

Beschleunigung der Gleichgewichtssuche in agenten-basierten Systemen

Elias Aptus

Semesterarbeit Raum- und Verkehrsplanung

Juni 2010

Inhaltsverzeichnis

1	Einleitung	1
2	Parameterwerte	2
2.1	Bereits untersuchte Parameterwerte	2
2.2	Die Wirkung von SelectExpBeta	2
2.3	Die Wirkung von ReRoute und TimeAllocationMutator	3
3	Kriterium für Equilibriumnähe: ChangeExpBeta	10
3.1	Ergebnisse in Equinet	13
3.2	Ergebnisse im Zürich-Szenario	17
4	Fazit	22

Abbildungsverzeichnis

1	Score abhängig vom Wert von SelectExpBeta mit 10 Iterationen	4
2	Score abhängig vom Wert von SelectExpBeta mit 20 Iterationen	4
3	Score abhängig vom Wert von SelectExpBeta mit 30 Iterationen	5
4	Score abhängig vom Wert von SelectExpBeta mit 40 Iterationen	5
5	Score abh. von Nummer der Iteration. Blau: SelectExpBeta = 0, rot: SelectExpBeta = 0.8	7
6	Score abhängig vom Wert von ReRoute mit 10 Iterationen	8
7	Score abhängig vom Wert von ReRoute mit 20 Iterationen	8
8	Score abhängig vom Wert von ReRoute mit 30 Iterationen	9
9	Dauer der Simulation in Sekunden abhängig vom Wert von ReRoute mit 10 Iterationen	10
10	Dauer der Simulation in Sekunden abhängig vom Wert von ReRoute mit 20 Iterationen	11
11	Dauer der Simulation in Sekunden abhängig vom Wert von ReRoute mit 30 Iterationen	11
12	Gleichgewichtsnähe abhängig vom gleitenden Mittelwert der Wechsel- quote	14
13	Gleichgewichtsnähe abhängig vom gleitenden Mittelwert der Wechsel- quote	16
14	Score im Zürich-Szenario abh. von der Nummer der Iteration für Chan- geExpBeta = 0.2	17
15	Gleitender Mittelwert über 50 Iterationen der Wechselquote im Zürich- Szenario abhängig von der Nummer der Iteration für ChangeExpBeta = 0.2	18

16	Durchschnittlicher ausgeführter Score im Zürich-Szenario abhängig von der Nummer der Iteration (0 bis 500) für ChangeExpBeta = 0.8	19
17	Gleitender Mittelwert über 50 Iterationen der Wechselquote im Zürich-Szenario abhängig von der Nummer der Iteration für ChangeExpBeta = 0.8	19
18	Score im Zürich-Szen. abh. von Num. der It. für ChangeExpBeta = 0.8 in stabilem Modus	20
19	Gleit. Mittelw. d. Wechselq. im Zürich-Szen. für ChangeExpBeta = 0.8 in stabilem Modus	21

Tabellenverzeichnis

1	Beschreibung der Abbildungen 1 - 5	6
2	Beschreibung der Abbildungen 6 - 8	9
3	Beschreibung der Abbildungen 9 - 11	12

Term Project

Beschleunigung der Gleichgewichtssuche in agenten-basierten Systemen

Elias Aptus
IVT
ETH Hönggerberg
CH-8093 Zurich
eaptus@student.ethz.ch

Juni 2010

Abstract

In this term project, the speed of convergence to a system equilibrium of the agent-based simulation tool MATSim is investigated. The aim is to accelerate the convergence to equilibrium. To achieve that, on the one hand parameter values of MATSim are found, which yield a faster convergence, and on the other hand a criterion for equilibrium is proposed, with the goal to reduce the minimum number of iterations necessary to guarantee arrival at the equilibrium.

Keywords

equilibrium, agent-based simulation, MATSim

Preferred citation style

Elias Aptus (2010) Beschleunigung der Gleichgewichtssuche in agenten-basierten Systemen, *Semesterarbeit Raum- und Verkehrsplanung*, Institute for Transport Planning and Systems (IVT), ETH Zurich, Zurich.

Semesterarbeit Raum- und Verkehrsplanung

Beschleunigung der Gleichgewichtssuche in agenten-basierten Systemen

Elias Aptus
IVT
ETH Hönggerberg
CH-8093 Zurich
eaptus@student.ethz.ch

Juni 2010

Zusammenfassung

In dieser Semesterarbeit wird die Geschwindigkeit der Konvergenz des agenten-basierten Simulationstools MATSim zum Systemequilibrium untersucht, mit dem Ziel, die Konvergenz zum Equilibrium zu beschleunigen. Dafür werden einerseits Parameterwerte von MATSim, die eine schnelle Konvergenz liefern, gefunden, und andererseits wird ein Kriterium für Equilibriumnähe untersucht, mit dem Ziel, die minimale Anzahl von Iterationen, die nötig ist, um mit Gewissheit im Equilibrium angelangt zu sein, zu verkleinern.

Schlagworte

Gleichgewicht, agenten-basierte Simulation, MATSim

Bevorzugter Zitierstil

Elias Aptus (2010) Beschleunigung der Gleichgewichtssuche in agenten-basierten Systemen, *Semesterarbeit Raum- und Verkehrsplanung*, Institut für Verkehrsplanung und Transportsysteme (IVT), ETH Zürich, Zürich.

1. Einleitung

Agentenbasierte Systeme werden sehr häufig verwendet, um Aufschluss über den Verkehrsfluss in Netzwerken zu erhalten (Raney und Nagel, 2006; Meister *et al.*, 2009). Vor allem um das Gleichgewicht eines Verkehrssystems zu bestimmen, werden oft Simulationen eingesetzt. Das Ziel ist dabei, dass nach einigen Iterationen, in denen jeder Agent sein Verhalten optimiert, das Gesamtsystem nahe am Equilibriumverkehrsfluss landet. Bei Netzwerken mit vielen Knoten und Verbindungen zwischen ihnen, sowie bei einer großen Anzahl von Agenten, ist der Rechenaufwand, den ein Computer benötigt, um so viele Iterationen einer Simulation laufen zu lassen, bis sich das Gleichgewicht einpendelt, oft sehr groß, was die Mächtigkeit der Simulationsmethode stark einschränkt.

Das in dieser Semesterarbeit untersuchte Simulationsprogramm ist MATSim (Raney und Nagel, 2006). Wir verwenden die Version MATSim 20090108 und die Version MATSim r11545¹. Bei einer Simulation mit MATSim werden dem Programm Parameterwerte übergeben, die vom Benutzer gewählt werden können.

