

## **MATSim-T: Architektur und Rechenzeiten**

### **Michael Balmer**

IVT, ETH Zürich, CH-8093 Zürich, Tel: +41-44-633 27 80, Fax: +41-44-633 10 57, E-Mail: balmer (at) ivt.baug.ethz.ch

### **Marcel Rieser**

VSP, TU Berlin, D-10587 Berlin, Tel: +49-30-314 25 258, Fax: +49-30-314 26 269, E-Mail: rieser (at) vsp.tu-berlin.de

### **Konrad Meister**

IVT, ETH Zürich, CH-8093 Zürich, Tel: +41-44-633 30 88, Fax: +41-44-633 10 57, E-Mail: meister (at) ivt.baug.ethz.ch

### **David Charypar**

IVT, ETH Zürich, CH-8093 Zürich, Tel: +41-44-633 35 62, Fax: +41-44-633 10 57, E-Mail: charypar (at) ivt.baug.ethz.ch

### **Nicolas Lefebvre**

IVT, ETH Zürich, CH-8093 Zürich, Tel: +41-44-633 49 97, Fax: +41-44-633 10 57, E-Mail: lefebvre (at) ivt.baug.ethz.ch

### **Kai Nagel**

VSP, TU Berlin, D-10587 Berlin, Tel: +49-30-314 23 308, Fax: +49-30-314 26 269, E-Mail: nagel (at) vsp.tu-berlin.de

### **Kay W. Axhausen**

IVT, ETH Zürich, CH-8093 Zürich, Tel: +41-44-633 39 43, Fax: +41-44-633 10 57, E-Mail: axhausen (at) ivt.baug.ethz.ch

## **Kurzfassung**

Traditionell werden in der Verkehrsplanung Meso- und Makro-Simulationen angewandt. Dies hat verschiedene Gründe. Einerseits sind die Datengrundlagen typischerweise aggregierter Natur, wie zum Beispiel Strassenzählungen oder Pendlermatrizen, andererseits waren die Rechenleistungen ungenügend, um detaillierte und demzufolge speicher- und prozessorintensive Berechnungen durchzuführen. Diese Grenzen haben sich in den letzten Jahren auf imposante Weise verschoben. In der Informationstechnik ist kontinuierlich ein exponentielles Wachstum

der Rechenkapazität zu einem beliebigen festen Preis zu beobachten. Hand in Hand mit dieser Entwicklung sind die Fragestellungen und Datengrundlagen in der Verkehrsplanung komplexer und detaillierter geworden. Es entstehen somit ganz neue Ansprüche an die Verkehrsplanungssoftware. Derzeit gewinnt die Mikrosimulation der Verkehrsnachfrage immer mehr an Bedeutung. Die wichtigsten Gründe dafür sind:

- Verringerter Rechenaufwand und Speicherbedarf von grossen, mehrdimensionalen Wahrscheinlichkeitsmatrizen
- Variationsreichere Ausgabeoptionen, von aggregierten statistischen Analysen bis hin zu detaillierten Informationen über einzelne Individuen eines Szenarios
- Explizites Modellieren des Entscheidungsfindungsprozesses jedes einzelnen Individuums

In dieser Arbeit wird eine solche Agenten-Simulation für die Verkehrsplanung vorgestellt. Die Arbeit ist Teil des Forschungsprojektes MATSim-T (Multi-Agent Transport Simulation Toolkit, <http://www.matsim.org>) und konzentriert sich auf Design- und Implementationsfragen des Systems, sowie auf die Rechenzeiten der einzelnen Teile des Toolkits.

Anhand des Anwendungsfalls der gesamt-schweizerischen Nachfrage (ca. 2.3 Mio. Individuen für den motorisierten Individualverkehr mit gesamthaft etwa 7.1 Mio. Wege auf dem nationalen Netzmodell mit ca. 60'000 Kanten, vollständig zeit-dynamisch simuliert und optimiert für einen durchschnittlichen Werktag) wird gezeigt, dass MATSim-T den den Verkehr in ca. 36 Stunden modelliert.

## 1 Einleitung

Traditionell werden in der Verkehrsplanung Meso- und Makro-Simulationen angewandt [z.B. 1, 2]. Dies hat verschiedene Gründe. Einerseits sind die Datengrundlagen typischerweise aggregierter Natur, wie zum Beispiel Strassenverkehrszählungen oder Pendlermatrizen, andererseits waren die Rechenleistungen ungenügend, um detaillierte und demzufolge speicher- und prozessorintensive Berechnungen durchzuführen.

Diese Grenzen haben sich in den letzten Jahren auf imposante Weise verschoben. In der Informationstechnik ist kontinuierlich ein exponentielles Wachstum der Rechenkapazität zu einem beliebigen festen Preis zu beobachten. Die wichtigsten, für die Verkehrssimulation relevanten Entwicklungen sind:

- Das Speichervolumen der prozessornahen Speichereinheiten hat massiv zugenommen.
- Multi-Prozessor-Systeme erlauben es, parallel Berechnungen durchzuführen, ohne dabei auf (administrativ aufwändige) Rechen-Cluster angewiesen zu sein.

- Shared Memory-Architekturen erlauben es, dass mehrere Prozessoren auf einfache und schnelle Art und Weise auf den gleichen physischen Arbeitsspeicher zugreifen können.

Hand in Hand mit dieser Entwicklung sind die Fragestellungen und Datengrundlagen in der Verkehrsplanung komplexer und detaillierter geworden. Gute Beispiele dafür sind die vermehrt durchgeführten Umfragen zur Erstellung von Verkehrstagebüchern [3, 4, 5], wie auch die Verwendung der GPS-Technologie zur detaillierten Analyse des Verhaltens einzelner Verkehrsteilnehmer [6].

Es entstehen somit ganz neue Ansprüche an die Verkehrsplanungssoftware. Derzeit gewinnt die Mikrosimulation der Verkehrsnachfrage immer mehr an Bedeutung [7, 8, 9]. Die wichtigsten Gründe dafür sind:

- Verringerter Rechenaufwand und Speicherbedarf von grossen, mehrdimensionalen Wahrscheinlichkeitsmatrizen
- Variationsreichere Ausgabeoptionen, von aggregierten statistischen Analysen bis hin zu detaillierten Informationen über einzelne Individuen eines Szenarios
- Explizites Modellieren des Entscheidungsfindungsprozesses jedes einzelnen Individuums

Der letzte Punkt ist besonders wichtig. Denn es sind nicht die Fahrzeuge, die den Verkehr produzieren, sondern es sind die Personen, die sie benutzen. Personen produzieren Verkehr nicht einfach zum Selbstzweck. Vielmehr versuchen sie, ihren Tag (ihre Woche, ihr Leben) auf die bestmögliche Art und Weise zu organisieren. So gehen sie beispielsweise zur Arbeit um Geld zu verdienen. Sie besuchen Freunde und Verwandte zu ihrem Vergnügen oder weil sie sich dazu verpflichtet fühlen. Sie gehen Einkaufen um ihr Abendessen vorbereiten zu können und vieles mehr. Weil viele dieser Aktivitäten nicht am gleichen Ort erledigt werden können, müssen sie reisen, und produzieren damit Verkehr.

Für eine erfolgreiche Tagesplanung muss eine Person viele verschiedene Entscheidungen fällen. Sie muss entscheiden, wo sie eine bestimmte Aktivität ausführt, welches Verkehrsmittel sie benutzt, um von einem Ort zum nächsten zu gelangen, in welcher Reihenfolge und um welche Zeit sie ihren Aktivitäten nachgeht, mit wem sie gewisse Aktivitäten gemeinsam plant, etc.

Darüber hinaus ist noch eine Reihe von weiteren Entscheidungen zu fällen. Manche davon sind spontan und kurzfristig und entstehen typischerweise aufgrund unvorhergesehener Umstände. Andere Entscheidungen bedürfen einer langen Vorlaufzeit und sorgfältiger Planung. Ein weiterer wichtiger Faktor ist, dass viele Entscheidungen andere Entscheidungsfindungsprozesse induzieren.

Daher ist es äusserst wichtig, zur Modellierung des Verkehrsverhaltens den *gesamten* Planungshorizont einer Person zu betrachten und diesen in einem Simulationsmodell umzuset-

zen. Denn es sind der gesamte Tagesablauf einer Person und die Entscheidungen, die dahinter stecken, die den Verkehr produzieren.

In dieser Arbeit wird eine solche Agenten-Simulation für die Verkehrsplanung vorgestellt. Die Arbeit ist Teil des Forschungsprojektes MATSim-T (Multi-Agent Transport Simulation Toolkit, <http://www.matsim.org>) und konzentriert sich auf Design- und Implementationsfragen des Systems, sowie auf die Rechenzeiten der einzelnen Teile des Toolkits. Auf der Basis einer vollständig integrierten, individualisierten, täglichen Nachfrageoptimierung implementiert in MATSim-T wird aufgezeigt, wie das System erweitert wurde, damit es auf verschiedenste Arten von Input-Daten angewendet werden kann. Ausserdem wird dargelegt, wie auf einfache Art und Weise neue, auf die Input-Daten angepasste Modelle und Algorithmen hinzugefügt werden können. Weitere Funktionalitäten sind: Daten-Aggregationen und Disaggregationen für beliebige räumlichen Auflösungen, robuste Interfaces zu externen Modellen, Programmen und Systemen, Bearbeitung einer unbegrenzten Anzahl von Individuen und ein einfach benutzbares Interface um neue Datenschnittstellen zu implementieren.

### **1.1 Verwandte Arbeiten**

Grundsätzlich ist diese Idee nicht neu. Aktivitäten-basierte Nachfragemodellierung wurde schon mehrfach implementiert [e.g., 7, 8, 9, 2, 10, 11]. Jedoch wird typischerweise das Resultat in Form von (zeitabhängigen) Start-Ziel-Matrizen auf dynamische Umlegeverfahren angewandt.

Vollständig Agenten-basierte Mikrosimulationen, wie zum Beispiel mobiTopp [12] legen den Fokus auf die Telematik oder auf Auswirkungen von Angebotsveränderungen [13]. Ereignisorientierte Simulation des täglichen Verkehrsverhaltens [z.B. 14, 15, 16] zeigen die Möglichkeiten von Mikrosimulationen im Verkehrswesen auf, arbeiten aber meist auf kleinen Szenarien.

Ein wichtige Ausnahme davon ist TRANSIMS [17]. Während TRANSIMS flache Strukturen der Datenformate verwendet, basiert MATSim-T auf XML [18]. Dies erlaubt eine flexible Möglichkeit zur Erweiterung des Systems (das "X" steht für *eXtensible*), ohne strukturell Veränderungen der bestehenden Implementation vorzunehmen. Des Weiteren hält MATSim-T die Konsistenz der Aktivitätenketten entlang der Zeitachse aufrecht, so dass Agenten entsprechend Zeit an Aktivitäten verwenden müssen, bevor sie weiter Reisen dürfen. Auch der hier im Detail presentierte Rechenaufwand ist um einiges kleiner gegenüber TRANSIMS. Abschliessend ist noch zu erwähnen, dass MATSim-T flexibel und universell auf gegebene Daten in Qualität und Quantität angewendet werden kann.

