnisse liefern wird, wenn sie ihrem realen Vorbild

möglichst gut entspricht. Dennoch ist natürlich klar, dass es nahezu unmöglich ist, alle Aspekte des Straßenverkehrs abzubilden, und dass man zwangsläufig starke Vereinfachungen vornehmen muss. Diese Reduzierung auf das Wesentliche ist das Wichtigste bei einer Simulation.

Implementierung der Simulation

Die Simulation nutzt die Fähigkeiten der Qt Klassenbibliothek um ein grafisches Abbild des simulierten Verkehrs zu erzeugen. Natürlich wäre auch eine rein numerische Simulation möglich, bei der alle Daten stets nur gesichtslose Zahlen im Arbeitsspeicher des Computers wären. Diese wäre sogar um ein Vielfaches effizienter und könnte schneller berechnet werden, jedoch wäre sie für Menschen nur schwer erfassbar und daher kaum zu bewerten. Denn letztendlich ist es natürlich der Mensch, der entscheiden muss, wie sehr das Abbild im Computer dem ihm bekannten realen Verkehr ähnelt. Kann er dies nicht, kann er die Simulation nicht nur nicht auswerten, er kann sie auch gar nicht erst erschaffen. Dies ist auch einer der Kerngedanken der objektorientierten Programmierung, die versucht, real existierende Strukturen wenn möglich in der Struktur des Programmes abzubilden, das sich mit ihnen beschäftigen soll. Hinzu kommt, dass das diese Arbeit den Einsatz evolutionärer Algorithmen ja lediglich demonstrieren soll, und dies kann sie mit Hilfe eines grafischen Abbildes wesentlich anschaulicher.

Es wurden zunächst Objekte erschaffen, die einzelne Kreuzungen repräsentieren, und diese wurden anschließend zu einem Straßennetz verbunden. Dabei wurden die Kreuzungen selbst so programmiert, dass sie angrenzende Kreuzungen suchen und die entsprechenden Verbindungen zeichnen. Um den Aufwand zu reduzieren, wurden ausschließlich rechtwinklig verlaufende Straßen realisiert. Die zunächst grauen Balken und Quadrate wurden dann noch durch das Zeichnen von weißen Mittelstreifen und Haltelinien etwas aufgehübscht. Anschließend wurden die Kreuzungen dahingehend erweitert, dass sie Ampeln erzeugten, wenn auf ihnen mehr als zwei Straßen zusammentrafen. Die Ampeln können dabei die Kreuzung entweder für den Verkehr auf der nördlichen und der südlichen Straße oder für den auf der westlichen und der östlichen freigeben. Zusätzliche Verfeinerungen, wie eine gesonderte Behandlung des Abbiegerverkehrs, wurden bewusst unterlassen. Beim Wechsel schalten die Ampeln zunächst das Leuchtzeichen der freien Straßen über Gelb zu Rot und anschließend das Signal der vorher gesperrten Richtung von Rot über Gelb zu Grün. Es gibt also eine Phase, in der für alle Richtungen ein rotes Signal ausgegeben wird. Diese dauert, ebenso wie die beiden Gelbphasen, in der Simulation 0,5 Sekunden. Die Dauer der beiden Grünphasen ist variabel und soll später durch den evolutionären Algorithmus optimiert werden, ebenso wie ein möglicher zeitlicher Versatz des gesamten Zyklus. Als dieser noch nicht programmiert war, schalteten alle Ampeln stets synchron entweder die Nord-Süd- oder die Ost-West-Richtung frei, wodurch Effekte wie eine grüne Welle nicht möglich waren. Zum Schluss wurden die Autos programmiert, die sich auf dem Straßennetz bewegen sollten. Sie werden durch türkise Kreise symbolisiert und fahren stets eine ihnen fest vorgeschriebene Strecke ab. Treffen sie dabei auf eine rote Ampel, so halten sie an und warten, bis diese grün wird. Müssen mehrere Autos an einer Ampel warten, so bilden sie eine Warteschlange. Um die Berechnung der Simulation zu beschleunigen, wurde jedoch vermieden, dass die Autos dafür miteinander kommunizieren müssen. Sie fragen stattdessen die Ampel, wie viele andere Autos bereits an ihr warten, und berechnen damit den Platz, an dem sie halten müssen. Die Ampel weiß daher immer, wie viele Fahrzeuge gerade an ihr warten, und zeigt dies auch durch eine kleine rote Zahl im Simulationsfenster an. Beschleunigungs- und

Bremszeiten der Fahrzeuge wurden ebenso vernachlässigt wie unterschiedliche Geschwindigkeiten. Wenn der Weg frei ist, bewegen sie sich mit konstanter Geschwindigkeit, ansonsten stehen sie.

