

# Ameisenalgorithmen in der Verkehrssimulation

Johannes Renfordt

[johannes.renfordt@\[alumni.\]uni-ulm.de](mailto:johannes.renfordt@[alumni.]uni-ulm.de)

Institut für Stochastik, Universität Ulm

# Übersicht

- (Meta-) Heuristiken
  - ◇ Warum verwendet man Heuristiken?
  - ◇ Beispiele für (Meta-) Heuristiken
- Ameisenalgorithmen
- Kürzeste Wege
  - ◇ klassisch
  - ◇ mit Ameisenalgorithmen
- Verkehrssimulation
  - ◇ Auswertung vorhandener Daten
  - ◇ Simulation
  - ◇ Ausblick

# (Meta-) Heuristiken

## *Begriffserklärung*

- Das Wort Heuristik stammt von dem griechischen Verb *εὕρισκω* (ich finde) ab und kann mit *Findekunst* übersetzt werden. Oft ist eine Heuristik sehr stark problembezogen.
- Eine Metaheuristik ist (im mathematischen Sinne) ein Verfahren, das Heuristiken nutzt, um eine ganze Klasse von Problemstellungen abzudecken. Dabei wird aber oft die Metaheuristik auf das Problem adaptiert.

# (Meta-) Heuristiken

## *Wo kommen Heuristiken zum Einsatz?*

In kombinatorischen Optimierungsproblemen ist es oft sehr aufwendig, alle möglichen Kombinationen tatsächlich auszuprobieren. Beispiele für solche Problemstellungen sind u.a.

- Rundreiseprobleme wie das Travelling Salesman Problem oder
- (als Verallgemeinerung) das Vehicle Routing Problem,
- das Rucksack-Problem.

Unter Zuhilfenahme von Heuristiken hofft man, den Suchraum entscheidend verkleinern zu können ohne dadurch an Lösungsqualität zu verlieren.

# (Meta-) Heuristiken

## *Wie funktioniert eine Heuristik?*

Man kann sich leicht verschiedene Ansätze vorstellen, wie man heuristische Lösungen finden kann:

- durch intensives Betrachten des Problems,
- durch Analyse der Gemeinsamkeiten von bekannten optimalen Lösungen verwandter Probleme und deren Verallgemeinerung zu heuristischen Lösungen von Problemen dieser Klasse,
- durch Kombination bekannter suboptimaler Lösungen usw.

# (Meta-) Heuristiken

## *Probleme von Heuristiken*

Es ist schwer,

- ... die Güte einer Heuristik einzuschätzen. Würde man die optimale Lösung kennen, bräuchte man die Heuristik nicht
- ... die Ergebnisse direkt auf andere Probleme zu übertragen: eine gut funktionierende Heuristik für das TSP nützt möglicherweise wenig für die Lösung des Rucksack-Problems.

I.a. kann eine Heuristik beliebig gut oder schlecht im Verhältnis zur optimalen Lösung sein!

# (Meta-) Heuristiken

## *Beispiele für Heuristiken*

- *erstbeste* Lösung
- unterteilen, vereinfachen oder zusammenfassen der Problemstellung
- TSP: nächster Nachbar, entferntester Nachbar
- Rucksackproblem: wertvolle Gegenstände werden bevorzugt betrachtet
- ...

Der Phantasie sind keine Grenzen gesetzt!

# (Meta-) Heuristiken

## Metaheuristiken

Metaheuristiken verwenden Heuristiken. Dabei

- fassen sie mehrere heuristische Lösungen zu einer hoffentlich besseren zusammen oder
- suchen nach ähnlichen Lösungen *in der Nähe* einer zulässigen Lösung

Mit Metaheuristiken kann man oft völlig verschiedene Probleme gleichermaßen angehen - vorausgesetzt, man paßt die zugrundeliegende Heuristik an.

# (Meta-) Heuristiken

## *Beispiele für Metaheuristiken*

- Evolutionäre bzw. Genetische Algorithmen
- Bergsteigeralgorithmus
- Simulierte Abkühlung
- Lokale Suche und Tabu-Suche
- Ameisenalgorithmen

Insbesondere liefert die Natur viele Ideen!