Das erste Ergebnis der Semesterarbeit sind Werte für einige der Parameter, die eine schnelle Konvergenz zum Equilibrium erlauben. Das zweite Ergebnis ist ein Kriterium, mit dem man Informationen darüber gewinnen kann, wie nahe man bereits am Equilibrium ist. Falls man bereits nahe am Equilibrium ist, lohnt es sich oft nicht, noch weitere zeitaufwendige Iterationen durchzuführen. Es ist offensichtlich besser, die Simulation bei genügender Equilibriumnähe zu beenden und so Zeit zu sparen.

Die Ergebnisse, die hier vorgestellt werden, basieren auf zwei Szenarien, die in MATSim simuliert wurden. Das erste Szenario heißt Equinet (siehe Raney und Nagel, 2006). Es ist ein künstliches und einfaches Netzwerk und wurde mit 2000 Agenten auf einem PC simuliert. Das zweite Szenario ist ein Modell des Züricher Straßennetzes. Es enthält die wichtigsten Knoten und Verbindungen des Züricher Straßennetzes (siehe Balmer *et al.*, 2009), und wurde mit 60000 Agenten auf dem ETHZ Cluster Brutus simuliert.

In Raney und Nagel (2006) wird die Struktur und Funktionsweise von MATSim ausführlich beschrieben, sowie einige Parameteranalysen durchgeführt. Eine mathematische Analyse und Verbesserung der Konvergenz zum stochastischen Nutzerequilibrium eines Systems unter Verwendung der erweiterten Methode der sukzessiven Durchschnitte findet sich in Liu *et al.* (2009). Nagel und Flötteröd (2009) geben eine all-

¹Siehe www.matsim.org

gemeine Übersicht über verschiedene Ansätze zur Verkehrsumlegung und binden die Methode der Simulation in diesen Kontext ein. In Zöllig und Axhausen (2010) werden verschiedene Dimensionen der Modellierung aufgezeigt und analysiert. Die Analyse geschieht dabei anhand einer agenten-basierten Simulation auf einem künstlichen kleinen Netzwerk. Schließlich wird in Charypar *et al.* (2006), wie auch in dieser Semesterarbeit, die Simulation mit MATSim beschleunigt, indem jedoch nicht allein Parameter untersucht werden, sondern Modulverbesserungen vorgestellt werden.

Im nächsten Kapitel werden Parameterwerte bestimmt, die für eine schnelle Konvergenz der Simulation sorgen. In Kapitel 3 wird ein Kriterium für die Erreichung des Nähe des Equilibriums untersucht. Kapitel 4 enthält einen Fazit.

2. Parameterwerte

2.1 Bereits untersuchte Parameterwerte

In Raney und Nagel (2006) wurden bereits einige Parameter untersucht. So wurden β , ein interner Parameter im Modul „SelectExpBeta“ und β_{travel} , der marginale Nutzen der Reisezeit, untersucht. Die Analyse basiert in Raney und Nagel (2006) auf den Statistiken der verschiedenen Scores und Reisezeiten bei verschiedenen Parameterwerten und -kombinationen. Unsere Analyse der Parameter wird ähnlich aufgebaut sein, wobei wir uns vor allem an den durchschnittlichen Scorewerten orientieren werden. In Raney und Nagel (2006) werden hauptsächlich Simulationen betrachtet, in denen nur Routenumplanung beziehungsweise nur Zeitumplanung stattfindet. Wir werden untersuchen, welches Verhältnis zwischen Zeit- und Routenumplanung das beste Ergebnis liefert. Alle folgenden Analysen, in denen nicht erwähnt wird, dass sie mit dem Zürich Szenario durchgeführt wurden, wurden mit Equinet, einem künstlichen und einfachen Netzwerk, mit 2000 Agenten durchgeführt.

2.2 Die Wirkung von SelectExpBeta

Der Parameter ModuleProbability für das Modul SelectExpBeta bestimmt den Anteil der Agenten, die mithilfe des Moduls SelectExpBeta aus den zur Verfügung stehenden Plänen einen Plan zur Ausführung in der gegebenen Iteration auswählen. Im Modul

SelectExpBeta wird jedem Plan eine Auswahlwahrscheinlichkeit zugeordnet, die als Funktion des Scores des Plans monoton wachsend ist.

In unserer Simulation wurden dem Modul „strategy“ folgende Parameter übergeben:

```
<module name="strategy">
  <param name="maxAgentPlanMemorySize" value="6" />

  <param name="ModuleProbability_1" value="x" />
  <param name="Module_1" value="SelectExpBeta" />

  <param name="ModuleProbability_2" value="y" />
  <param name="Module_2" value="TimeAllocationMutator" />

  <param name="ModuleProbability_3" value="y" />
  <param name="Module_3" value="ReRoute" />
</module>
```

für $x = 0, 0.1, 0.2, \dots, 0.9, 1$ und $y \geq 0$, sodass $x + 2y = 1$.

Motivation dieser Anordnung: wir erwarten, dass der optimale Wert für SelectExpBeta² in etwa unabhängig ist von den einzelnen Werten der Parameter TimeAllocationMutator und ReRoute, sondern nur abhängig ist von ihrer Summe. Im Folgenden betrachten wir den durchschnittlichen (über alle Agenten) ausgeführten Score (DAS) abhängig vom Wert von SelectExpBeta und für verschiedene Anzahlen von Iterationen. Die Ergebnisse sind dargestellt in den Abbildungen 1 - 5. Tabelle 1 beschreibt diese Abbildungen.

Insgesamt sieht man an den Abbildungen 1 - 5, dass SelectExpBeta = 0 für gegebene Anzahl von Iterationen den besten Score liefert.

2.3 Die Wirkung von ReRoute und TimeAllocationMutator

Der Parameter ReRoute bestimmt den Anteil der Agenten, welche ihre Routen neu planen. Der Parameter TimeAllocationMutator bestimmt den Anteil der Agenten, die ihre Startzeit neu planen. In unserer Simulation wurden dem Modul „strategy“ folgende

²Eigentlich ist gemeint der Wert des Parameters ModuleProbability für das Modul SelectExpBeta. Der kürzeren und bündigeren Darstellung halber wird der Modulname ab dieser Stelle stellvertretend für den Parameter ModuleProbability im betrachteten Modul verwendet. Falls das Modul SelectExpBeta selbst, und nicht der Wert des Parameters ModuleProbability gemeint ist, wird dies explizit angegeben. Dasselbe gilt für die Module ReRoute, TimeAllocationMutator und ChangeExpBeta.