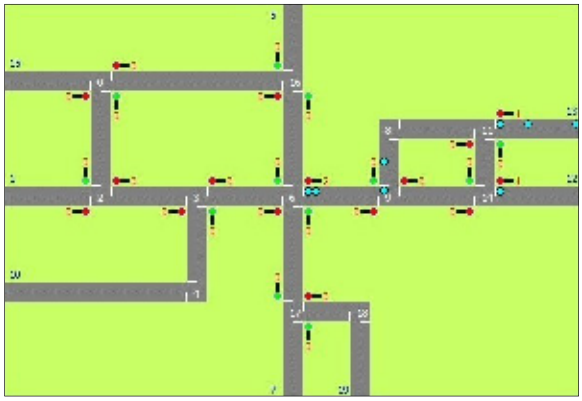

Abbildung 2: Straßennetz

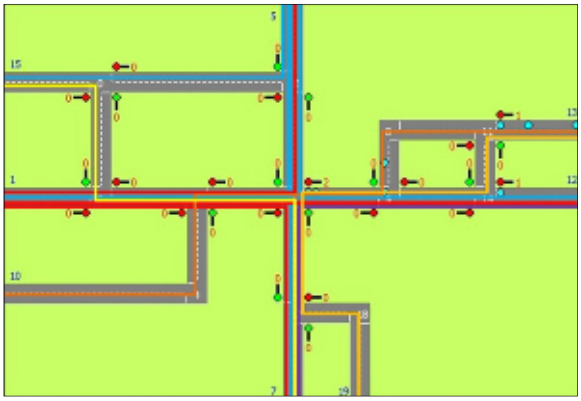

Abbildung 3: Routen

Es wurden zehn verschiedene Routen durch das Straßennetz definiert, die jeweils in beide Richtungen befahren werden. Die Autos werden von unsichtbaren Objekten in zufälligen Abständen erzeugt und auf eine der Routen festgelegt. Die Häufigkeit, mit der neue Autos ihre Fahrt antreten, hängt dabei von den jeweiligen Routen ab. Natürlich soll dies bei Fahrten über die Hauptverkehrsstraßen häufiger passieren als bei anderen. Der zeitliche Aspekt wird dabei von einem globalen Zeitgeberobjekt gesteuert, das auch die Geschwindigkeit der Autos und das Umschalten der Ampeln festlegt. Die Taktrate dieses Zeitgebers kann verändert werden, so dass die Simulation mit bis zu zehnfacher Geschwindigkeit abgespielt werden kann. Bei Versuchen, die Geschwindigkeit wieder zu steigern, zeigte sich, dass der Computer dabei so stark belastet wurde, dass das Programm instabil wurde.

Zuletzt musste das Optimierungsziel festgelegt werden. Es musste ein Maß gefunden werden, anhand dessen der Verkehrsfluss beurteilt werden kann und welches der Algorithmus positiv beeinflussen soll. Dieses Maß sollte die durchschnittliche Frustration der Verkehrsteilnehmer sein, die sich innerhalb der Simulation bewegen. Es wurde festgelegt, dass jedes Fahrzeug mit einem Frustrationswert von 0 startet und sich dieser Wert im Verlauf der Fahrt entweder nach oben oder nach unten verändern kann. Erreicht ein Fahrzeug sein Ziel, übermittelt es seinen Frustrationswert an eine Auswerteeinheit, die den durchschnittlichen Frustrationswert aller Fahrzeuge berechnet. Der Zusammenhang zwischen der Veränderung dieses Kennwertes mit den Situationen, mit denen das Fahrzeug konfrontiert wird, ist dabei essentiell, denn er bestimmt, wie gut oder schlecht eine Verkehrssituation bewertet wird. Man sollte daher nicht einfach nur für jede Ampel, an der das Fahrzeug halten muss, den Frustrationswert um einen festen Betrag erhöhen. Würde man dies tun, so hätten Fahrzeuge, die längere Strecken fahren und somit auch mehr Ampeln passieren müssen, statistisch immer einen höheren Frustrationswert als solche, die nur kurze Strecken fahren. Es ist klar, dass die Frustration sinken muss, wenn das Fahrzeug an einer grünen Ampel vorbeifährt, ohne warten zu müssen. Dies deckt sich auch mit unserer eigenen Erfahrung. Schließlich muss noch erwähnt werden, dass die Frustration, die beim Warten an einer Ampel entsteht, nicht linear mit der Wartezeit steigen darf, denn das könnte dazu führen, dass wenig befahrene Strecken nahezu komplett gesperrt werden, um viel befahrene Strecken dauerhaft freizugeben. Zwar ist eine Bevorzugung viel befahrener Strecken gewünscht, jedoch muss auch auf Nebenstraßen ein Vorankommen grundsätzlich möglich sein. Daher wurde sich dafür entschieden, dass die wartezeitbedingte Frustration quadratisch mit der Wartezeit anwächst. Die Möglichkeit einer