# Übersicht

- (Meta-) Heuristiken
  - ◇ Warum verwendet man Heuristiken?
  - ◇ Beispiele für (Meta-) Heuristiken
- Ameisenalgorithmen
- Kürzeste Wege
  - ◇ klassisch
  - ◇ mit Ameisenalgorithmen
- Verkehrssimulation
  - ◇ Auswertung vorhandener Daten
  - ◇ Simulation
  - ◇ Ausblick

# Ameisenalgorithmen

## *Motivation*

Ameisen bilden oft zu Futterquellen sogenannte Ameisenstraßen, diese sind in aller Regel gerade bzw. führen auf vglw. direkten Wegen ans Ziel



# Ameisenalgorithmen

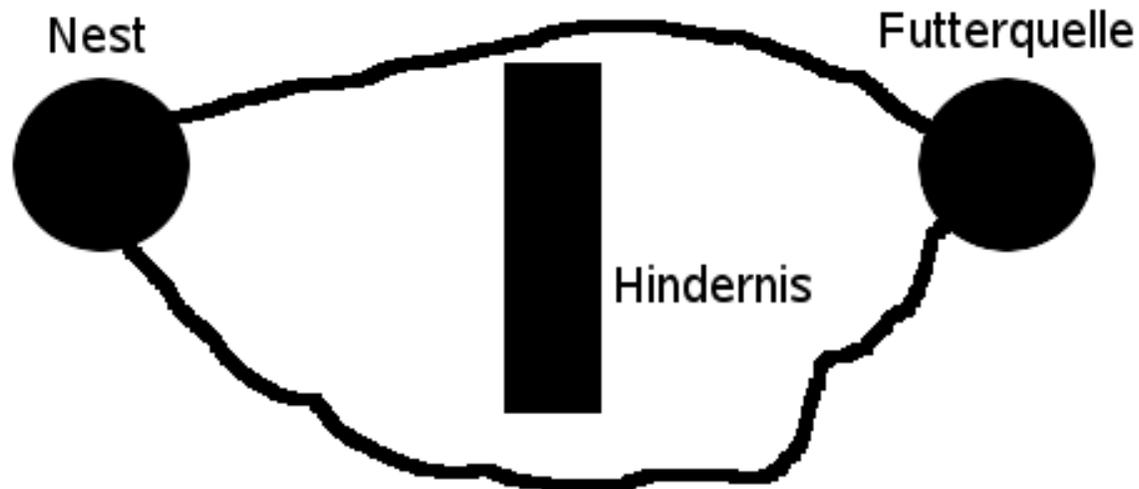
## *Biologie*

- Viele Ameisenarten kommunizieren untereinander durch sogenannte Pheromone.
- Je nach Situation werden von einer Ameise verschiedene Pheromone ausgebracht, die entsprechendes Verhalten bei anderen Ameisen bewirken.
- Pheromon verdunstet nach einiger Zeit.

Die verschiedenen Pheromonspuren sind ein kollektives Gedächtnis der Ameisenkolonie.

# Ameisenalgorithmen

## *Warum entstehen Ameisenstraßen?*



In selben Zeitabständen können mehr Ameisen den kürzeren Weg durchlaufen und mehr Pheromon hinterlassen!

# Ameisenalgorithmen

## *Ameisen im Rechner I*

Übertragen der Situation in *mathematische Begriffe*:

- Graph bildet Wegmöglichkeiten ab
- Startknoten entspricht Ameisenhaufen oder Nest
- Zielknoten entspricht Futterquelle

Gesucht ist eine möglichst kurze Ameisenstraße bzw. ein kürzester Weg vom Start- zum Zielknoten im Graphen.

# Ameisenalgorithmen

## *Ameisen im Rechner II*

- reale Ameisen wählen proportional zum Wegpheromon einen Weg aus
- z.B. seien drei Pheromonspuren  $\tau_i, i \in \{1, 2, 3\}$  gegeben. Mit Wahrscheinlichkeit

$$p_i = \frac{\tau_i}{\sum_{j=1}^3 \tau_j}$$

wählt die Ameise den Weg  $i$

- die Pheromonintensität kann man in den Kanten des Graphen speichern.