Abbildung 1: Durchschnittlicher ausgeführter Score abhängig vom Wert von SelectExpBeta mit 10 Iterationen

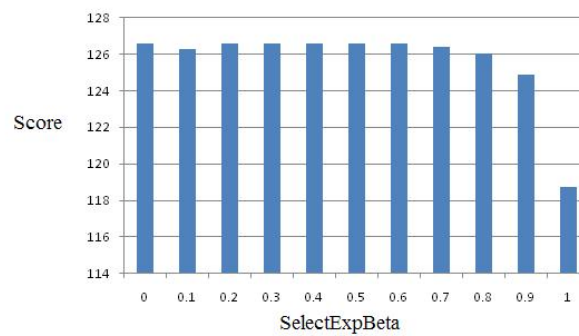


Abbildung 2: Durchschnittlicher ausgeführter Score abhängig vom Wert von SelectExpBeta mit 20 Iterationen

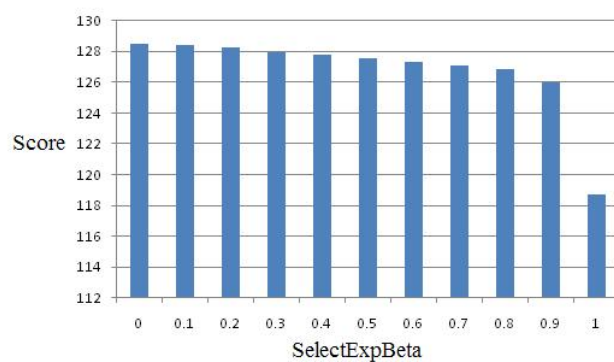


Abbildung 3: Durchschnittlicher ausgeführter Score abhängig vom Wert von SelectExpBeta mit 30 Iterationen

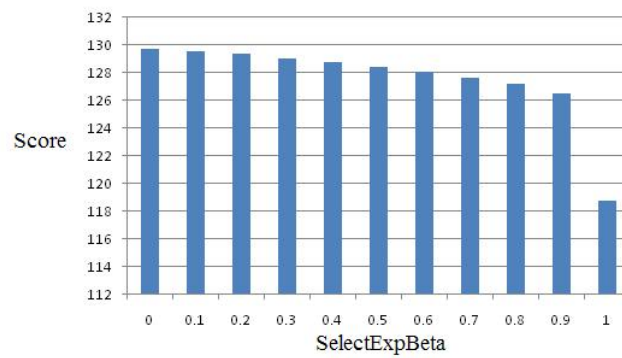


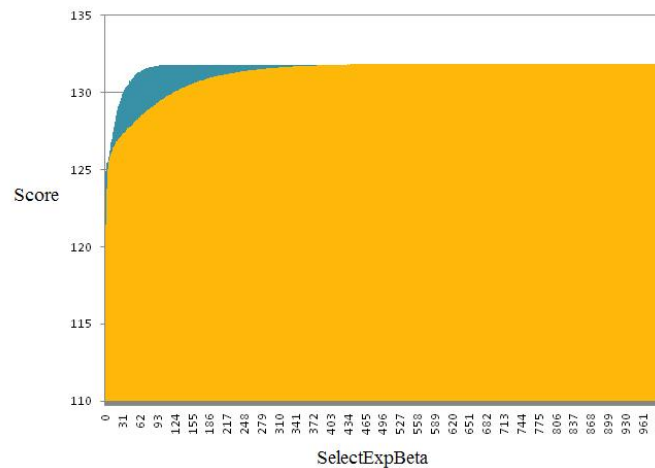
Abbildung 4: Durchschnittlicher ausgeführter Score abhängig vom Wert von SelectExpBeta mit 40 Iterationen



Tabelle 1: Beschreibung der Abbildungen 1 - 5

- Abbildung 1: Der durchschnittliche (über alle Agenten) ausgeführte Score (DAS) beträgt 126.6 für $\text{SelectExpBeta} = 0$. Der Durchschnitt von DAS über alle Werte von SelectExpBeta beträgt 125.6, die Standardabweichung beträgt 2.4, und der maximale Wert von DAS beträgt 126.6.
- Abbildung 2: Der DAS beträgt 128.5 für $\text{SelectExpBeta} = 0$. Der Durchschnitt von DAS über alle Werte von SelectExpBeta beträgt 126.8, die Standardabweichung beträgt 2.8, und der maximale Wert von DAS beträgt 128.5.
- Abbildung 3: Der DAS beträgt 129.7 für $\text{SelectExpBeta} = 0$. Der Durchschnitt von DAS über alle Werte von SelectExpBeta beträgt 127.5, die Standardabweichung beträgt 3.2, und der maximale Wert von DAS beträgt 129.7.
- Abbildung 4: Der DAS beträgt 130.5 für $\text{SelectExpBeta} = 0$. Der Durchschnitt von DAS über alle Werte von SelectExpBeta beträgt 128.2, die Standardabweichung beträgt 3.4, und der maximale Wert von DAS beträgt 130.5.
- Abbildung 5: Für die Werte 0 und 0.8 von SelectExpBeta pendelt sich nach genügend vielen Iterationen derselbe Equilibriumscore (131.87) ein. Jedoch wird für $\text{SelectExpBeta}=0$ nach einer wesentlich kleineren Anzahl von Iterationen (100) ein Wert, nämlich 131.78, der sehr nahe am Equilibriumscore ist, erreicht.
-

Abbildung 5: Durchschnittlicher ausgeführter Score abhängig von der Nummer der Iteration. Blau: SelectExpBeta = 0, rot: SelectExpBeta = 0.8



Parameter übergeben:

```

    <module name="strategy">
      <param name="maxAgentPlanMemorySize" value="6" />

      <param name="ModuleProbability_1" value="0" />
      <param name="Module_1" value="SelectExpBeta" />

      <param name="ModuleProbability_2" value="1-x" />
      <param name="Module_2" value="TimeAllocationMutator" />

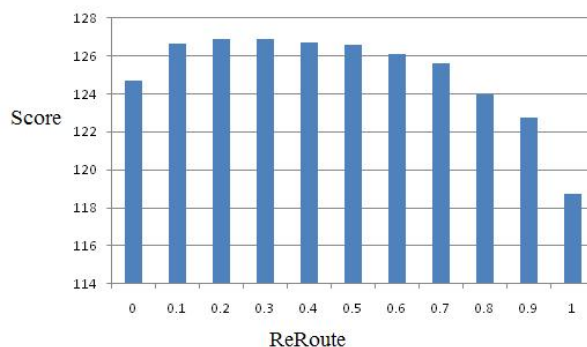
      <param name="ModuleProbability_3" value="x" />
      <param name="Module_3" value="ReRoute" />
    </module>

```

für $x = 0, 0.1, 0.2, \dots, 0.9, 1$. Motivation dieser Anordnung: die Untersuchung der Werte von SelectExpBeta hat gezeigt, dass der optimale Wert für SelectExpBeta der Wert 0 ist.