dauerhaften Sperrung wurde zudem ausgeschlossen, indem dem Algorithmus entsprechende Randbedingungen vorgegeben wurden. Letztendlich wurde folgendes einfache Regelwerk aufgestellt:

- Jedes Fahrzeug verliert 100 Frustpunkte, wenn es, ohne warten zu müssen, eine grüne Ampel passiert. Der Frustwert kann dabei auch negativ werden.
- Muss ein Fahrzeug halten, so steigert sich sein Frustwert um 100 und es beginnt damit, die Wartezeit zu messen.
- Fährt das Fahrzeug weiter, so erhöht sich der Frustwert, abhängig von der gemessenen Wartezeit nach der Formel: 4 x (Wartezeit in Sekunden)².

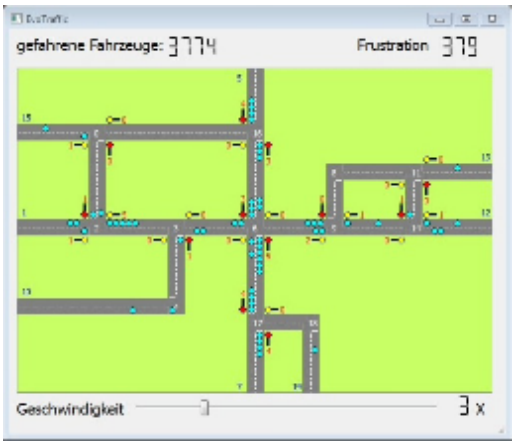

Abbildung 4: Bestimmung des Frustrationswerts

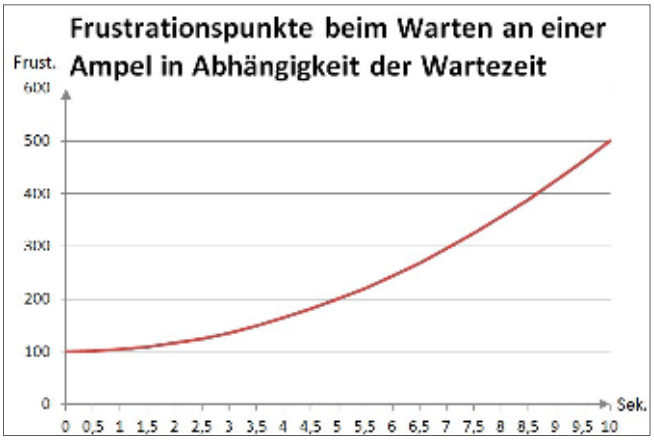

Abbildung 5: Frustrationszuwachs an roten Ampeln

Implementierung des evolutionären Algorithmus

Nachdem die Verkehrssimulation fertiggestellt wurde und auch die Bewertung des Verkehrsgeschehens anhand einer Kennziffer realisiert war, konnte mit der Implementierung des Optimierungsverfahrens begonnen werden. Wie schon bei der Simulation wurden dabei nicht nur rein funktionale Aspekte berücksichtigt, sondern es wurde auch Zeit in die Realisierung einer ansprechenden Darstellung der Ergebnisse investiert. Es wurde sich dabei für die naheliegende Darstellung aller Simulationsergebnisse in einem Stammbaum entschieden, wobei der jeweils erfolgreichste Parametersatz einer Generation, welcher der Ausgangspunkt für die Parametersätze der folgenden Generation ist, hervorgehoben wird. Ein Parametersatz übernimmt also in unserer simulierten Evolution die Rolle eines Individuums. In einem Parametersatz sind für die drei Parameter jeder der neun Kreuzungen feste Werte definiert. Wie bereits erwähnt sind die drei Parameter die Zeiten der beiden Grünphasen sowie die zeitliche Verschiebung des Zyklus. Diese drei Parameter werden auch im Simulationsfenster neben den entsprechenden Kreuzungen angezeigt. Die Simulation startet mit einem Parametersatz, bei dem alle Grünphasen 2,5 Sekunden lang dauern und keiner der Ampelzyklen einen Zeitversatz aufweist, also alle Ampeln synchron schalten. Während die Simulation läuft, werden ständig die ankommenden Fahrzeuge gezählt und die durchschnittliche Frustration wird ermittelt. Testläufe haben gezeigt, dass sich diese nach einigen hundert Fahrzeugen kaum mehr ändert, da sie sich dann auf den