Andere Entwicklungen sind die Agenten-basierten Modelle zur Landnutzung, so zum Beispiel URBANSIM [19], ILUTE [20] oder die Modelle von Abraham/Hunt [21].

### **1.2 Aufbau der Arbeit**

Da es sich bei MATSim-T um ein modulares System handelt, wird im Speziellen darauf eingegangen, wie sich bestehende und zukünftige Module auf die Funktionalität des Toolkits aus-

wirken. Es wird auch detailliert aufgezeigt, inwiefern sich die Module auf die Rechenzeit des Gesamtsystems auswirken. Wichtig dabei ist die Tatsache, dass durch den modularen Aufbau der Benutzer vollständige Kontrolle darüber erhält, wie er das Verhältnis zwischen Funktionalität, Detaillierungsgrad und Laufzeiten festlegen kann.

MATSim-T besteht im Kern aus zwei Teilen: Die *initiale individuelle Nachfrage-Modellierung* und die *iterative Nachfrage-Optimierung*. Letzteres liefert den relaxierten Zustand der Nachfrage. Dieser Teil wird in den folgenden Abschnitten vorgestellt. Abschnitt 7 beschäftigt sich mit der Erzeugung des Startzustands für den Optimierungsprozess. Abschliessend werden die momentanen Entwicklungsarbeiten vorgestellt, die das System in Grösse, Geschwindigkeit und Funktionalität erweitern.

Es ist zu erwähnen, dass das hier vorgestellte System unterdessen auf realen Planungsprojekten angewendet wird. Die Arbeit von Meister *et al.* [22], die ebenfalls im Rahmen der Heureka 2008 vorgestellt wird, präsentiert die ersten Resultate eines gesamt-schweizerischen Nachfrage-Modells, produziert mit dem hier vorgestellten Produkt. Aus diesem Grund wird hier der Anwendungsfall nicht weiter analysiert. Jedoch wird das Modell der gesamt-schweizerischen Nachfrage benutzt, um aufzuzeigen, wie sich die Rechenzeiten der einzelnen Bestandteile von MATSim-T aufteilen. Die Grössenordnungen sind die folgenden:

- Das schweizerische Planungsnetz [Nationales Netzmodell 23] besteht aus etwa 24'000 Knoten und 60'000 Kanten.
- Basierend auf den schweizerischen Betriebszählungsdaten vom Jahr 2000 [24] und der Volkszählung 2000 [25] werden ca. 1.7 Mio. Gebäude modelliert, inklusive Informationen über Aktivitäten, die pro Gebäude ausgeführt werden können. Es sind die Typen "Heim"-, "Arbeits"-, "Schul"- "Einkaufs"- und "Freizeit"-Aktivität definiert.
- Wiederum aus der Volkszählung 2000 und dem Mikrozensus 2005 [26] konnten ca. 7 Mio. synthetische Personen generiert werden, inkl. demographischen Attributen wie Alter, Geschlecht, Führerscheinbesitz, Fahrzeugverfügbarkeit, ÖV-Zeitkarten und Arbeitsverhältnis.
- Die Modellierung der individuellen, zeitlich dynamischen, täglichen Nachfrage wird in Ciari *et al.* [27] und Meister *et al.* [22] im Detail beschreiben. Die gesamt-Schweizerische Nachfrage beinhaltet etwa 22 Mio. Wege, die Anzahl MIV-Wege beläuft sich auf ~7.1 Mio.
- Alle Berechnungen in dieser Arbeit wurden auf einem Rechner durchgeführt mit 8 Dual-Core Prozessoren und je 2.2 GHz Taktfrequenz.
- Der benötigte Arbeitsspeicher für diesen Anwendungsfall beträgt 22 GByte.

```
<plans name="example plans file">
...
<person id="393241" sex="f" age="27" license="yes" car_avail="always"
  employed="yes">
  <travelcard type="regional-abo" />
  <plan>
    <act type="home" link="58" start_time="00:00" dur="07:00"
      end_time="07:00" />
    <leg mode="car" dept_time="07:00" trav_time="00:25" arr_time="07:25">
      <route>1932 1933 1934 1947</route>
    </leg>
    <act type="work" link="844" start_time="07:25" dur="09:00"
      end_time="16:25" />
    <leg mode="car" dept_time="16:25" trav_time="00:14" arr_time="16:39">
      <route>1934 1933</route>
    </leg>
    <act type="home" link="58" start_time="16:39" dur="07:21"
      end_time="24:00" />
  </plan>
</person>
...
</plans>
```

**Abbildung 1: Beschreibung der Nachfrage einer synthetischen Person (inkl. demographischen Angaben) für den gesamten Tag. Dieser *Agent*, ID 393241 will um 07:00 sein Zuhause (an Strecke Nummer 5834) verlassen um zur Arbeit zu fahren. Dazu benutzt er eine Route über 4 Knoten (5 Strecken) mit erwarteter Reisezeit von 25 Minuten. Der Agent verweilt für 9 Stunden bei der Arbeit und fährt danach zurück nach Hause (mit erwarteter Reisezeit von 14 Minuten). Die Nachfrage beschreibt somit nicht nur einzelne Teile eines Tages, sondern die *gesamte* Abfolge des Agenten Nummer 393241, kontinuierlich in der Zeit. - Quelle: Balmer [28]**

## 2 Iterative Nachfrage-Optimierung

Der Begriff *Multi-Agenten Mikrosimulation* wird in der Verkehrsplanung auf verschiedene Art und Weise interpretiert. So zum Beispiel ist mit dem Wort mikroskopisch oft das Fahrzeugfolgemodell [29] gemeint, das in einigen Produkten [z.B. VISSIM 2] zur Anwendung kommt. In MATSim-T bezieht sich dieser Begriff auf das vollständig individualisierte Modellieren jeder einzelnen Person. Jede Person wird als *Agent* modelliert, sodass die Summe der Agenten die statistisch repräsentative Demographie der zu analysierenden Region darstellt. Zudem wird die Nachfrage für jeden einzelnen Agenten individuell modelliert und optimiert. Dies nicht nur für einzelne Teile der Nachfrage — wie zum Beispiel Ziel- oder Routenwahl — sondern als komplett *zeitlich dynamische* Beschreibung einer gesamten Tagesnachfrage jedes Agenten.

Diese Nachfrage wird in MATSim-T als *Tagesplan* bezeichnet. Abbildung 1 zeigt die Beschreibung des Tagesplanes eines Agenten in XML-Format [18]. Das Wichtigste dabei ist nun, dass diese Struktur während der gesamten Modellierung und Optimierung der Nachfrage einer Region nicht auseinandergebrochen wird. Das heisst, dass auch bei der Umlegung der Verkehrs-

nachfrage die Wege nicht einzeln betrachtet werden, sondern der gesamte Tag zeitlich dynamisch umgesetzt wird, inkl. der Ausführung der Aktivitäten. Ergo bezieht sich das Wort *Mikrosimulation* auf die mikroskopische (individuelle) Nachfrage jeder Person im Untersuchungsgebiet.

## 2.1 Der koevolutionäre Algorithmus von MATSim-T

Da die Erzeugung der individuellen Nachfrage in Abbildung 1 Informationen enthält, die sehr schwierig oder gar unmöglich anhand von Erhebungen und Umfragen zu modellieren ist — ein gutes Beispiel dafür ist die Routenwahl — braucht es einen Mechanismus, der diese ungenauen oder fehlenden Angaben berechnet und ein realistisches Abbild liefert. Es bedarf eines Ausgleichsverfahrens, dessen Ziel es ist, einen Gleichgewichtszustand des Untersuchungsgebietes zu finden. Das Wardrop Equilibrium [30] beschreibt einen solchen Gleichgewichtszustand für die Routenwahl.

In MATSim-T ist es aber nicht allein die Routenwahl, die zu optimieren ist, es ist der gesamte Tag (Routen, Zeiten, Orte, Reihenfolge der Aktivitäten, Art der Aktivitäten, usw.) jedes Agenten, der möglichst erfolgreich (mit möglichst hohem Nutzen) zu absolvieren ist. Dabei ist der Erfolg eines Tagesplans abhängig von den infrastrukturellen Randbedingungen (Kapazitäten der Strassen, Öffnungszeiten von Läden, usw.) und von den Plänen der anderen Agenten im System. Das heisst, der effektive Nutzen einer Planung kann nur im Zusammenspiel aller Agenten ermittelt werden. Hier kommt das Optimierungsverfahren von *koevolutionären Algorithmen* [31, 32] ins Spiel, das im Optimierungsverfahren von MATSim-T implementiert ist.

Im Wesentlichen besteht ein koevolutionärer Algorithmus aus folgenden Schritten:

1. initialisiere  $P(t=0)$  — Generation der Individuen zur Zeit  $t=0$
2. bewerte  $P(t)$  — Berechne die "Fitness"
3. selektiere  $P'(t)$  aus  $P(t)$  — "survival of the fittest"
4. rekombiniere und mutiere  $P'(t)$  — "Crossover" und "Mutation"
5.  $P(t+1) = P'(t)$ ;  $t = t+1$  — nächste Generation von Individuen
6. Zurück zu Punkt 2

Angewandt auf die Optimierung der Nachfrage in MATSim-T, d.h. die Optimierung der Tagespläne, heisst das:

1. Initialisiere Tagespläne pro Agent (Generation von Plänen), für alle Agenten im System
2. Berechne und bewerte die Umsetzung von individuellen Tagesplänen
3. Lösche "schlechte" Tagespläne (Pläne mit tiefem Nutzen)



**Abbildung 2: Schematischer Aufbau des evolutionären Algorithmus von MATSim-T**

4. Dupliziere und variiere Tagespläne
5. Erstellung der nächsten Generation von Tagesplänen ist abgeschlossen
6. Zurück zu Punkt 2

Die Abbildung 2 zeigt den schematischen Aufbau des Optimierungsverfahrens. Für jeden der oben aufgeführten Schritte sind einzelne Module verantwortlich. Die Umsetzung (*EXEC*) übernimmt ein entsprechendes *Verkehrsfluss*-Modul, indem die einzelnen Individuen interagieren. Zum Beispiel produzieren sie Stau auf gemeinsam benutzten Strassenabschnitten. Das *SCORING* Modul bewertet daraufhin die Ausführung der umgesetzten Tagespläne. Bewertete Pläne mit hohem Nutzen (hohe "Fitness") überleben, während Pläne mit schlechtem Nutzen (z.B. lange Stauzeiten) gelöscht werden.