Im Folgenden betrachten wir den durchschnittlichen (über alle Agenten) ausgeführten Score (DAS) abhängig vom Wert von ReRoute und von TimeAllocationMutator und für

Abbildung 6: Durchschnittlicher ausgeführter Score abhängig vom Wert von ReRoute mit 10 Iterationen



verschiedene Anzahlen von Iterationen. Die Ergebnisse sind dargestellt in den Abbildungen 6 - 8. Tabelle 2 beschreibt diese Abbildungen.

Man sieht also an den Abbildungen 6-8, dass für gegebene Anzahl von Iterationen der beste Score erreicht wird mit $\text{ReRoute} = 0.1$ (und $\text{TimeAllocationMutator} = 0.9$).

Als nächstes betrachten wir die Dauer der Simulation in Sekunden abhängig vom Wert von ReRoute und von $\text{TimeAllocationMutator}$ und für verschiedene Anzahlen von Iterationen. Die Ergebnisse sind dargestellt in den Abbildungen 9 - 11. Tabelle 3 beschreibt diese Abbildungen.

Man sieht also an den Abbildungen 9-11 also, dass die Gesamtdauer der Simulation am kleinsten ist für kleine Werte (um 0.1) von ReRoute .

Insgesamt sieht man also, dass mit einem kleinen Wert von ReRoute sowohl der beste Score erreicht wird, als auch die kleinste Gesamtdauer der Simulation (bei gegebener Anzahl von Iterationen) benötigt wird.

3. Kriterium für Equilibriumnähe: ChangeExpBeta

Eine wichtige Rolle für die Dauer von Simulationen spielt die Antwort auf die Frage, wann das Gleichgewicht erreicht wird. Der einfachste Indikator für Gleichgewichtsnähe ist der unveränderte Score über eine genügend große Anzahl von aufeinanderfolgenden

Abbildung 7: Durchschnittlicher ausgeführter Score abhängig vom Wert von ReRoute mit 20 Iterationen

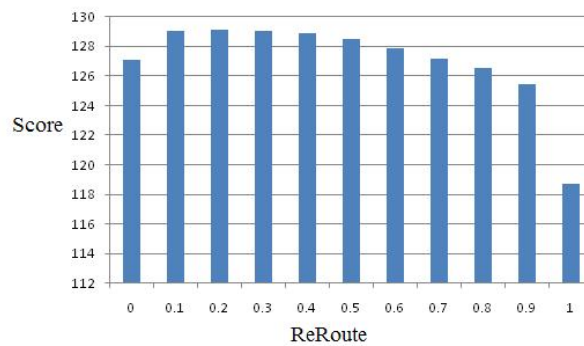


Abbildung 8: Durchschnittlicher ausgeführter Score abhängig vom Wert von ReRoute mit 30 Iterationen

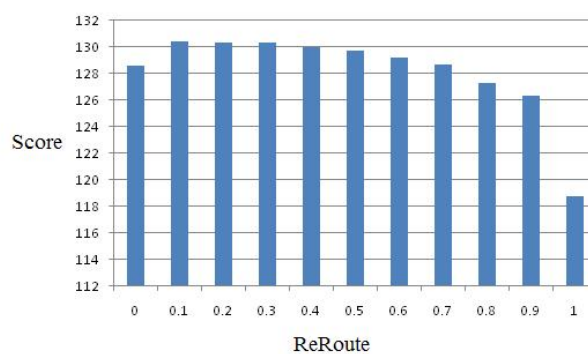


Tabelle 2: Beschreibung der Abbildungen 6 - 8

- Abbildung 6: Der durchschnittliche (über alle Agenten) ausgeführte Score (DAS) beträgt 126.6 für ReRoute = 0.1 (und TimeAllocationMutator=0.9). Der Durchschnitt von DAS über alle Werte von ReRoute beträgt 125.1, die Standardabweichung beträgt 2.6, und der maximale Wert von DAS beträgt 126.9.
- Abbildung 7: Der DAS beträgt 129.1 für ReRoute = 0.1 (und TimeAllocationMutator=0.9). Der Durchschnitt von DAS über alle Werte von ReRoute beträgt 127.0, die Standardabweichung beträgt 3.2, und der maximale Wert von DAS beträgt 129.1.
- Abbildung 8: Der DAS beträgt 130.4 für ReRoute = 0.1 (und TimeAllocationMutator=0.9). Der Durchschnitt von DAS über alle Werte von ReRoute beträgt 128.1, die Standardabweichung beträgt 3.6, und der maximale Wert von DAS beträgt 130.4.

Abbildung 9: Dauer der Simulation in Sekunden abhängig vom Wert von ReRoute mit 10 Iterationen

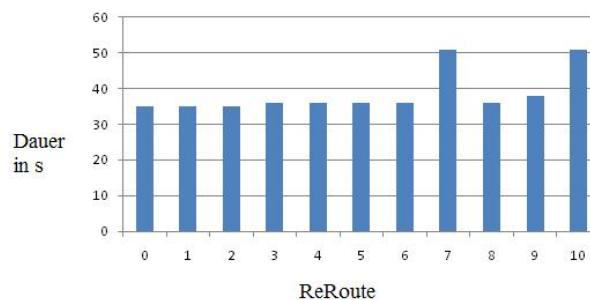


Abbildung 10: Dauer der Simulation in Sekunden abhängig vom Wert von ReRoute mit 20 Iterationen

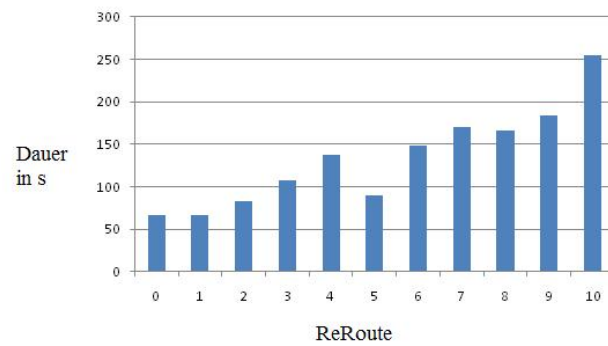


Abbildung 11: Dauer der Simulation in Sekunden abhängig vom Wert von ReRoute mit 30 Iterationen