# Ameisenalgorithmen

## *Pheromon und Heuristik*

Das Pheromon stellt nur ein *Gedächtnis* dar. Daraus ergibt sich folgende Idee:

- finde heuristische Lösungen, die sich an den höchsten Pheromongehalten orientieren
- die Güte einer gefundenen zulässigen Lösung wird entlang des dazugehörigen Weges im Pheromongehalt dokumentiert
- iteriere diesen Vorgang

# Ameisenalgorithmen

## *Algorithmus I*

### Pseudo-Code:

- Initialisierung
- Solange Terminierungsbedingung nicht erfüllt ist:
  - ◇ Generiere für alle Ameisen jeweils eine Lösung
  - ◇ Vergleiche neue Lösungen mit der bisher besten bekannten
  - ◇ verändere den Pheromongehalt der Kanten des Graphen in Abhängigkeit der gefundenen Lösungen
- Liefere dem Aufrufer die beste Lösung zurück

# Ameisenalgorithmen

## Algorithmus II

Wie generiert man eine Lösung?

- Die Ameise befindet sich im Startknoten
- Solange Zielknoten nicht erreicht ist
  - ◊ Wähle nächsten Knoten  $j$  vom aktuellen Knoten  $i$  mit der Wahrscheinlichkeit  $p_{ij} := \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{l \in N(i)} \tau_{il}^\alpha \eta_{il}^\beta}$ ,  $\alpha > 0$ ,  $\beta > 0$
- Elimiere Zyklen in der Abfolge der besuchten Knoten

Dabei ist  $N(i)$  die (ggf. nur noch zu betrachtende) Nachbarschaft des Knoten  $i$ ,  $\eta_{ij}$  gibt an, wie gut die Kante zwischen den Knoten  $i$  und  $j$  die Heuristik erfüllt.

# Ameisenalgorithmen

## *Varianten*

Die verschiedenen Varianten des Algorithmus unterscheiden sich in erster Linie in der Veränderung des Pheromongehaltes der Kanten des Graphen:

- klassische Ameisen
- elitäre Ameisen
- rangbasierte Ameisen
- MAX-MIN-System
- ...

# Ameisenalgorithmen

## *Anwendbarkeit*

Ameisenalgorithmen kann man mindestens dann einsetzen, wenn

- ein graphentheoretisches Rundreise- oder Kürzeste-Wege-Problem (ggf. unter Nebenbedingungen) gegeben ist **und**
- die Heuristik  $\eta$  sinnvoll gewählt werden kann

Wichtig ist außerdem, daß eine Startlösung bestimmt wird und die Kanten des Graphen mit entsprechenden Pheromongehalten initialisiert werden.

# Ameisenalgorithmen

## *Parameter*

Wichtige Parameter der Ameisenalgorithmen sind:

- $\alpha, \beta$  bzw. die Funktionen  $\tau, \eta$
- Funktion, die einem Lösungsweg einen Pheromongehalt zuordnet,
- Update-Regel bzw. Variante des Ameisenalgorithmus,
- Verdunstungsfaktor,
- Anzahl der Ameisen pro Iteration,
- Anzahl der Iterationsschritte.

Viele Parameter stellen aber auch eine Verpflichtung zum Testen dar!

# Übersicht

- (Meta-) Heuristiken
  - ◇ Warum verwendet man Heuristiken?
  - ◇ Beispiele für (Meta-) Heuristiken
- Ameisenalgorithmen
- **Kürzeste Wege**
  - ◇ klassisch
  - ◇ mit Ameisenalgorithmen
- Verkehrssimulation
  - ◇ Auswertung vorhandener Daten
  - ◇ Simulation
  - ◇ Ausblick

# Kürzeste Wege

## *Dijkstra*

Der Algorithmus von Dijkstra ist ...

- schnell und führt zu der optimalen, d.h. kürzesten Lösung;
- basiert auf einem statischen Graphen;
- nicht parallelisierbar.

# Kürzeste Wege

## *Ameisenalgorithmen*

Ameisenalgorithmen sind ...