Die Erzeugung und Variation eines neuen Tagesplans (*REPLANNING*) übernehmen verschiedene Module welche sich auf Veränderung einzelner Teile eines Tagesplans konzentrieren. Diese Planungsänderungen eines Agenten berechnen sich völlig unabhängig von denen der anderen Agenten im System. Die Replanning-Module beziehen sich dabei einzig auf Informationen der letzten Verkehrsflusssimulation. Im Folgenden wird auf die einzelnen Module, deren Umsetzung und auf die Berechnungszeiten eingegangen.

### 3 Verkehrsflusssimulation

Die Verkehrsflusssimulation hat die Aufgabe, die Tagespläne in einer physikalischen Umgebung umzusetzen. Grundsätzlich sind beliebig detaillierte Modelle anwendbar, wie zum Beispiel das Fahrzeugfolgemodell [29] oder ein Zellulärer Automat von Nagel und Schreckenberg [33], die aber entsprechend grosse Rechenzeiten benötigen. Die Ansprüche der Verkehrsplanung sind jedoch nicht das detaillierte Fahrverhalten, sondern eine realitätsnahe Abbildung des dynamischen Verkehrsaufkommen, der Staubildung und des daraus entstehenden Rückstaus, die Stauauflösung, usw. Diese Anforderungen erfüllt das Warteschlangen Modell [34]. Dabei wird jede Strasse durch eine Warteschlange modelliert, in der Fahrzeuge mindestens die benötigte Fahrzeit auf freier Strasse abwarten müssen.

MATSim-T stellt zwei Implementierungen dieses Modells zur Verfügung, die für die realistische Umsetzung der Nachfrage notwendigen Anforderungen erfüllen. Sie beachten die Randbedin-



gungen des Verkehrs (Kapazität einer Strasse, Höchstgeschwindigkeit, etc.), der infrastrukturellen Randbedingungen (z.B. Öffnungszeiten einer Aktivität), wie auch den physikalisch machbaren Rahmen. Es ist einem Agenten zum Beispiel nicht gestattet, eine Aktivität zu verlassen, bevor er dort angekommen ist.

Als Resultat produziert die Simulation für jeden Agent, zu jedem Zeitpunkt des gesamten Tages Informationen darüber, wo dieser gerade ist, und was er gerade macht. Anders formuliert heisst dass, jeder Agent beschreibt für jede Aktion (Start/Ende einer Aktivität, Einbiegen/Verlassen einer Strasse, etc.) ein zeitlich und örtlich lokalisiertes *Event*.

### 3.1 Mobility Simulation

Die *Mobility Simulation* ist eine Re-Implementierung der “Ein-Prozessor Variante” der Arbeit vorgestellt in Cetin [35] als integraler Bestandteil des Toolkits. Das heisst, es ist — ebenso wie das gesamte Projekt — in der Programmiersprache *JAVA* implementiert. Dies hat zum Vorteil, das die *Mobility Simulation* direkt auf jeden Datenpunkt der Objektdatenbank von MATSim-T zugreifen kann und damit zeit-intensiven Input und Output eingespart werden kann. Zudem ist *JAVA* plattform-unabhängig und somit anwendbar auf Windows, Mac und Linux/Unix.

Die *Mobility Simulation* arbeitet auf Sekundenbasis. Das heisst, zu jeder simulierten Sekunde des gesamten Tages werden Operationen ausgeführt, um den Verkehrsfluss abzubilden. Bei dieser zeitschritt-basierten Mikrosimulation wird jede Warteschlange synchron ein neuer Zustand zugewiesen. Demzufolge ist die Laufzeit proportional zu den Anzahl Strassen des Verkehrssystems. Zudem wird eine weitere Warteschlange pro Strecke modelliert, die die Aufgabe hat, Agenten von einer abgeschlossenen Aktivität auf das Strassennetz zu führen. Daraus folgt für die Rechenzeit

$$T_{mobsim} \propto \frac{t_{sim}}{\Delta t} \cdot 2 \cdot s, \quad (1)$$

wobei  $t_{sim}$  der zu simulierende Zeitsraum ( $1Tag = 86'400Sek.$ ),  $\Delta t$  der Zeitschritt der *Mobility Simulation* (1 Sek.) und  $s$  die Anzahl Strassen des Strassennetzes ist. Zu erwähnen ist, dass die Rechenzeit nicht von der Anzahl der Agenten — resp. deren Nachfrage — im System abhängt. Dies ist sicher günstig für ein stark ausgelastetes Netz (z.B. die Morgenspitze), jedoch benötigt die *Mobility Simulation* die gleiche Rechenzeit für einen schwach ausgeprägten Nachtverkehr. Für die *Mobility Simulation* angewandt auf die gesamt-schweizerischen Nachfrage (Abschnitt 1.2) bedeutet dies:

$$T_{mobsim}^{Schweiz} \propto \frac{86'400}{1} \cdot 2 \cdot 60'000. \quad (2)$$

Es sind somit 10.368 Milliarden Warteschlangen-Operationen auszuführen. Die Gesamt-rechenzeit auf auf einem 2.2 GHz Prozessor beläuft sich auf ca. 70 Minuten.

### 3.2 Deterministische, event-basierte Warteschlangen-Simulation (DEQSim)

Eine weitere Verkehrsflusssimulation, im Detail beschrieben in Charypar *et al.* [36] benutzt eine erweiterte Form des Warteschlangen-Modells, das zusätzlich zum FIFO (First-In, First-Out) Verhalten eine sich rückwärts bewegende Lücke simuliert, die das Rückstauverhalten noch realistischer abbildet. Auch die Implementierung unterscheidet sich von der Mobility Simulation. Einfach ausgedrückt operiert das System nur dort, wo auch wirklich etwas passiert. Das heisst, nur dann, wann Agenten *events* produzieren, wird Rechenzeit verwendet. Demzufolge ist die Rechenzeit

$$T_{deqsim} \propto e(a, s), \quad (3)$$

wobei die Anzahl events  $e$  abhängig ist von der Anzahl Agenten ( $a$ ), respektive deren Tagesplänen und der Auflösung des Strassennetzes. Letzteres kommt daher, dass bei hoch aufgelösten Netzen die Anzahl Strassenabschnitte, die ein Agent während einer Fahrt passiert, ansteigt und damit auch die Anzahl Events erhöht wird.

Für die gesamt-schweizerische Nachfrage werden ca. 162 Mio. Events produziert. Die daraus resultierende Gesamt-Rechenzeit der DEQSim im Ein-Prozessor Modus beläuft sich auf ca. 50 Minuten. Die DEQSim ist zudem auch als Multi-Prozessor Programm benutzbar. Die Zeiteinsparung steigt in etwa linear mit der Anzahl benutzter Prozessoren. Bei 8 Prozessoren kommt das Modul somit auf eine Laufzeit von ca. 8 Min.

Es ist aber zu vermerken, dass die DEQSim in der Programmiersprache *C++* geschrieben ist. Das bedeutet, dass die benötigten Daten via Dateien ausgeschrieben werden müssen. Zudem müssen nach der Simulation die Resultate wiederum in MATSim-T eingelesen werden. Dies benötigt in etwa 20 Min. zusätzlicher Rechenzeit.

## 4 Bewertung (Scoring)

Nach Beendigung der Verkehrsflusssimulation ist es nun möglich, den effektiven Nutzen eines Tagesplans zu bestimmen. Denn erst durch die Interaktion mit anderen Agenten zeigt sich, wie realistisch die Planung des Tages eines Agenten ist. Der Erfolg eines Tages wird durch eine individuelle Nutzenfunktion bestimmt. Sie beschreibt das anzustrebende Ziel jedes einzelnen Agenten, und somit auch sein *Verhalten*. Prinzipiell kann eine beliebige Bewertungsmethode herangezogen werden, zum Beispiel die *prospect theory* [37]. Charypar und Nagel [38] beschreibt die Nutzenfunktion, die zur Zeit in Anwendung ist. Diese orientiert sich an Vickrey's Modell der Wahl der Abfahrtszeit [39, 40], wird jedoch von einzelnen Wegen auf unser Konzept der Verkehrsnachfrage mit kompletten Tagesplänen erweitert [38, 41]. Der Ansatz funktioniert wie folgt:

Der Gesamtwert eines Plans ergibt sich aus der Summe der Werte seiner Bestandteile:

$$U_{total} = \sum_{i=1}^n U_{perf,i} + \sum_{i=1}^n U_{late,i} + \sum_{i=1}^n U_{travel,i}, \quad (4)$$

wobei  $U_{total}$  der Gesamtnutzen eines Plans,  $n$  die Anzahl Aktivitäten (entspricht Anzahl Wege: ein Aufenthalt zuhause wird immer als erste Aktivität angenommen),  $U_{perf,i}$  der (positive) Nutzen des Durchführens von Aktivität  $i$ ,  $U_{late,i}$  der (negative) Nutzen durch Verspätung bei Aktivität  $i$  und  $U_{travel,i}$  der (negative) Nutzen des Reisens auf Weg  $i$  ist. Für den (in der Regel positiven) Nutzen des Durchführens einer Aktivität wird eine logarithmische Form angenommen:

$$U_{perf,i}(t_{perf,i}) = \beta_{perf,i} \cdot t_{*,i} \cdot \ln\left(\frac{t_{perf,i}}{t_{0,i}}\right), \quad (5)$$

wobei  $t_{perf,i}$  die tatsächliche Dauer der Durchführung der Aktivität  $i$  und  $t_{*,i}$  die (individuell) angestrebte Dauer von Aktivität  $i$  ist.  $\beta_{perf,i}$  definiert den Grenznutzen von Aktivität  $i$  bei  $t_{*,i}$ . Dieser Parameter ist für alle Aktivitäten gleich. Im Gleichgewicht haben somit alle Aktivitäten den gleichen Grenznutzen, und damit (in Abwesenheit zeitlicher Beschränkungen und ohne Berücksichtigung der Reisezeit) die Gesamtnutzenfunktion ihr Maximum. Dies entspricht der natürlich-logarithmischen Transformation einer Cobb-Douglas-Nutzenfunktion für normale und streng monotone Präferenzen.

$t_{0,i}$  beeinflusst sowohl die Minimaldauer als auch die Wichtigkeit einer Aktivität. Wenn  $t_{perf,i} < t_{0,i}$ , dann  $U_{perf,i} < 0$  ist, sollte der Agent diese Aktivität komplett aus seinem Plan nehmen und die Zeit in lohnendere Aktivitäten investieren. Ist  $t_{0,i}$  nur minimal kleiner als  $t_{*,i}$ , steigt der Grenznutzen rapide mit sinkender Durchführungsdauer. In diesem Fall lohnt es sich andere Aktivitäten zu opfern, um dieses kleine Zeitfenster einzuhalten. Definiert ist  $t_{0,i}$  als

$$t_{0,i} = t_{*,i} \cdot \exp\left(\frac{-A}{t_{*,i}}\right), \quad (6)$$

wobei  $A$  ein Skalierungsfaktor ist. Unter der Voraussetzung, dass in der Anwendung den Agenten nicht die Möglichkeit geboten wird, Aktivitäten aus der Nachfrage zu löschen oder hinzuzufügen, ist das Nutzenniveau einer Aktivität unabhängig vom Typ:

$$U_{perf,i}(t_{*,i}) = \beta_{perf,i} \cdot A, \quad (7)$$

Des weiteren beträgt der (in der Regel negative) Nutzen des Zu-Spät-Kommens von Formel 4:

$$U_{late,i} = \beta_{late} \cdot t_{late,i}, \quad (8)$$

mit dem Grenznutzen des Zu-Spät-Kommens  $\beta_{late}$  und  $t_{late,i}$  für die Verspätung bei Aktivität  $i$ .