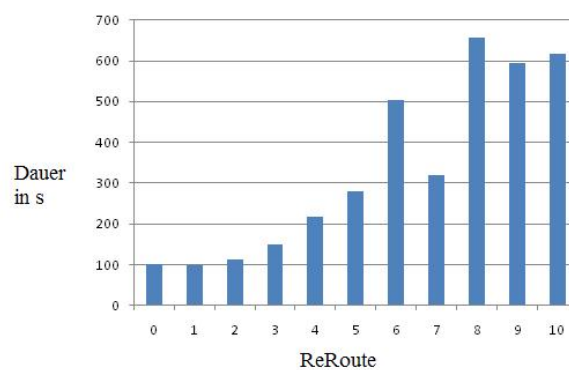


Tabelle 3: Beschreibung der Abbildungen 9 - 11

Abbildung 9:	Die Laufzeit von 10 Iterationen beträgt 35 Sekunden für $ReRoute = 0.1$. Die durchschnittliche Laufzeit von 10 Iterationen über alle Werte von $ReRoute$ ist 38.6 Sekunden, die Standardabweichung beträgt 6.2, und die minimale Laufzeit von 10 Iterationen beträgt 35 Sekunden.
Abbildung 10:	Die Laufzeit von 20 Iterationen beträgt 67 Sekunden für $ReRoute = 0.1$. Die durchschnittliche Laufzeit von 20 Iterationen über alle Werte von $ReRoute$ ist 134.1 Sekunden, die Standardabweichung beträgt 58.3, und die minimale Laufzeit von 20 Iterationen beträgt 66 Sekunden.
Abbildung 11:	Die Laufzeit von 30 Iterationen beträgt 99 Sekunden für $ReRoute = 0.1$. Die durchschnittliche Laufzeit von 30 Iterationen über alle Werte von $ReRoute$ ist 332 Sekunden, die Standardabweichung beträgt 221.2, und die minimale Laufzeit von 30 Iterationen beträgt 99 Sekunden.

Iterationen. Um dieses Kriterium zu nutzen, muss man allerdings nachdem man das Equilibrium schon erreicht hat (aber es noch nicht weiß), noch eine genügend große Anzahl von Iterationen durchführen. Diese Iterationen verbessern den Score nicht wesentlich, kosten dabei aber Zeit. Sie sind somit retrospektiv eine Verschwendung. Deshalb wäre ein Kriterium für das Erreichen des Gleichgewichts, welches ohne zusätzliche Iterationen auskommen würde, sehr nützlich.

Eine Idee für ein solches Kriterium liefert das Modul `ChangeExpBeta`. Die Aufgabe dieses Moduls ist gleich wie die Aufgabe von `SelectExpBeta`, nämlich die Auswahl eines Plans aus der Menge der verfügbaren Pläne eines gegebenen Agenten. Konzeptuell verfährt `ChangeExpBeta` ähnlich wie `SelectExpBeta`, und wählt einen Plan mit hoher Wahrscheinlichkeit aus, falls sein Score hoch ist. `ChangeExpBeta` hat aber den Vorteil, dass man die Wahrscheinlichkeit, mit der ein Agent von seinem aktuellen Plan (also dem Plan, den er in der vorhergegangenen Iteration ausgeführt hat) zu einem anderen Plan wechselt, direkt ablesen kann. Es ist intuitiv, dass im Equilibrium diese Wechselwahrscheinlichkeit für alle Agenten niedrig sein sollte. Nun stellt sich die Frage, ob umgekehrt niedrige Wechselwahrscheinlichkeiten ein Indikator für Equilibriumnähe sind. Diese Frage werden wir im Folgenden untersuchen und daraus mögliche Kriterien für Equilibriumnähe herleiten.

Dem Modul „strategy“ wurden folgende Parameter übergeben:

```
<module name="strategy">
  <param name="maxAgentPlanMemorySize" value="6" />

  <param name="ModuleProbability_1" value="0.8" />
  <param name="Module_1" value="ChangeExpBeta2" />

  <param name="ModuleProbability_2" value="0.1" />
  <param name="Module_2" value="TimeAllocationMutator" />

  <param name="ModuleProbability_3" value="0.1" />
  <param name="Module_3" value="ReRoute" />
</module>
```

und

```
<module name="strategy">
  <param name="maxAgentPlanMemorySize" value="6" />

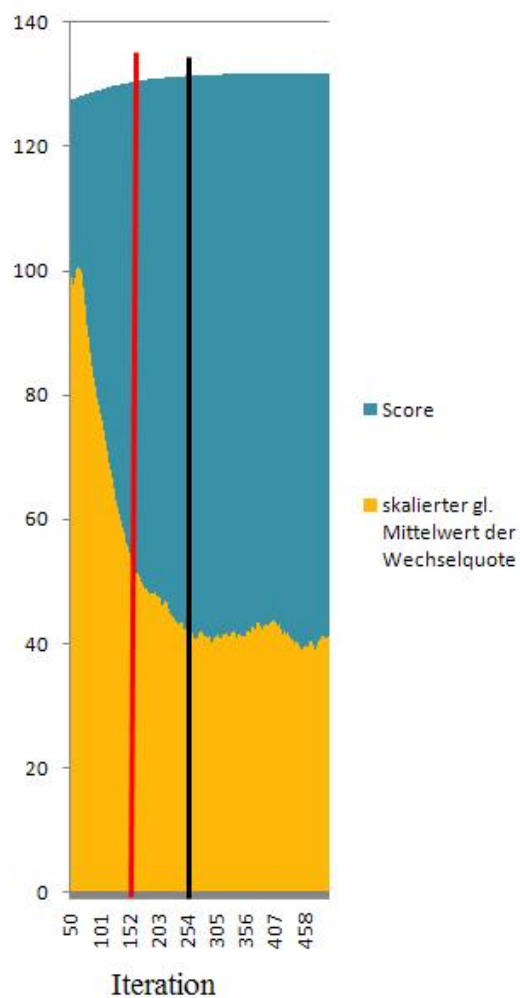
  <param name="ModuleProbability_1" value="0.2" />
  <param name="Module_1" value="ChangeExpBeta2" />

  <param name="ModuleProbability_2" value="0.4" />
  <param name="Module_2" value="TimeAllocationMutator" />

  <param name="ModuleProbability_3" value="0.4" />
  <param name="Module_3" value="ReRoute" />
</module>
```

Das Modul ChangeExpBeta2 (welches bis jetzt in keiner Version von MATSim implementiert ist) tut dasselbe wie ChangeExpBeta, und gibt zusätzlich die Quote aller Agenten, die ihren Plan gewechselt haben, aus. Diese Quote kann man, bei genügend großer Anzahl Agenten, als eine Näherung für die aggregierte Wechselwahrscheinlichkeit verwenden.