tatsächlichen, langfristigen Durchschnittswert eingeschwungen hat. Am Anfang des Simulationslaufs ist die Frustrationszahl gewöhnlich etwas niedriger, da die ersten Fahrzeuge auf zunächst noch leere Straßen treffen. Man könnte den Effekt umgehen, indem man die ersten Fahrzeuge nicht in die Wertung einbezieht, jedoch wurde hier darauf verzichtet. Die Simulation läuft so lange, bis insgesamt 1.000 Fahrzeuge ihr Ziel erreicht

haben. Ist dies geschehen, wird die nächste Generation an Parametersätzen erzeugt. Diese besteht aus 9 Parametersätzen, die durch zufällige Variation aller Einzelparameter erzeugt werden. Wie stark die neuen Parameter von den alten abweichen können, ist dabei vorgegeben und ändert sich auch von Generation zu Generation. Wie Abbildung 6 zeigt, kam es bei Versuchsläufen häufig vor, dass alle Parametersätze einer Generation ein schlechteres Ergebnis erzielten als ihr Vorfahre, aus dem sie erzeugt wurden. Daher wurde dieser als 10. Parametersatz in die neue Generation übernommen (zu sehen ist dies auf Abbildung 7), jedoch wurde er natürlich nicht noch einmal simuliert.

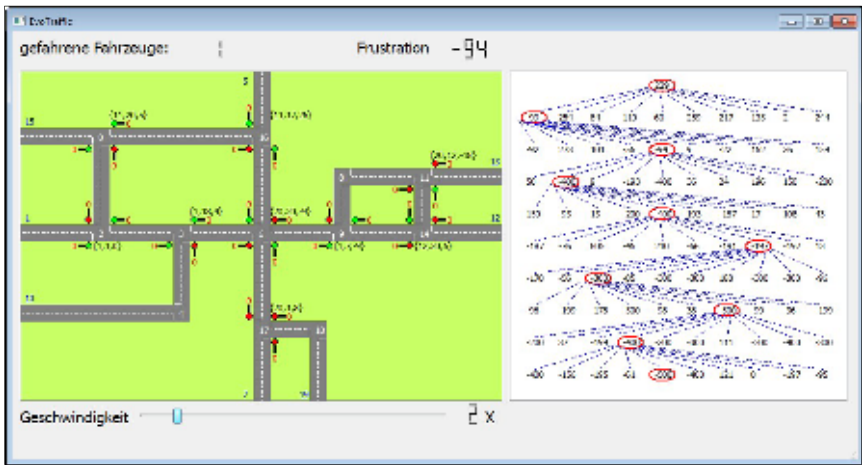

Abbildung 6: die Folgegenerationen haben zum Teil höhere Frustrationswerte als ihre Vorfahren

Das Programm simuliert dann alle Parametersätze der neuen Generation und zeigt die jeweiligen Ergebnisse an. Die Verwandtschaftsbeziehung zu dem Parametersatz, aus dem sie generiert wurden wird dabei durch gestrichelte Linien in einem Baumdiagramm dargestellt. Wurden alle Parametersätze einer Generation simuliert, wird aus dem mit dem besten Ergebnis wieder die nahfolgende Generation generiert. Dies kann auch der Vorfahr der Generation sein, da er ja als 10. Parametersatz gegen alle seine Nachkommen antritt. Dies ist zwar in einigen Bereichen, in denen die natürliche Evolution wirkt, anders, da dort die Vorfahren oft sterben bevor sie mit der neuen Generation in Konkurrenz treten können, jedoch verhindert es, dass gute Lösungen verloren gehen, ohne weiterverfolgt zu werden. Das Programm stoppt, sobald die 10. Generation simuliert und ausgewertet wurde. Es werden dann alle Ampeln mit den Parametern des Siegers dieser Generation konfiguriert und die Simulation wird gestartet, welche dann endlos weiterläuft. Bei maximaler Simulationsgeschwindigkeit benötigt das Programm ca. eine Stunde, um diesen Punkt zu erreichen.