Schlussendlich wird der (in der Regel negative) Nutzen des Reisens als

$$U_{travel,i} = \beta_{travel} \cdot t_{travel,i} \quad (9)$$

definiert, mit  $\beta_{late}$  als Grenznutzen des Reisens und  $t_{late,i}$  als Reisezeit für Weg  $i$ . Denkbar sind weitere Bestrafungen, etwa für Wartezeit oder zu frühes Verlassen einer Aktivität mit festem Endzeitpunkt. Es ist jedoch nicht unbedingt eine direkte Strafe nötig, da die Zeit nutzlosen Wartens schon für andere Aktivitäten verloren ist, womit der Nutzen sinkt (Opportunitätskosten), d.h. der effektive Grenznutzen des Wartens beträgt  $-\beta_{perf}$ . Diese Grenzkosten fallen im Übrigen auch bei der Bewertung der Reisezeit an: Der effektive Grenznutzen des Reisens beträgt  $-|\beta_{travel}| - \beta_{perf}$ . Bei Verspätungen fallen keine Opportunitätskosten an, da die Verspätungszeit schon woanders verbracht und dementsprechend bewertet wurde. Der effektive Grenznutzen von Verspätungen bleibt daher bei  $\beta_{late}$ .

Diese Werte  $\beta_{perf}$ ,  $\beta_{perf} + |\beta_{travel}|$  und  $|\beta_{late}|$  entsprächen den Werten des Vickrey-Modells, würde MATSim nur Verspätungen einzelner Wege berücksichtigen.

Die Bewertungsfunktion beschreibt das Verhalten des Agenten, denn dieser sucht im Lösungsraum der Bewertungsfunktion nach der best-möglichen Bewertung, und somit nach dem best-möglichen Tagesplan. Ausserhalb des Lösungsraums von Formel 4 ist es dem Agenten nicht möglich zu optimieren. Dieser Aspekt wird im Abschnitt 6 detailliert erklärt.

Die Rechenzeit zur Ermittlung des Nutzens eines umgesetzten Tagesplanes ist vernachlässigbar klein und fällt somit auch nicht bei 7 Mio. Agenten für die Gesamt-Schweizerische Nachfrage ins Gewicht.

#### 4.1 “Survival of the fittest”

Ein kleiner, aber wichtiger Prozessschritt ist das Löschen eines “schlechten” Plans. Da in jeder Iteration für einen Teil der Agenten neue Pläne erzeugt werden, füllt sich diese *Population von Plänen* pro Agent bis zu einem, vom Anwender definierten Maximum (typischerweise 3 bis 6 Pläne pro Agent). Bevor man nun einen neuen Plan erstellen darf, wird der schlechteste (derjenige mit tiefstem Gesamtnutzen) aus dieser Population gelöscht. Somit “überleben” nur die guten Pläne. Selbstverständlich kostet dieser Schritt kaum Rechenzeit.

## 5 Variation (Replanning)

Die Variation (in MATSim-T genannt *Replanning*) sorgt dafür, dass jeder Agent seinen Lösungsraum der Tagesplanung “erforscht”. Dabei wird ein bestehender Plan eines Agenten dupliziert und variiert (mutiert) und in der nächsten Iteration wiederum ausgeführt und bewertet. Jedes verwendete Modul des Optimierungsprozesses übernimmt spezielle Aufgaben. So zum Beispiel berechnet das *Router* Modul die einzelnen Wege eines Planes aufgrund des Verkehrsaufkommens der letzten Verkehrsflusssimulation. Abfahrtszeiten und Dauern von Aktivitäten

eines Tagesplanes werden durch das *Time Allocation Mutator* Modul verändert. Es mutiert entsprechende Zeiten nach einem Zufallsprinzip. Weitere Module können neue Orte bestimmen, wo Aktivitäten ausgeführt werden, oder auch die Reihenfolge von Aktivitäten verändern. Wichtig dabei ist, dass all diese Module *unabhängig* voneinander arbeiten. Aus diesem Grund ist es möglich, eine beliebige Anzahl von Replanning Modulen in die Optimierung einzufügen.

Ein äusserst wichtiger Aspekt dabei ist dabei, ob ein Modul einen bestimmten Teil eines Planes rein zufällig verändert (*Random Mutation*) oder ob es die best möglich Lösung aufgrund der Resultate der letzten Verkehrsflusssimulation ermittelt (*Best Response*). Ersteres hat den Vorteil, kaum Rechenzeit zu verbrauchen. Zudem sucht es — früher oder später — den ganzen ihm zugetragenen Suchraum ab. Der Nachteil dabei ist, dass es (zu) viele Iterationen braucht, bis das Optimierungssystem relaxiert.

Im Gegensatz dazu liefert ein *Best Response* Modul eine optimale Lösung aufgrund des momentanen Wissens um das System. Diese Module können sehr rechenintensiv werden, relaxieren das System aber weit aus schneller.

### 5.1 Time Allocation Mutator

Das Time Allocation Mutator Modul ist ein typisches Beispiel für ein *Random Mutation* Modul. Es variiert zufällig die Abfahrtszeiten und Dauern der Aktivitäten eines Tagesplanes. Für das mutieren der Zeiten der gesamt-schweizerischen Nachfrage benötigt es ein paar Sekunden und ist demzufolge für die Gesamtrechenzeit vernachlässigbar.

### 5.2 Router Module

Router Module berechnen Aufgrund gegebener Abfahrtszeit der Wege in einem Tagesplan und der dynamischen Reisezeit *aller* Strassen (erhalten von der letzten Ausführung der Verkehrsflusssimulation) die best mögliche Route (Route mit geringst möglichen negativen Nutzen). Dieses *Best Response* Module benutzt somit *globales Wissen* des Systems zur Ermittlung der Route.

MATSim-T beinhaltet zur Zeit drei verschiedene Implementationen des Router Moduls, alle basierend auf dem zeit-dynamischen Dijkstra Algorithmus und liefern entsprechend identische Resultate. Die Laufzeit steigt quadratisch mit der Anzahl Knoten im Netzwerk und linear mit der Anzahl von Wegen in einem Plan. Die neuste Implementation [Landmarks-A\* Module, siehe 42] liefert im Mittel die beste Performanz: Für die Gesamt-schweizerische Nachfrage benötigt es im Schnitt 0.1 Millisekunden zur Berechnung eines Weges.

### 5.3 “planomat”

Ein weiteres *Best Response* Modul ist der *planomat* wie beschrieben in Meister [43]. Dieses Modul optimiert nicht nur einen bestimmten Aspekt eines Planes, sondern alle Teile gleichzeitig. Es stützt sich wiederum auf die Informationen, die die letzte Ausführung der Nachfrage

(Abschnitt 3) liefert. Zudem liefert es für alle Agenten des gleichen Haushaltes einen entsprechenden Tagesplan, sodass auch gemeinsame Aktivitäten (z.B. das gemeinsame Abendessen zu Hause) zeitlich synchronisiert wird.

Eine vereinfachte Version des *planomat*'s ist Teil des Toolkits und optimiert die Zeitplanung. Es ist somit ein Substitut des *Time Allocation Mutator*, beschrieben in Abschnitt 5.1. Der *planomat* benutzt zur Zeitoptimierung wiederum einen evolutionären Algorithmus und ist somit ein weitaus zeitintensiverer Prozess als das *Time Allocation Mutator* Modul. Im Durchschnitt beläuft sich die Rechenzeit zur Zeitoptimierung eines Planes für den Anwendungsfall auf ca. 5.7 Millisekunden.

#### 5.4 Weitere Module

Auf Grund der in den letzten Jahren erreichten Flexibilität und Robustheit von MATSim-T sind weitere Module hinzugekommen, manche mit Potential zur Generalisierbarkeit und demzufolge als funktionale Erweiterung von MATSim-T zu sehen, manche aber auch als Experimente zur Erforschung spezieller Fragestellungen. Zur Zeit in Entwicklung sind Module zur Wahl von Nebenaktivitäten eines Tages (z.B. Einkaufen oder Freizeit) implementiert als einfache Suche im Lösungsraum aller in der Region vorhandenen Möglichkeiten, wie auch im Zusammenhang mit sozialen Strukturen zum Austausch von Informationen [44]. Im Rahmen einer Studienarbeit [45] wurde ein weiteres Routing Modul erstellt, um fahrplanfeine Routensuche basierend auf dem Netz der Berliner Verkehrsbetriebe (BVG) durchzuführen. Dieses Modul ist somit nur in bestimmten Szenarien einsetzbar, zeigt aber die Flexibilität von MATSim-T.

Im Weiteren existieren Module, die die Simulation von "Road Pricing"-Szenarien in MATSim-T ermöglichen [46]. Dabei werden bestehende Komponenten von MATSim-T erweitert, beispielsweise die Nutzenfunktion (Berücksichtigung bezahlter Maut-Abgaben während der Simulation) oder der Routing-Algorithmus (Berücksichtigung von bemauteuten Strecken während der Routensuche).

In dieser Arbeit wird jedoch nicht weiter auf diese Module eingegangen, da sie zur Zeit noch im Rahmen von Forschungsarbeiten weiterentwickelt und validiert werden.

## 6 Relaxation des EA

Entsprechend den Bedürfnissen des Nutzers, ist es nun möglich eine Kombination all der oben erwähnten Module zusammen zu stellen. Der Optimierungsprozess — das heisst das Iterative Ausführen der einzelnen Prozessschritte — übernimmt dann das Toolkit. Jedoch ist bei der Zusammenstellung ein Aspekt dringend zu beachten: Jedes hinzugefügte Replanning Modul erweitert den Lösungsraum für den Tagesplan eines Agenten. Dieser Lösungsraum muss zwingend *vollständig* durch die Nutzenfunktion abgedeckt werden. Hierzu ein Beispiel:

Erlaubt man einem Agent nur die Routenwahl zu optimieren, reicht es durchaus aus, die Nut-

zenfunktion aus Abschnitt 4 zu reduzieren auf  $U_{total} = \sum_{i=1}^n U_{travel,i}$ , da der Agent keinerlei Möglichkeiten hat, an seiner Zeitplanung etwas zu verändern. Fügt man aber zusätzlich zur Routenwahl ein Zeitwahl Modul für den Agenten hinzu, muss diese Erweiterung des Lösungsraums auch in die Nutzenfunktion, und somit in das Verhaltensmodell des Agenten mit einfließen ( $U_{total} = \sum_{i=1}^n U_{perf,i} + \sum_{i=1}^n U_{travel,i}$ ). Andererseits ist es legitim, die erweiterte Nutzenfunktion zur Ermittlung des Optimums für Agenten mit alleiniger Routenwahl zu benutzen, da auch diese den gesamten Lösungsraum abdeckt. Aus diesem Grund geht die funktionale Weiterentwicklung des Optimierungsprozesses in MATSim-T (Implementierung neuer Replanning Module) Hand in Hand mit der Erweiterung des Verhaltensmodells der Agenten.