Abbildung 12: Score und mit dem Faktor 4000 skaliertes gleitendes Mittelwert der Wechselquote in Equinet abhängig von der Nummer der Iteration für $\text{ChangeExpBeta} = 0.8$



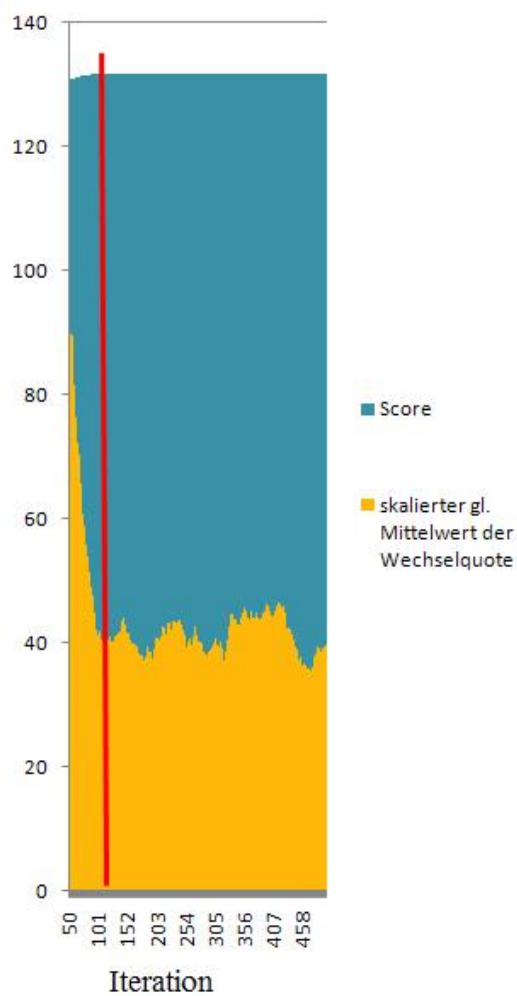
3.1 Ergebnisse in Equinet

Das Equilibrium (131.8) wird für $\text{ChangeExpBeta}=0.8$ zum ersten Mal erreicht nach 430 Iterationen. Für diesen Wert von ChangeExpBeta erreicht der gleitende Mittelwert der Wechselquote ab 236 Iterationen den Wert 0.01 (entspricht dem skalierten Wert 40, siehe schwarze Linie in der Abbildung 12), wonach er mit einer Standardabweichung von 0.0009 um diesen Wert streut. Der durchschnittliche ausgeführte Score nach 150 Iterationen beträgt 130.2 (siehe rote Linie in der Abbildung 12), ist also nahe am Equilibriumscore. In diesem Fall wäre also die Nummer der Iteration, in der der gleitende Mittelwert der Wechselwahrscheinlichkeit zum ersten mal den Wert 0.01 erreicht, aus Sicht des Scores ein gutes Kriterium, um die Simulation anzuhalten. Die kleine Standardabweichung deutet darauf hin, dass es sinnvoll wäre, den gleitenden Mittelwert über eine kleinere Anzahl von Iterationen zu betrachten, um zu prüfen, ob das Kriterium auch mit einer kleineren Anzahl von Iterationen, über die gleitend gemittelt wird, verwendet werden kann (was von Vorteil wäre). Anzumerken ist schließlich, dass der gleitende Mittelwert der Wechselquote (über 50 Iterationen) während der Iterationen 50 bis 63 bei zwischen 0.024 und 0.025 liegt, und erst ab der 63. Iteration anfängt sich dem Wert 0.01 zu nähern.

Das Equilibrium für $\text{ChangeExpBeta} = 0.2$ wird erreicht nach 136 Iterationen. Für diesen Wert von ChangeExpBeta erreicht der gleitende Mittelwert der Wechselquote ab 103 Iterationen den Wert 0.01 (entspricht dem skalierten Wert 40, siehe rote Linie in der Abbildung 13), wonach er mit einer Standardabweichung von 0.0007 um diesen Wert streut. Der durchschnittliche ausgeführte Score nach 100 Iterationen beträgt 131.7 (siehe Abbildung 13), ist also nahe am Equilibriumscore. In diesem Fall wäre also die Nummer der Iteration, in der der gleitende Mittelwert der Wechselwahrscheinlichkeit zum ersten mal den Wert 0.1 erreicht, aus Sicht des Scores ein gutes Kriterium, um die Simulation anzuhalten. Die kleine Standardabweichung deutet darauf hin, dass es sinnvoll wäre, den gleitenden Mittelwert über eine kleinere Anzahl von Iterationen zu betrachten um zu prüfen, ob das Kriterium auch mit einer kleineren Anzahl von Iterationen, über die gleitend gemittelt wird, verwendet werden kann (was von Vorteil wäre).

Die Abbildungen 12 und 13 weisen also darauf hin, dass eine Verbindung zwischen dem gleitenden Mittelwert der Wechselquote und der Equilibriumnähe besteht. Es sieht

Abbildung 13: Score und mit dem Faktor 4000 skaliertes gleitendes Mittelwert der Wechselquote in Equinet abhängig von der Nummer der Iteration für $\text{ChangeExpBeta} = 0.2$



danach aus, dass nach einer Iteration, in der der gleitende Mittelwert der Wechselquote etwa konstant bei 0.1 liegt, ein Score erreicht wurde, der Nahe am Equilibriumscore liegt. Der konstante gleitende Mittelwert der Wechselquote scheint also ein Indikator für Equilibriumnähe zu sein, und kann als Alternative zum gleitenden Mittelwert über den Score genommen werden.

Auch falls man das Modul `ChangeExpBeta` nicht zur Equilibriumsuche verwendet, so kann man, wenn man dieses Kriterium für Equilibriumnähe verwenden will, das Modul `ChangeExpBeta` (mit kleinen Änderungen im Code) zur Ausgabe der Wechselquote verwenden, ohne es zur Planwahl zu verwenden.

Anmerkung: Simulationen mit den beiden Versionen `MATSim 20090108` und `MATSim r11545` haben im Equinet Szenario unterschiedliche Equilibria berechnet. Wir wissen nicht, woran das liegt, und konnten aufgrund dieser Tatsache nicht die Resultate mit einander vergleichen. In diesem Kapitel wurden nur die Ergebnisse vorgestellt, die mit `MATSim r11545` produziert wurden. Im nächsten Kapitel, welches nach dem Versionswechsel entstanden ist, wurden alle Ergebnisse mit der Version `MATSimr11545` erzeugt.

3.2 Ergebnisse im Zürich-Szenario

Der Score verhält sich für `ChangeExpBeta=0.2` über 500 Iterationen chaotisch (Mittelwert=-560.8, Standardabweichung=358.2, siehe Abbildung 14). Der gleitende Mittelwert der Wechselquote hingegen liegt nach etwa 130 Iterationen nahe bei 0.01 (Mittelwert = 0.11), mit einer Standardabweichung von 0.002 (siehe Abbildung 15). In diesem Fall wäre also die Nummer der Iteration, in der der gleitende Mittelwert der Wechselwahrscheinlichkeit zum ersten mal den Wert 0.1 erreicht, ein zu optimistisches Kriterium für Equilibriumnähe.