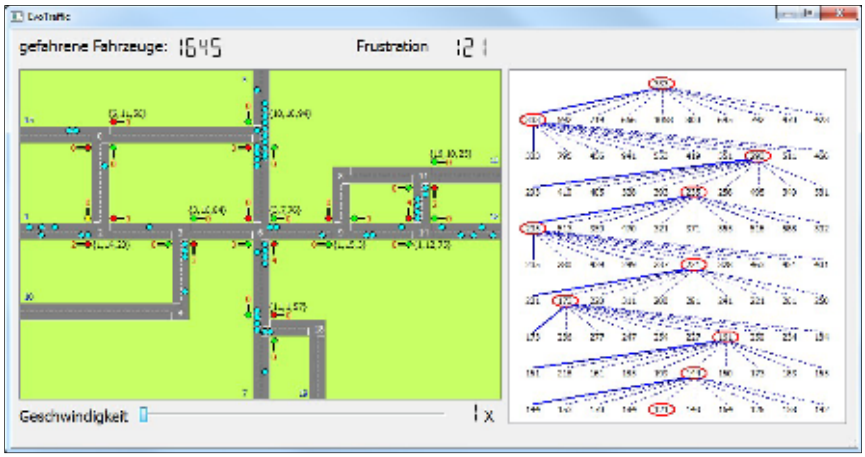

Abbildung 7: Programm nach Abschluss aller Simulationen

Wie bereits erwähnt nimmt die Variation der Parameter von Generation zu Generation immer mehr ab. Dies ist zwar bei echter Evolution anders – wobei auch hier die Mutationsrate variieren kann – jedoch wird dieser Trick verwendet, um in weniger Iterationen zu guten Ergebnissen zu kommen. In den ersten Generationen, in denen die Variation groß ist, werden verschiedene Konfigurationen, die alle sehr unterschiedlich sind, miteinander verglichen. Später werden die erfolgreichen Konfigurationen nur noch leicht verändert, um optimale Feineinstellungen zu ermitteln ohne jedoch das Gesamtzusammenspiel, das zum Erfolg der Konfiguration geführt hat, zu verwerfen. Das gleiche Prinzip wird auch bei Optimierungsverfahren in der Chipindustrie verwendet, deren Aufgabe es ist, eine Menge an Schaltungen möglichst effizient auf einem Stück Silizium zu platzieren. Die große Variation am Anfang hilft nicht nur dabei schneller zu einem Ergebnis zu kommen, sondern kann auch verhindern, dass sich das Verfahren in einem lokalen Optimum festfährt und dabei die tatsächliche Ideallösung übersieht. Man bezeichnet dieses Vorgehen auch als simuliertes Abkühlen, da beim physikalischen Prozess der Abkühlung die Teilchenbewegung ständig abnimmt.