Im Folgenden wird das Relaxationsverhalten, wie auch die dazu benötigte Rechenzeit des koevolutionären Algorithmus analysiert.

## 6.1 Setup

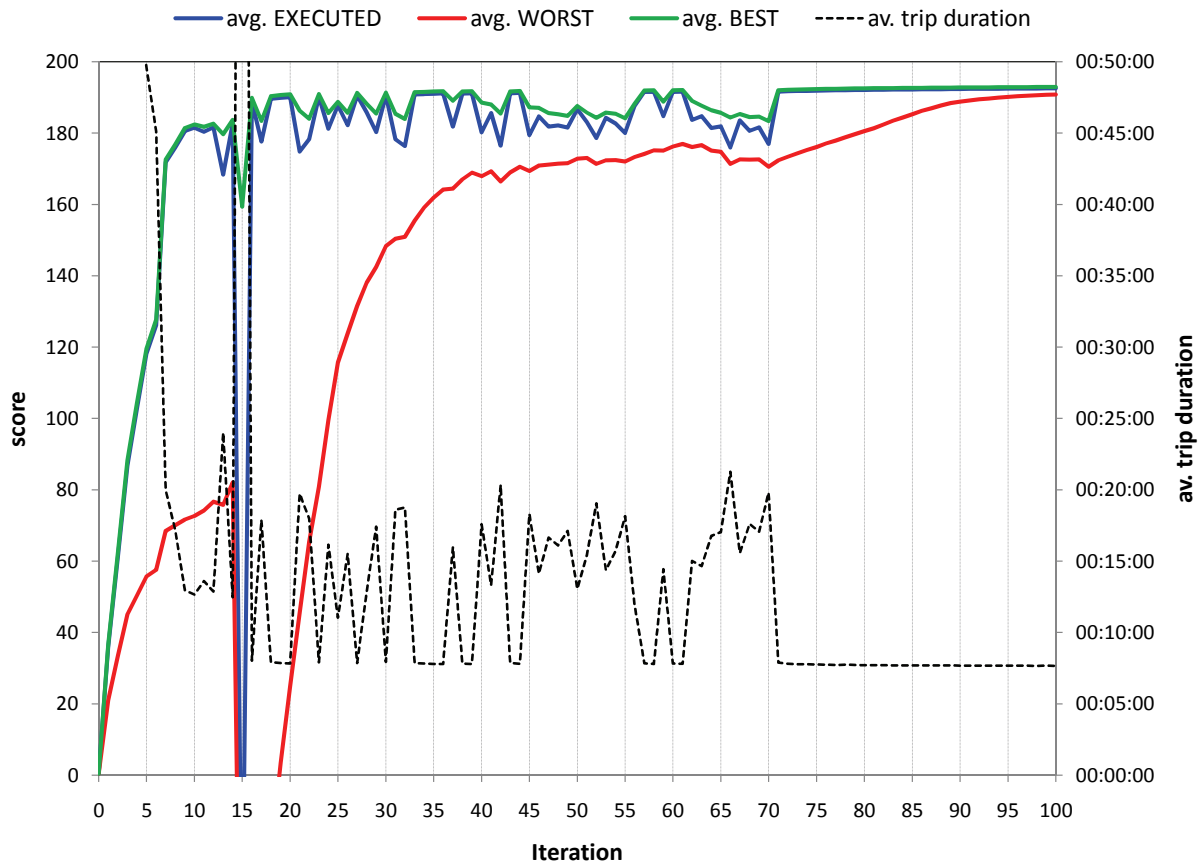
Um den Relaxationsprozess entsprechend zu analysieren, wird zunächst ein typisches, in den letzten Jahren mehrfach angewandtes Setup definiert:

1. Jeder Agent hat die Möglichkeit zur Routenwahl.
2. Jeder Agent hat die Möglichkeit zur Zeitwahl.
3. Pro Iteration werden zufällig 10% der Agenten ausgewählt, die durch erneute Routenwahl eines bestehenden Planes einen neuen Plan erstellen.
4. Pro Iteration werden zufällig 10% der Agenten ausgewählt, die durch erneute Zeitwahl eines bestehenden Planes einen neuen Plan erstellen.
5. Die restlichen 80% der Agenten wählen einen bestehenden Plan zur erneuten Ausführung aus. Die Auswahlwahrscheinlichkeit entspricht der Logit-Funktion  $p \sim \exp(\beta \cdot S_i)$ , wobei  $S_i$  der Nutzenwert von Plan  $i$  ist und  $\beta$  eine empirische Konstante.
6. Die Nutzenfunktion für jeden Agenten entspricht demjenigen aus Abschnitt 4.
7. Die Population von Plänen jedes Agenten ist begrenzt auf ein Maximum von 4 Plänen.
8. Das System wird als relaxiert beurteilt, sobald die Reihe der durchschnittlichen Nutzenwerte aller vorhandener Pläne ein stationärer Prozess über die Iterationen ist.

Eine detailliertere Beschreibung des Setups und der Parameterwerte der Nutzenfunktion ist in Meister *et al.* [22] zu finden.

## 6.2 Relaxation

Der relaxierte Zustand des koevolutionären Algorithmus von MATSim-T ist dann erreicht, wenn durch Variation der Tagespläne der Nutzen für jeden Agent nicht mehr merklich verändert wer-



**Abbildung 3: Mittlerer Nutzen und Weg-Reisezeit der Tagespläne**

den kann. Da bei jeder Iteration die schlechten Tagespläne nicht “überleben”, folgt daraus, dass beim relaxierten Zustand alle übrigen Pläne eines Agenten einen ähnlich grossen Nutzen haben. Bild 3 zeigt dieses Verhalten. Dabei beschreibt die blaue Kurve den durchschnittlichen Nutzen des Plans aller Agenten, welcher in der entsprechenden Iteration ausgeführt wurde. Die grüne, resp. rote Kurve bildet den durchschnittlichen Nutzen des besten, resp. des schlechtesten Plans aller Agenten ab, die bei der aktuellen Iteration noch vorhanden ist. Man erkennt, dass in diesem Beispiel nach der Iteration 70 dem oben beschriebenen relaxierten Zustand nähert und und bei 100 Iterationen nur noch eine durchschnittliche Varianz von ca. 2 Punkten aufweist.

Interessant ist aber zudem, was vor Iteration 70 passiert, im Speziellen bei Iteration 15. Aus Abbildung 3 ist ersichtlich, dass in Iteration 15 die Durchschnitte der Scores für die ausgeführten Pläne (blaue Kurve) sowie die schlechtesten Pläne (rote Kurve) sehr schlecht sind. Dies ist auf sogenannte Network Breakdowns zurückzuführen [47]. Durch den Optimierungsprozess und die gegebenen Randbedingungen (beispielsweise Zeitfenster für Beginn der Arbeit-Aktivität) kann es passieren, dass viele Agenten ähnliche Pläne ausprobieren, was zu hohen Verkehrs-



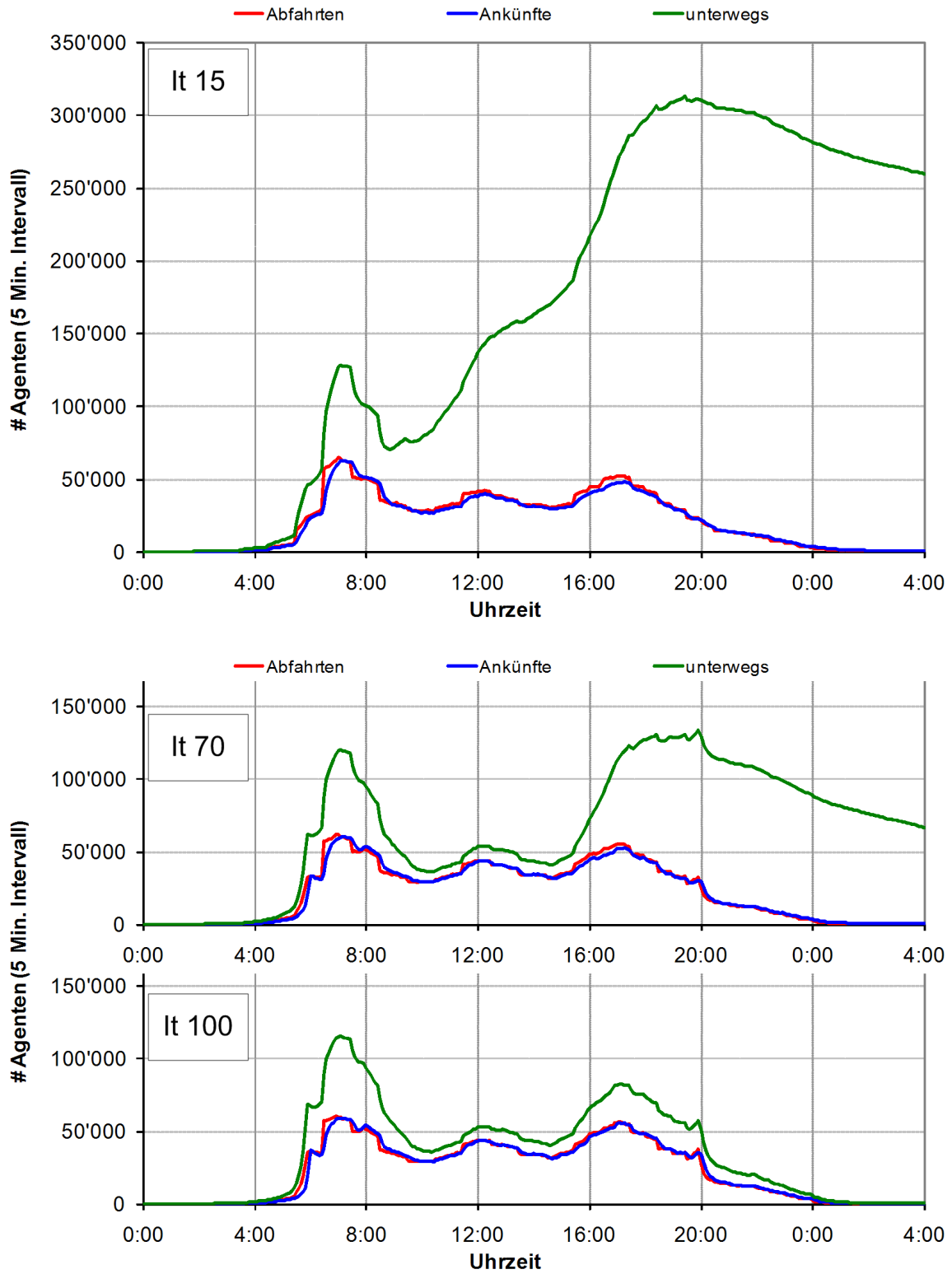


Abbildung 4: Abfahrts-, Ankunfts- und "unterwegs"-Histogramm für Iteration 15, 70 und 100

dichten auf bevorzugten Strassen führt. Entsteht dann an einer Stelle im Netzwerk ein Stau durch temporäre Überlastung, kann dieser Stau durch die allgemein hohe Verkehrsdichte nicht mehr aufgefangen werden. Es bilden sich Rückstaus, die sich immer weiter im Netzwerk ausbreiten. Die Auflösung dieser Staus dauert im Modell sehr lange, wodurch hohe Reisezeiten entstehen (schwarze Kurve in Abbildung 3), was in der Nutzenfunktion zu grossen negativen Beiträgen führt.