Der Score verhält sich für `ChangeExpBeta=0.8` über 500 Iterationen chaotisch (Mittelwert=109.1, Standardabweichung=84.3, siehe Abbildung 16). Der gleitende Mittelwert der Wechselquote, der monoton fallend ist, liegt ebenfalls sogar während den Iterationen 400 bis 500 deutlich über 0.01, (Mittelwert = 0.022), mit einer Standardabweichung von 0.0003 (siehe Abbildung 17). In diesem Fall wäre also der gleitende Mittelwert der Wechselwahrscheinlichkeit, der nicht um den Wert 0.1 streut, ein gutes

Abbildung 14: Durchschnittlicher ausgeführter Score im Zürich-Szenario abhängig von der Nummer der Iteration (0 bis 500) für $\text{ChangeExpBeta} = 0.2$

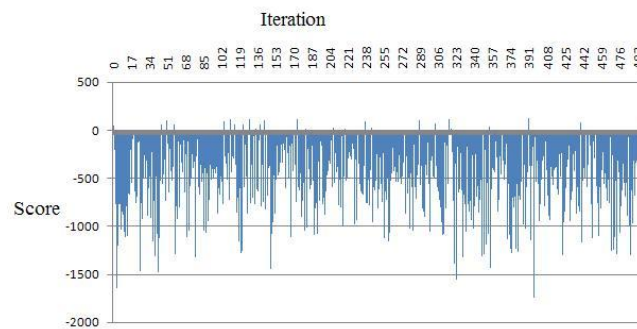


Abbildung 15: Gleitender Mittelwert über 50 Iterationen der Wechselquote im Zürich-Szenario abhängig von der Nummer der Iteration für $\text{ChangeExpBeta} = 0.2$

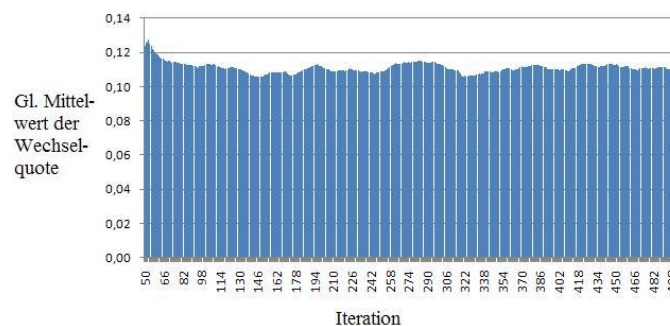


Abbildung 16: Durchschnittlicher ausgeführter Score im Zürich-Szenario abhängig von der Nummer der Iteration (0 bis 500) für ChangeExpBeta = 0.8

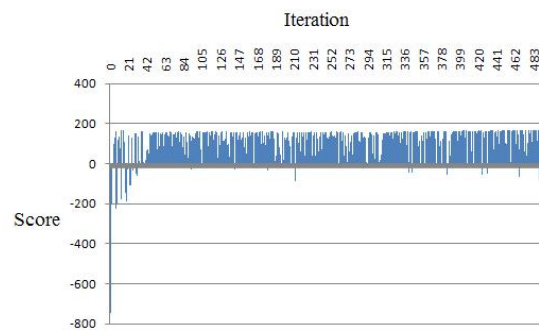
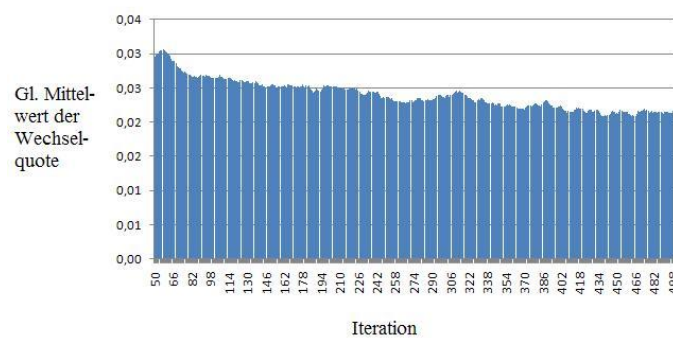


Abbildung 17: Gleitender Mittelwert über 50 Iterationen der Wechselquote im Zürich-Szenario abhängig von der Nummer der Iteration für ChangeExpBeta = 0.8



Kriterium für das Nichterreichen des Equilibriums.

Vergleicht man die Abbildungen 14 und 16, so bemerkt man, dass für $\text{ChangeExpBeta}=0.8$ sich der Score weniger chaotisch verhält als für $\text{ChangeExpBeta}=0.2$. Das entspricht unseren Erwartungen: bei einem geringeren Anteil von umplanenden Agenten verhält sich das System stabiler.

Um das chaotische Systemverhalten einzudämmen, haben wir noch eine Simulation des Zürich-Szenarios laufen lassen, mit $\text{ChangeExpBeta}=0.8$ und mit drei Parameteränderungen im Modul „JDEQSim“.

```
<module name="JDEQSim">
    <param name="flowCapacityFactor" value="0.1" />
    <param name="storageCapacityFactor" value="1.0" />
    <param name="gapTravelSpeed" value="10.0" />
    <param name="squeezeTime" value="60" />
</module>
```

Der Parameter „flowCapacityFactor“ reguliert die Übersetzung der Belastung einer Strecke in den Fluss auf dieser Strecke und hat in den vorherigen Simulationen den Wert 1.0. Der Parameter „gapTravelSpeed“ bestimmt die „Reaktionszeit“ der Agenten, die nach einer Wartezeit weiterfahren dürfen. Dieser Parameter hat in den vorherigen Simulationen den Wert 15 m/s. Eine Verringerung dieses Parameters lässt die Agenten in kürzeren Zeiten zu ihren Zielen gelangen. Schließlich bestimmt der Parameter „squeezeTime“ die maximale Dauer, die ein Agent an einem Knoten warten muss, bis er in die nächste Kante einfahren kann. Dieser Parameter hat in den vorherigen Simulationen den Wert 1800 s. Eine Verringerung dieses Parameters ist also eine implizite Kapazitätserweiterung des Netzwerks. Insgesamt bewirken also die Verringerungen der obigen Parameter eine Verringerung von Staubildungen, welche erfahrungsgemäß für ein chaotisches Verhalten des Systems verantwortlich sind. In Abbildung 18 sieht man, dass durch diese Änderungen der Score als Funktion von der Nummer der Iteration stabiler geworden ist.