Ein gutes Beispiel aus der Natur für das Festfahren der Evolution in einem lokalen Optimum ist das Auge der Wirbeltiere. Bei diesem sind nämlich die lichtempfindlichen Zellen auf der Netzhaut zum Inneren des Kopfes hin ausgerichtet, während die Nervenstränge, die die Signale dieser Zellen ans Gehirn übertragen, auf der Augeninnenseite angeschlossen sind. Dass es besser wäre, wenn die Zellen andersherum angeordnet wären, ist offensichtlich. Denn dadurch bekommen nicht nur die Sinneszellen weniger Licht ab, sondern es müssen auch die Nervenstränge erst durch die Netzhaut verlaufen, bevor sie das Gehirn erreichen. Der Punkt, an dem alle Nervenstränge der Sehzellen die Netzhaut passieren, ist der so genannte blinde Fleck, an dem die Netzhaut keine optischen Reize erfassen kann. Man könnte meinen, dass die Evolution derartig grobe Fehlkonstruktionen schnell ausselektiert, was jedoch anscheinend nicht geschehen ist. Die Begründung ist in der Komplexität des Auges und der großen Bedeutung dieses Sinnesorgans für die meisten sehenden Wirbeltiere zu finden. Denn eine so grundlegende Veränderung wie das neue Anordnen der Sehzellen würde sich zweifellos über unzählige Generationen hinziehen. Es wäre davon auszugehen, dass die zahllosen Zwischenschritte dabei weit schlechtere Eigenschaften hätten als die jetzige Form, die ja als Gesamteinheit sehr gut auf das Funktionieren mit der umgedrehten Netzhaut eingestellt ist. Daher hätten die betreffenden Individuen einen erheblichen Nachteil im Überlebenskampf und würden sterben, bevor sie ihre Mutation weitervererben könnten. Die langsam verlaufende Evolution kann das Wirbeltierauge also nicht verbessern, da sie es dafür zunächst verschlechtern müsste, was wiederum den Prinzipien der Evolution zuwiderläuft. Ironischer Weise sind gerade solche Schwächen die stärksten Belege für die Evolutionstheorie.

In der Simulation wird der Effekt durch die große Variation am Anfang ausgehebelt. Denn dadurch können auch verhältnismäßig große Umstrukturierungen innerhalb einer Generation stattfinden. Natürlich führt dies auch zu vielen Parametersätzen, die erheblich schlechter sind als ihre Vorfahren. Wie man in Abbildung 7 erkennen kann, ist die gesamte erste Generation schlechter als der Startwert, und zwar zum Teil mit einem nahezu dreifach höheren Frustrationswert. In den späteren Generationen nehmen die Unterschiede zwischen den Individuen deutlich ab, jedoch ist ab hier eine kontinuierliche Verbesserung des Frustrationswerts festzustellen. Auch in der Natur ist die überwältigende Mehrheit der Mutationen für die betroffenen Individuen nachteilhaft. Deshalb muss die Mutationsrate so klein sein wie sie ist, da es sonst passieren könnte, dass so viele Individuen einer Art nachteilhafte Mutationen haben, dass die Art ausstirbt. Oder anders ausgedrückt: Alle Arten, die jemals gelebt haben und hohe Mutationsrate aufwiesen, sind bereits ausgestorben.

Abschließende Bewertung

Das Projekt hat seine angestrebten Ziele erreicht. Es konnte gezeigt werden, dass sich evolutionäre Algorithmen eignen, um Ampelschaltzeiten zu optimieren. Die von dem im Rahmen dieser Arbeit erstellten Programm gefundene Lösung wies einen Frustrationswert auf, der kaum mehr als ein Drittel von dem der Startkonfiguration betrug. Und das, obwohl die Startkonfiguration keine besonders schlechte Einstellung war, wie die zahlreichen Iterationen mit deutlich schlechteren Ergebnissen zeigen. Es wurde auch klar, dass man sich der Mechanismen der Evolution und der Effekte, die sie hervorrufen, bewusst sein sollte, wenn man dieses Verfahren anwendet, da sie in den Simulationen ebenso auftreten wie in der Natur. Somit eignen sich derartige Computerprogramme nicht nur zum Lösen konkreter Aufgabenstellungen, sondern können auch dazu dienen, die Evolution und ihre Wirkprinzipien zu verstehen und zu vermitteln.

Quellen

[1] Dawkins, Richard: Der blinde Uhrmacher, Deutscher Taschenbuchverlag, München, 2008

[2] McClurg, Mike: Evolutionary Traffic Signals, Rose-Hulman Institute of Technology, o.J

[3] Wikipediaartikel über den Hahnenschweif-Widah:
http://en.wikipedia.org/wiki/Long-tailed_Widowbird, o.V, o.J, Einsichtnahme: 2.7.2012

Abbildungsverzeichnis

Abbildung 1: männlicher Hahnenschweif-Widah 4
Abbildung 2: Straßennetz 6
Abbildung 3: Routen 6
Abbildung 4: Bestimmung des Frustrationswerts 7
Abbildung 5: Frustrationszuwachs an roten Ampeln 7
Abbildung 6: die Folgegenerationen haben zum Teil höhere Frustrationswerte als ihre Vorfahren 8
Abbildung 7: Programm nach Abschluss aller Simulationen 8