Nach einem solchen Zusammenbruch des Netzwerks wechseln die meisten Agenten ihren Plan, da der zuletzt ausgeführte Plan nun einen schlechten Nutzen aufweist. Es wird somit also quasi die letzte Optimierungsstufe verworfen, und es werden stärker diversifizierte Pläne ausgewählt. Durch die stärkere Diversifizierung hinsichtlich Abfahrtszeit und Routenwahl kommt es im allgemeinen zu kürzeren Fahrzeiten (schwarze Kurve in Abbildung 3), was sich durch den darauf folgenden höheren durchschnittlichen Nutzen bemerkbar macht.

Nach Iteration 70 scheint sich eine solche Kombination von Plänen zu ergeben, welche eine stabile Verkehrslage erzeugen, die robust gegen Veränderungen einzelner Agenten ist. Gute Pläne werden durch das Replanning verdoppelt und beibehalten, sofern sie auch gut sind. Dafür fallen schlechte Pläne weg, so dass zu letzt hauptsächlich gute Pläne übrig sind, was in Abbildung 3 durch die Annäherung der roten Kurve zu den beiden anderen Kurven ersichtlich ist.

Diese Effekte zeigen auch die Darstellungen in Abbildung 4. Die roten, resp. blauen Kurven zeigen für eine bestimmte Iteration die Anzahl Agenten auf, die zum gegebenen Zeitpunkt losfahren, resp. ankommen. Die grüne Kurve gibt Auskunft darüber, wie viele Agenten zur Zeit im System unterwegs sind. In Iteration 15 erkennt man, dass um ca. 09:30 Uhr ein solcher "Breakdown" stattfindet, und sich danach das System nicht mehr davon erholt (etwa 310'000 reisende Agenten um 20:00 Uhr). Die Situation in Iteration 70 ist nicht mehr so dramatisch, jedoch erkennt man einen kleineren Zusammenbruch während der Abendspitze. Nach dieser Iteration beginnt aber das System zu relaxieren, was sich im Histogramm widerspiegelt: Es bildet sich die zu erwartenden Morgen und Abendspitzen heraus, mit Rückgang des Verkehrsaufkommens während des Tages und kaum Verkehr in der Nacht (grüne Kurve von Iteration 100 aus Abbildung 4).

### **6.3 Rechenzeiten der Optimierung**

Die Gesamtrechenzeit eines Relaxationsprozesses besteht aus den Teilprozessen wie abgebildet in Bild 2. Hinzu kommt, dass zur Speicherung von Zwischenergebnissen und -analysen zusätzlich Rechenleistung benötigt wird. Diese Funktion kann grundsätzlich vom Benutzer ausgeschaltet werden, so dass nur das Endresultat gespeichert wird. Es ist aber ein hilfreiches Feature, damit auch während des Prozesses die Optimierung analysiert werden kann, um gegebenenfalls frühzeitig abubrechen. Aus diesem Grund wird im Folgenden dieser Teil der Gesamtrechenzeit nicht aussen vor gelassen. Im Detail lässt sich der Prozess zeitlich folgender-

massen aufteilen:

1. *Initialisierung*: Einlesen und Verwalten der Infrastrukturdaten (Netzwerk und Gebäude) und der initialen Nachfrage
2. *Iteration 0*: Erstmaliges Ausführen der Verkehrsflusssimulation und der Berechnung des Nutzens (Iteration 0)
3. *Iteration 1-n*: Replanning, Verkehrsflusssimulation und Scoring
4. *Iteration 0,1,2 und jede zehnte Iteration*: Speichern der Zwischenergebnisse und der Analysen
5. *Finalisierung*: Speichern des Endzustandes (relaxierte Tagespläne)

Zusätzlich zu den einzelnen Prozessschritten wird für manche Module zusätzliche Rechenzeit verwendet zur Initialisierung und Finalisierung. So zum Beispiel, benötigt der initialisierungsprozess des “Landmarks-A\*” Router Modules beschrieben in Abschnitt 5.2 einige Sekunden zur Berechnung der Orientierungspunkte (*Landmarks*) [siehe dazu 42]. Die DEQSim (Abschnitt 3.2) benötigt wiederum einige Minuten um das Netzwerk und die individuelle Nachfrage einzulesen und in optimierten Datenstrukturen zu speichern. Im Falle der parallelisierten Anwendung der DEQSim wird zum Aufbau der Prozesse wiederum Zeit aufgewendet. Weitere, JAVA interne Funktionen, wie zum Beispiel der “Garbage Collector”, wie auch Hardware-Grenzen (I/O-Zeiten) verursachen zusätzliche Zeitverzögerungen.

Die Abbildung 5 zeigt für die ersten 40 Iterationen eines Relaxationsprozesses die Anteile der Rechenzeiten an der Gesamtrechenzeit auf. In diesem Setup wird zur Routenberechnung 8 parallel laufende “Landmarks-A\*” Router Module verwendet (Abschnitt 5.2). Zeitplanung geschieht anhand weiterer 8 “Time Allocation Mutator” Module (Abschnitt 5.1). Zur Ausführung der Nachfrage kommt die parallele Version der DEQSim mit 8 Prozessen zur Anwendung. Dieser Aufbau produziert die Relaxation wie in Abbildung 3 geschildert.

Mit den oben erwähnten Zusatzaufwänden zeigt sich, dass die DEQSim im Schnitt etwa 8–10 Minuten benötigt (grüne Balken in Abbildung 5). Dazu kommt noch der Overhead von ca. 20 Minuten des Datenaustausches mit der JAVA Implementierung von MATSim-T (blaue und orange Balken). Replanning (Zeiten und Routen) beansprucht in etwa 90 Sekunden Rechenzeit, wobei der Löwenanteil die Landmarks-A\* Router verbrauchen. Das Replanning nach einer “Breakdown”-Situation (siehe Iteration 15 aus Abbildung 3) verursacht einen signifikanten Anstieg der Rechenzeit des Routers (ca. 9 Minuten). Der Grund dafür ist, dass die Performanz des Landmarks-A\* Routers bei massiven Unterschieden zwischen Kantenreisezeiten bei freier Fahrt und starker Belastung nachlässt. Detaillierte Informationen zur Performanz des Routers findet man in Lefebvre und Balmer [42].

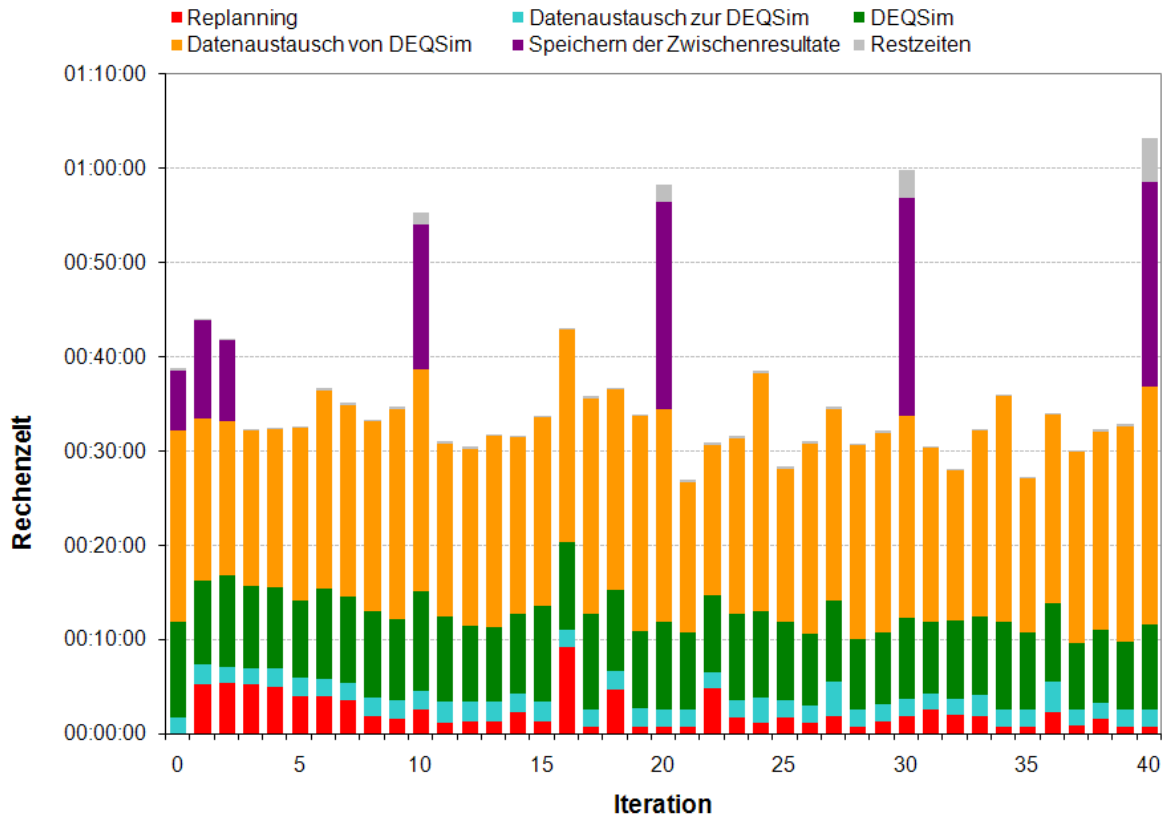


Abbildung 5: Anteile der Teilprozesse an der Gesamtrechenzeit

Gesamthaft benötigt somit eine Iteration des koevolutionären Algorithmus von MATSim-T zur Berechnung der individuellen, zeit-dynamischen Gesamt-Tagesnachfrage für 7.1 Mio Wege auf dem Schweizerischen Planungsnetz mit 60'000 Kanten im Mittel etwa 32 Minuten. Mit dem zusätzlichen Overhead von ca. 22 Minuten zur Speicherung von Zwischenresultaten und Analysen (violette und graue Balken in Abbildung 5) und dem Eintreten des relaxierten Zustandes nach 100 Iterationen (Abbildung 3) benötigt die Berechnung der resultierenden Nachfrage etwa 3.2 Tage.