Man sieht in Abbildung 18, dass der Score ab etwa 200 Iterationen sich kaum noch ändert (Mittelwert=180.1, Standardabweichung=0.8). Der gleitende Mittelwert der Wechselwahrscheinlichkeit ist monoton fallend, erreicht aber selbst in den Iterationen 400 bis 500 nicht den Wert 0.01 (Mittelwert=0.017, Standardabweichung=0.0003, siehe Abbil-

Abbildung 18: Durchschnittlicher ausgeführter Score im Zürich-Szenario abhängig von der Nummer der Iteration (0 bis 500) für $\text{ChangeExpBeta} = 0.8$ in stabilem Modus

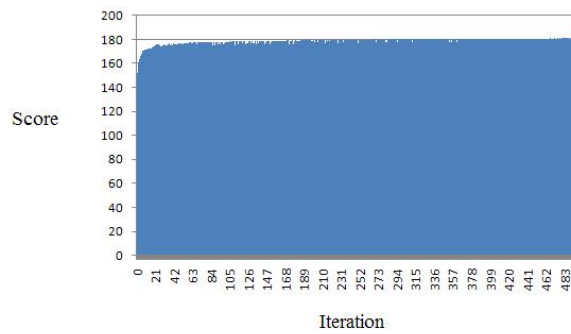
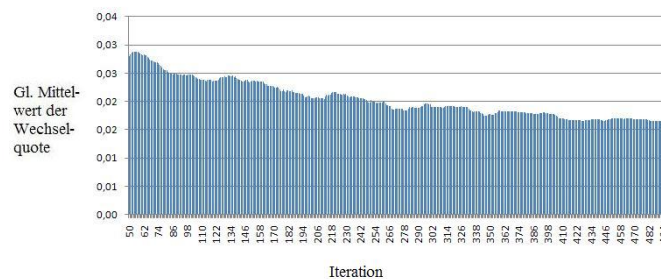


Abbildung 19: Gleitender Mittelwert über 50 Iterationen der Wechselquote im Zürich-Szenario abhängig von der Nummer der Iteration für $\text{ChangeExpBeta} = 0.8$ in stabilem Modus



dung 19). Eine mögliche Erklärung dafür ist die Folgende. Nach 500 Iterationen beträgt der Score 180.8, nach 1000 Iterationen (nicht in der Abbildung dargestellt) beträgt der Score 181.5. Man ist also in Equilibriumnähe, jedoch nicht im Equilibrium selbst. Auf diese Tatsache reagiert der gleitende Mittelwert der Wechselwahrscheinlichkeit sensibel, und bleibt trotz Equilibriumnähe deutlich über 0.01. Eine andere Erklärung für den hohen gleitende Mittelwert der Wechselwahrscheinlichkeit könnte sein, dass wir uns weder nach 500, noch nach 1000 Iterationen in Equilibriumnähe befinden (also dass der gleitende Mittelwert des Scores weiterhin streng monoton wachsend ist). Das könnte man überprüfen, indem man eine Simulation mit mehr Iterationen laufen lässt, und beobachtet, wie sich der Score und der gleitende Mittelwert der Wechselwahrscheinlichkeit verhält.

Insgesamt stellen wir also fest, dass, wie erwartet, eine Beziehung zwischen der Wechselwahrscheinlichkeit und der Equilibriumnähe des Scores besteht. Sicherlich zeugt eine allzugroße Wechselwahrscheinlichkeit davon, dass der Equilibriumscore nicht erreicht ist. Ein scharfes Kriterium für Equilibriumnähe jedoch haben wir unmittelbar ausschließlich aus der Wechselwahrscheinlichkeit nicht herleiten können, da sich die Beziehung zwischen der Wechselwahrscheinlichkeit und dem Score als stark abhängig vom simulierten Szenario ergeben hat. Tendenziell scheint jedoch ein gleitender Mittelwert der Wechselwahrscheinlichkeit, der mit einer kleinen Varianz um den Wert 0.01 streut, für Equilibriumnähe des Scores zu sprechen.

4. Fazit

In dieser Semesterarbeit haben wir Beschleunigungsmöglichkeiten der Konvergenz des agenten-basierten Simulationstools MATSim zu einem Systemequilibrium untersucht. Zum einen haben wir optimale Werte für die Parameter `SelectExpBeta`, `ReRoute` und `TimeAllocationMutator` aus dem Modul `strategies` herausgefunden. Zum anderen haben wir den Zusammenhang zwischen dem Wert der Wechselwahrscheinlichkeit im Planauswahlmodul `ChangeExpBeta` und der Nähe des durchschnittlichen ausgeführten Scores zum Equilibrium untersucht. Falls die Wechselwahrscheinlichkeit nahe am Wert 0.01 liegt, deutet das auf Nähe zum Equilibrium hin. Jedoch ist dieser Zusammenhang nicht eindeutig und kann allein, ohne weitere Indikatoren, nicht als Bedingung für Equilibriumnähe verwendet werden.

Literatur

- Balmer, M., A. Horni, K. Meister, F. Ciari, D. Charypar und K. W. Axhausen (2009) Wirkungen der Westumfahrung Zürich: Eine Analyse mit einer Agenten-basierten Mikrosimulation, *Endbericht*, Baudirektion Kanton Zürich, IVT, ETH Zürich, Zürich, Februar 2009, <http://www.ivt.ethz.ch/vpl/publications/reports/ab550.pdf>.
- Charypar, D., K. W. Axhausen und K. Nagel (2006) Implementing activity-based models: Accelerating the replanning process of agents using an evolution strategy, Paper Präsentation *the 11th International Conference on Travel Behaviour Research (IATBR)*, Kyoto, August 2006, <http://www.ivt.ethz.ch/vpl/publications/reports/ab387.pdf>.
- Liu, H. X., X. He und B. He (2009) Method of successive weighted averages (MS-WA) and self-regulated averaging schemes for solving stochastic user equilibrium problem, *Networks and Spatial Economics*, **9**, 485–503.
- Meister, K., M. Rieser, F. Ciari, A. Horni, M. Balmer und K. W. Axhausen (2009) Anwendung eines agentenbasierten Modells der Verkehrsnachfrage auf die Schweiz, *Straßenverkehrstechnik*, **53** (5) 269–280.
- Nagel, K. und G. Flötteröd (2009) Agent-based traffic assignment: Going from trips to behavioral travelers, Paper Präsentation *the 12th International Conference on Travel Behaviour Research (IATBR)*, Jaipur, Dezember 2009, <https://svn.vsp.tu-berlin.de/repos/public-svn/publications/vspwp/2009/09-14/>.
- Raney, B. und K. Nagel (2006) An improved framework for large-scale multi-agent simulations of travel behavior, in P. Rietveld, B. Jourquin und K. Westin (Hg.) *Towards better performing European Transportation Systems*, 305–347, Routledge, London.
- Zöllig, C. und K. W. Axhausen (2010) Calculating benefits of infrastructural investment, *Arbeitsbericht*, **612**, IVT, ETH Zürich, Zürich, <http://www.ivt.ethz.ch/vpl/publications/reports/ab612.pdf>.