#### 6.4 Kombinationen

Der Anwendungsfall, beschreiben in Abschnitt 1.2 mit dem Setup von Abschnitt 6.1 kann mit den oben beschriebenen Modulen relaxiert werden. Tabelle 1 zeigt ein Set von möglichen Modul-Kombinationen, deren durchschnittliche Laufzeiten und die Gesamt-Laufzeit, die benötigt wird, um den stabilen Zustand zu erreichen. Durch Ersatz eines *Random Mutation* Modules (Time Allocator Mutator) mit einem *Best Response* Modul (planomat) erreicht man eine signifikante Reduktion der benötigten Anzahl Iterationen. Andererseits verbraucht der planomat im

Mikrosimulation	Routen	Zeiten	# Iterationen	Gesamtlaufzeit
DEQSim (1 CPU)	Landmarks-A*	Time Allocation Mutator	100	~5.2 Tage
MobSim (1 CPU)	Landmarks-A*	Time Allocation Mutator	100	~5.5 Tage
DEQSim (1 CPU)	Landmarks-A*	planomat	30	~1.9 Tage
MobSim (1 CPU)	Landmarks-A*	planomat	30	~2.1 Tage
DEQSim (8 CPU)	Landmarks-A*	Time Allocation Mutator	100	~3.2 Tage
DEQSim (8 CPU)	Landmarks-A*	planomat	30	~1.5 Tage

Modul	Mittl. Laufzeit	Bemerkungen
DEQSim (1 CPU)	ca. 70 Minuten	ca. 50 Minuten DEQSim und ca. 20 Minuten I/O Overhead
DEQSim (8 CPU)	ca. 28 Minuten	
Mobsim	ca. 70 Minuten	Nach "Breakdowns" signifikant länger (ca. 6–10 Minuten) Vernachlässigbar klein
Landmarks-A*	ca. 1.5 Minuten	
TAM	-	
planomat	ca. 22 Minuten	

**Tabelle 1: Laufzeiten verschiedener Modul-Kombinationen für den Anwendungsfall: ca. 2.3 Mio. Agenten mit gesamthaft etwa 7.1 Mio. Wege auf dem nationalen Netzmodell mit ca. 60'000 Kanten, vollständig zeit-dynamisch simuliert und optimiert für einen durchschnittlichen Werktag.**

Mittel 30 Minuten mehr Rechenzeit pro Iteration. Dieser Tradeoff zahlt sich aber letztlich aus und man erreicht einen Gesamt-Performanzgewinn von ca. 100%.

Die Rechenleistung der Mobility Simulation (Mobsim) und der DEQSim sind in etwa gleichwertig, rechnet man mit ein, dass durch den notwendigen Datenaustausch zwischen MATSim-T und der DEQSim einen nicht zu vernachlässigen Overhead entsteht. Durch die Parallelisierung der DEQSim ist der Performancegewinn zwar beeindruckend, der I/O Anteil bleibt jedoch unverändert.

Zusammenfassend ist aber zu vermerken, dass diese Resultate klar aufzeigen, dass Mikrosimulationen durchaus die Leistungen erreichen, um den Anforderungen der Verkehrsplanung (grosse Planungsgebiete, zeitdynamisch und detailliert) zu genügen.

## 7 Initiale Individuelle Nachfrage-Modellierung

In Abbildung 2 des Abschnitt 2 wird die initiale Nachfrage als Voraussetzung für die Optimierung in MATSim-T angegeben. In diesem Abschnitt wird aufgezeigt, wie man das Toolkit dazu verwenden kann, die initiale Tages-Nachfrage jedes einzelnen Individuums zu erstellen. Der Grund warum dieser Pre-Prozess am Ende dieser Arbeit vorgestellt wird, liegt darin, dass der Lösungsraum — definiert im Setup der Optimierung — festlegt, welche Aspekte des Tagespläne im Pre-Prozess nicht modelliert werden müssen.

Die *initiale individuelle Nachfrage-Modellierung* hat somit die Aufgabe, diejenigen Bereiche der Tagespläne jedes einzelnen Agenten zu modellieren, die nicht im Iterationsprozess optimiert werden. Um ein bestmögliches Abbild der realen Nachfrage zu erreichen, stützt sich dieser Teil

auf Kenntnisse, Umfragen und soziodemographische Grundlagen der zu untersuchenden Region. MATSim-T ist so konstruiert, dass es flexibel mit verschiedensten Input-Daten operieren kann. Je nach Region können dabei die vorhandenen Daten in Qualität, Detailgrad und Quantität variieren. Die Module zur initialen Nachfrage-Modellierung werden entsprechend angepasst oder ersetzt. Aus diesem Grund variieren entsprechend auch die Laufzeiten zur Generierung der initialen Nachfrage. Grundsätzlich ist aber dazu zu sagen, dass dieser Pre-Prozess sequenzieller Natur ist. Entsprechend werden alle zur Generierung benötigten Module nur einmal verwendet.

Die Modellierung der individuellen, initialen Nachfrage für den Anwendungsfall ist detailliert in Ciari *et al.* [27] und Meister *et al.* [22] zu finden. Die dazu verwendete Rechenzeit betrug ca. 6 Stunden.

## 7.1 Architektur

MATSim-T operiert auf disagregierten Informationen, das heisst, die Infrastruktur basiert auf Koordinaten und nicht auf Aggregaten, wie zum Beispiel Zonen (Kreise, Gemeinden, etc.). Aktivitäten — und demzufolge die Gebäude in denen Aktivitäten ausgeführt werden können — werden direkt den Kanten des Netzwerkes zugeordnet. Da das Netz jedoch eine bestimmte Auflösung hat, definiert es die “Unschärfe” der Modellierung. Einfach ausgedrückt bedeutet dies, dass das Untersuchungsgebiet genau so viele “Zonen” aufweist, wie es (gerichtete) Kanten im Netzwerk gibt, in dem hier verwendeten Anwendungsfall somit ca. 60'000 Zonen. Für hoch aufgelöste Netze kann dies bis über eine Million ansteigen. Da die Ausgangsdaten aber typischerweise aggregierter Natur sind, müssen solche Informationen entsprechend disaggregiert werden können. MATSim-T stellt eine beliebige Anzahl Aggregationsebenen zur Verfügung, um solche Daten zu halten und bei Gebrauch auf die Gebäude, Aktivitäten, Personen und das Netzwerk zu disaggregieren.

Die Modellierung der initialen Nachfrage kann, je nach Bedürfnissen des Anwenders und den zur Verfügung stehenden Ausgangsdaten in beliebig viele Teilprozesse unterteilt werden. Jeder dieser Prozesse wird von einem Modul implementiert. Diese Module können somit beliebig verwendet, erweitert, ersetzt oder ausgelassen werden.

Zu jedem beliebigen Zeitpunkt während des Modellierungsprozesses können zudem die Zwischenresultate ausgegeben werden. Dies ist umso wichtiger, da es typischerweise notwendig ist, die Resultate eines als Modul implementierten Modells statisch zu validieren. Die Zwischenresultate können wiederum als Ausgangslage für weitere Modellierungsschritte benutzt werden. Ebenso ist es möglich mehrere Prozessschritte hintereinander, “in einem Zug” auszuführen.

Ein weiterer, wichtiger Aspekt ist das sogenannte *streaming* Verfahren zur Erzeugung einer initialen Nachfrage. Während die Infrastrukturdaten (Gebäude, Netzwerk, aber auch Aggregationsebenen) verhältnismässig wenig Speicher bedarf, benötigt die Nachfrage ohne weiteres

mehrere GByte an Memory. Da aber die Nachfrage individuell für jede synthetische Person im Untersuchungsraum erzeugt wird, kann der Speicherbedarf auf einfache Weise minimiert werden: Man speichert einen Agenten im Memory ab, wendet darauf die Nachfragemodellierungsmodule an, schreibt den Agenten wieder aus und löscht ihn danach aus dem Speicher. Anschliessend wird der nächste eingelesen, und so weiter. Damit lässt sich eine unbegrenzte Anzahl an Agenten mit individueller Nachfrage auf einem herkömmlichen Rechner modellieren. Eine ausführlichere Beschreibung der Funktionalitäten von MATSim-T findet sich in der Dissertation von Balmer [28] wie auch in Balmer *et al.* [48].

## 8 Diskussion und Ausblick

Diese Arbeit zeigt auf, dass die Entwicklungen der letzten Jahre in Rechenarchitektur, Prozessorleistung und Optimierung von Programm-Implementationen es nun möglich macht, grosse Szenarien in der Verkehrsplanung mit Agenten-basierten Simulationen in vernünftiger Zeit zu berechnen. Zudem zeigt sich, dass dies noch lange nicht das Optimum der Performanz ist. So zum Beispiel ist es durch eine Re-Implementation der parallelen DEQSim (Abbschnitt 3.2) in JAVA als integraler Bestandteil von MATSim-T möglich, den Overhead, der durch den Datenaustausch zwischen DEQSim und MATSim-T *pro Iteration* entsteht (ca. 20 Min. für den geschilderten Anwendungsfall), einzusparen, was eine Reduktion von ca. 40% des Gesamtaufwandes bedeutet. Ein Szenario der Grössenordnung der gesamten Schweiz lässt sich somit in etwa einem Tag berechnen.

Das Setup der Optimierung lässt zudem weitere Optimierungsmöglichkeiten zu. So zum Beispiel ist es durch adaptive Variationsraten möglich, die Anzahl Iterationen bis zum relaxierten Zustand weiter zu minimieren. Auch die Replanning Module haben Optimierungspotential, im Speziellen das Routing Modul und der *planomat*. All diese Optimierungen sind anzustreben, denn durch die funktionale Erweiterung des Systems, wie zum Beispiel Orts- oder Verkehrsmittelwahl wird sicherlich wieder mehr Rechenzeit benötigt, sei es, weil die Module komplex sind, oder weil damit das System mehr Iterationen bis zur Relaxation benötigt.

Zum Abschluss sei hier noch erwähnt, dass das Resultat von MATSim-T nicht der Verkehr ist, sondern die detaillierte Beschreibung jedes einzelnen Agenten. Anders ausgedrückt heisst dies, dass man zu jedem Zeitpunkt des Tages, für jede synthetische Person ermittelt, *wo* sie gerade ist, und *was* sie gerade macht. Das daraus entstehende Verkehrsaufkommen ist dabei nur das "Nebenprodukt".

## Literatur

- [1] DE PALMA, A. und F. MARCHAL (2002) Real cases applications of the fully dynamic METROPOLIS tool-box: An advocacy for large-scale mesoscopic transportation systems, *Networks and Spatial Economics*, 2 (4) 347–369.

- [2] PTV (2006) Planung Transport Verkehr (PTV), Webseite, <http://www.ptv.de>.
- [3] HANSON, S. und K. O. BURNETT (1982) The analysis of travel as an example of complex human behaviour in spatially-constraint situation: Definition and measurement issues, *Transportation Research Part A: Policy and Practice*, 16 (2) 87–102.
- [4] AXHAUSEN, K. W., A. ZIMMERMANN, S. SCHÖNFELDER, G. RINDSFÜSER und T. HAUPT (2002) Observing the rhythms of daily life: : A six-week travel diary, *Transportation*, 29 (2) 95–124.
- [5] SCHÖNFELDER, S., K. W. AXHAUSEN, N. ANTILLE und M. BIERLAIRE (2002) Exploring the potentials of automatically collected GPS data for travel behaviour analysis - a swedish data source, in J. Mölthen und A. Wytzisk (Hg.) *GI-Technologien für Verkehr und Logistik - IfGI*, 155–179, IfGI prints, Institut für Geoinformatik, Münster.
- [6] WOLF, J., S. SCHÖNFELDER, U. SAMAGA, M. OLIVEIRA und K. W. AXHAUSEN (2004) Eighty weeks of Global Positioning System traces, *Transportation Research Record*, 1870, 46–54.
- [7] VOVSHA, P., E. PETERSEN und R. DONNELLY (2002) Microsimulation in travel demand modeling: Lessons learned from the New York best practice model, *Transportation Research Record*, 1805, 68–77.
- [8] BOWMAN, J. L., M. BRADLEY, Y. SHIFTAN, T. K. LAWTON und M. E. BEN-AKIVA (1999) Demonstration of an activity-based model for Portland, *World Transport Research*, 3, 171–184.
- [9] BHAT, C. R., J. Y. GUO, S. SRINIVASAN und A. SIVAKUMAR (2004) A comprehensive econometric microsimulator for daily activity-travel patterns, *Transportation Research Record*, 1894, 57–66.
- [10] PENDYALA, R. M. (2004) Phased implementation of a multimodal activity-based travel demand modeling system in florida. volume II: FAMOS users guide, Forschungsbericht, Florida Department of Transportation, Tallahassee, <http://www.eng.usf.edu/~pendyala/publications/report/FAMOS%20Users%20Guide.pdf>.
- [11] ARENTZE, T. A., F. HOFMAN, H. MOURIK und H. J. P. TIMMERMANS (2000) Albatross: A multi-agent rule-based model of activity pattern decisions, *Transportation Research Record*, 1706, 136–144.
- [12] SCHNITTGER, S. und D. ZUMKELLER (2004) Longitudinal microsimulation as a tool to merge transport planning and traffic engineering models - the MobiTopp model, Paper Presentation European Transport Conference, Strasbourg, Oktober 2004.



- [13] ZUMKELLER, D. (1989) Ein sozialökologisches Verkehrsmodell zur Simulation von Maßnahmewirkungen, Dissertation, Technische Universität Braunschweig, Institut für Verkehr und Stadtbauwesen (IVS), Braunschweig.
- [14] AXHAUSEN, K. W. (1988) Eine ereignisorientierte Simulation von Aktivitätenketten zur Parkstandwahl, Dissertation, Universität Karlsruhe, Karlsruhe.
- [15] BALMER, M. und K. NAGEL (2006) Shape morphing of intersection layouts using curb side oriented driver simulation, in J. P. van Leeuwen und H. J. P. Timmermans (Hg.) Innovations in Design & Decision Support Systems in Architecture and Urban Planning, 167–183, Springer, Eindhoven.
- [16] AXHAUSEN, K. W. und R. HERZ (1989) Simulating activity chains: German approach, Journal of Transportation Engineering, 115 (3) 316–325.
- [17] TRANSIMS (2006) TRansportation ANalysis and SIMulation System, Webseite, Dezember 2006, <http://transims.tsasa.lanl.gov>.
- [18] W3C (2006) eXtensible Markup Language (XML), World Wide Web Consortium (W3C), <http://www.w3.org/XML>.
- [19] WADDELL, P., A. BORNING, M. NOTH, N. FREIER, M. BECKE und G. ULFARSSON (2003) Microsimulation of urban development and location choices: Design and implementation of urbansim, Networks and Spatial Economics, 3 (1) 43–67.
- [20] SALVINI, P. A. und E. J. MILLER (2005) ILUTE: An operational prototype of a comprehensive microsimulation model of urban systems, Networks and Spatial Economics, 5 (2) 217–234.
- [21] HUNT, J. D., R. JOHNSTON, J. E. ABRAHAM, C. J. RODIER, G. R. GARRY, S. H. PUTMAN und T. DE LA BARRA (2000) Comparisons from Sacramento model test bed, Transportation Research Record, 1780, 53–63.
- [22] MEISTER, K., M. RIESER, F. CIARI, A. HORNI, M. BALMER und K. W. AXHAUSEN (2008) Anwendung eines agentenbasierten Modells der Verkehrsnachfrage auf die Schweiz, Paper Presentation Heureka '08, Stuttgart, März 2008.
- [23] VRTIC, M., P. FRÖHLICH und K. W. AXHAUSEN (2003) Schweizerische Netzmodelle für Strassen- und Schienenverkehr, in T. Bieger, C. Lässer und R. Maggi (Hg.) Jahrbuch 2002/2003 Schweizerische Verkehrswirtschaft, 119–140, SVWG Schweizerische Verkehrswissenschaftliche Gesellschaft, St. Gallen.
- [24] Bundesamt für Statistik (2001) Eidgenössische Betriebszählung 2001 - Sektoren 2 und 3, Bundesamt für Statistik, Neuchâtel.

- [25] Bundesamt für Statistik (2000) Eidgenössische Volkszählung 2000.
- [26] Bundesamt für Statistik (2006) Ergebnisse des Mikrozensus 2005 zum Verkehrsverhalten, Bundesamt für Statistik, Neuchâtel.
- [27] CIARI, F., M. BALMER und K. W. AXHAUSEN (2007) Mobility tool ownership and mode choice decision processes in multi-agent transportation simulation, Paper Presentation the 6th Swiss Transport Research Conference, Ascona, September 2007, [http://www.strc.ch/pdf\\_2007/ciari.pdf](http://www.strc.ch/pdf_2007/ciari.pdf).
- [28] BALMER, M. (2007) Travel demand modeling for multi-agent traffic simulations: Algorithms and systems, Dissertation, ETH Zürich, Zürich, Mai 2007.
- [29] WIEDEMANN, R. (1974) Simulation des Verkehrsflusses, Dissertation, Universität Karlsruhe, Karlsruhe.
- [30] WARDROP, J. G. (1952) Some Theoretical Aspects of Road Traffic Research, Institute of Civil Engineers.
- [31] HOLLAND, J. H. (Hg.) (1992) Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT Press, Cambridge.
- [32] PALMER, R. G., B. W. ARTHUR, J. H. HOLLAND, B. LEBARON und P. TAYLER (1994) Artificial economic life: A simple model of a stockmarket, *Physica D: Nonlinear Phenomena*, 75 (1-3) 264–274.
- [33] NAGEL, K. und M. SCHRECKENBERG (1992) A cellular automaton for freeway traffic, *Journal de Physique I*, 2 (1992) 2221–2229.
- [34] SIMON, P. M., J. ESSER und K. NAGEL (1999) Simple queueing model applied to the city of Portland, *International Journal of Modern Physics C (IJMPC)*, 10 (5) 941–960.
- [35] CETIN, N. (2005) Large-scale parallel graph-based simulations, Dissertation, ETH Zürich, Zürich.
- [36] CHARYPAR, D., K. W. AXHAUSEN und K. NAGEL (2007) An event-driven parallel queue-based microsimulation for large scale traffic scenarios, Paper Presentation the 11th World Conference on Transportation Research, Berkeley, Juni 2007, <http://www.ivt.ethz.ch/vpl/publications/reports/ab425.pdf>.
- [37] AVINERI, E. und J. N. PRASHKER (2003) Sensitivity to uncertainty: Need for a paradigm shift, *Transportation Research Record*, 1854, 90–98.

- [38] CHARYPAR, D. und K. NAGEL (2005) Generating complete all-day activity plans with genetic algorithms, *Transportation*, 32 (4) 369–397.
- [39] VICKREY, W. S. (1969) Congestion Theory and Transport Investment, *The American Economic Review*, 59 (2) 251–260.
- [40] ARNOTT, R., A. DE PALMA und R. LINDSEY (1993) A structural model of peak-period congestion: A traffic bottleneck with elastic demand, *The American Economic Review*, 83 (1) 161–179.
- [41] 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.
- [42] LEFEBVRE, N. und M. BALMER (2007) Fast shortest path computation in time-dependent traffic networks, Paper Presentation the 6th Swiss Transport Research Conference, Ascona, September 2007, [http://www.strc.ch/pdf\\_2007/lefebvre.pdf](http://www.strc.ch/pdf_2007/lefebvre.pdf).
- [43] MEISTER, K. (2004) Erzeugung kompletter Aktivitätenpläne für Haushalte mit genetischen Algorithmen, Diplomarbeit, IVT, ETH Zürich, Zürich, <http://www.ivt.ethz.ch/docs/students/dip44.pdf>.
- [44] HACKNEY, J. K. und K. W. AXHAUSEN (2006) An agent model of social network and travel behavior interdependence, Paper Presentation the 11th International Conference on Travel Behaviour Research (IATBR), Kyoto, August 2006.
- [45] TITZE, T. (2007) Entwicklung eines ÖV-Routingmoduls für Multiagentensimulationen, Übungsbericht, Verkehrssystemplanung und Verkehrstelematik (VSP), Technische Universität Berlin, Berlin.
- [46] RIESER, M., U. BEUCK und K. NAGEL (2007) Researching the influence of time-dependent tolls with a multi-agent traffic simulation, Arbeitsbericht, 07-17, Verkehrssystemplanung und Verkehrstelematik (VSP), Technische Universität Berlin, Berlin, <http://fgvsp01.vsp.tu-berlin.de/biblio/230/>.
- [47] RIESER, M. und K. NAGEL (2007) Network breakdown “at the edge of chaos” in multi-agent traffic simulations, Paper Presentation Proceedings of the European Conference on Complex Systems, Dresden, Oktober 2007, <http://vwitme011.vkw.tu-dresden.de/TrafficForum/dresden/>.
- [48] BALMER, M., K. NAGEL und B. RANEY (2006) Agent-based demand modeling framework for large scale micro-simulations, *Transportation Research Record*, 1985, 125–134.