- flexibel aber möglicherweise ungenau;
- leicht parallelisierbar;
- beinhalten viele Parameter, die beliebig gesetzt werden können (und müssen)

# Kürzeste Wege

## *Beispiel*

Beispiel für einen per Ameisenalgorithmus gefundenen kurzen Weg unter gegebenen Geschwindigkeitswerten:



# Kürzeste Wege

## *Vorzüge von Ameisenalgorithmen*

Vorzüge von Ameisenalgorithmen gegenüber Dijkstra sind:

- Ameisenalgorithmen bieten bei mehreren Durchgängen ggf. eine ganze Reihe von ähnlich guten Lösungen und damit eine gewisse Variabilität
- Ameisenalgorithmen können auch Änderungen an der Graphenstruktur während der Laufzeit des Algorithmus problemlos verarbeiten und damit aktuellste Informationen direkt verarbeiten
- Anzahl der Iterationen kann dynamisch gehandhabt werden, d.h. in einem gewissen Umfang ist Genauigkeit vorgebbar

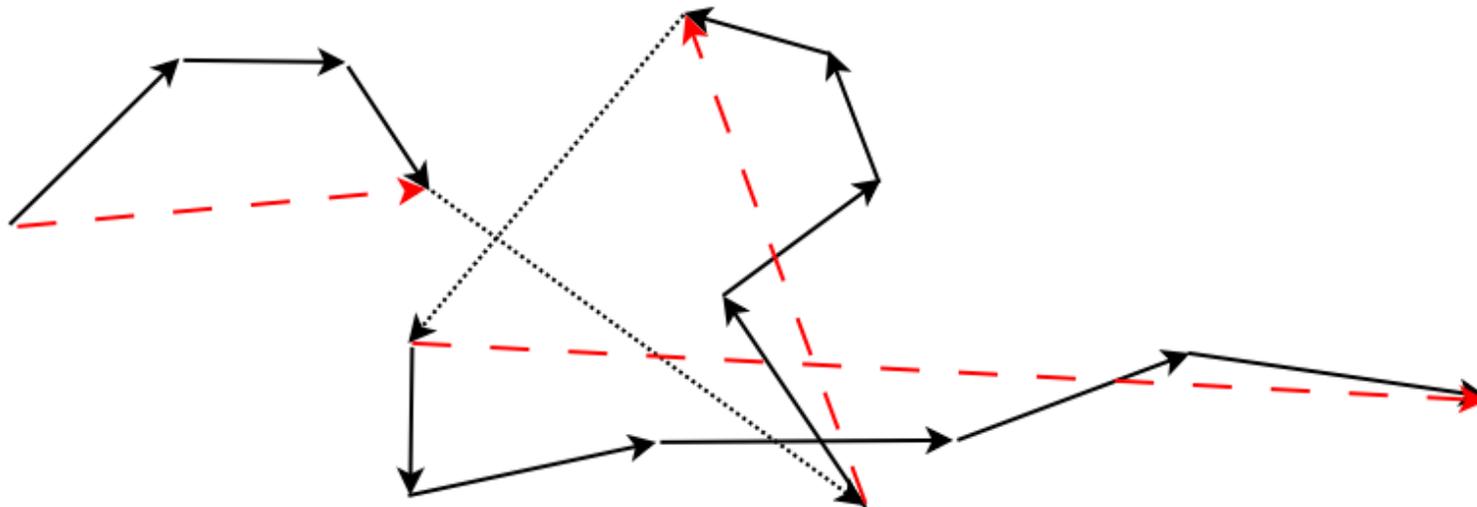
# Übersicht

- (Meta-) Heuristiken
  - ◇ Warum verwendet man Heuristiken?
  - ◇ Beispiele für (Meta-) Heuristiken
- Ameisenalgorithmen
- Kürzeste Wege
  - ◇ klassisch
  - ◇ mit Ameisenalgorithmen
- **Verkehrssimulation**
  - ◇ Auswertung vorhandener Daten
  - ◇ Simulation
  - ◇ Ausblick

# Verkehrssimulation

## *Begrifflichkeiten*

- Trajektorien sind zusammengehörende Teile eines Polygonzuges;
- Trajektorien sind wiederum in Segmente unterteilt.



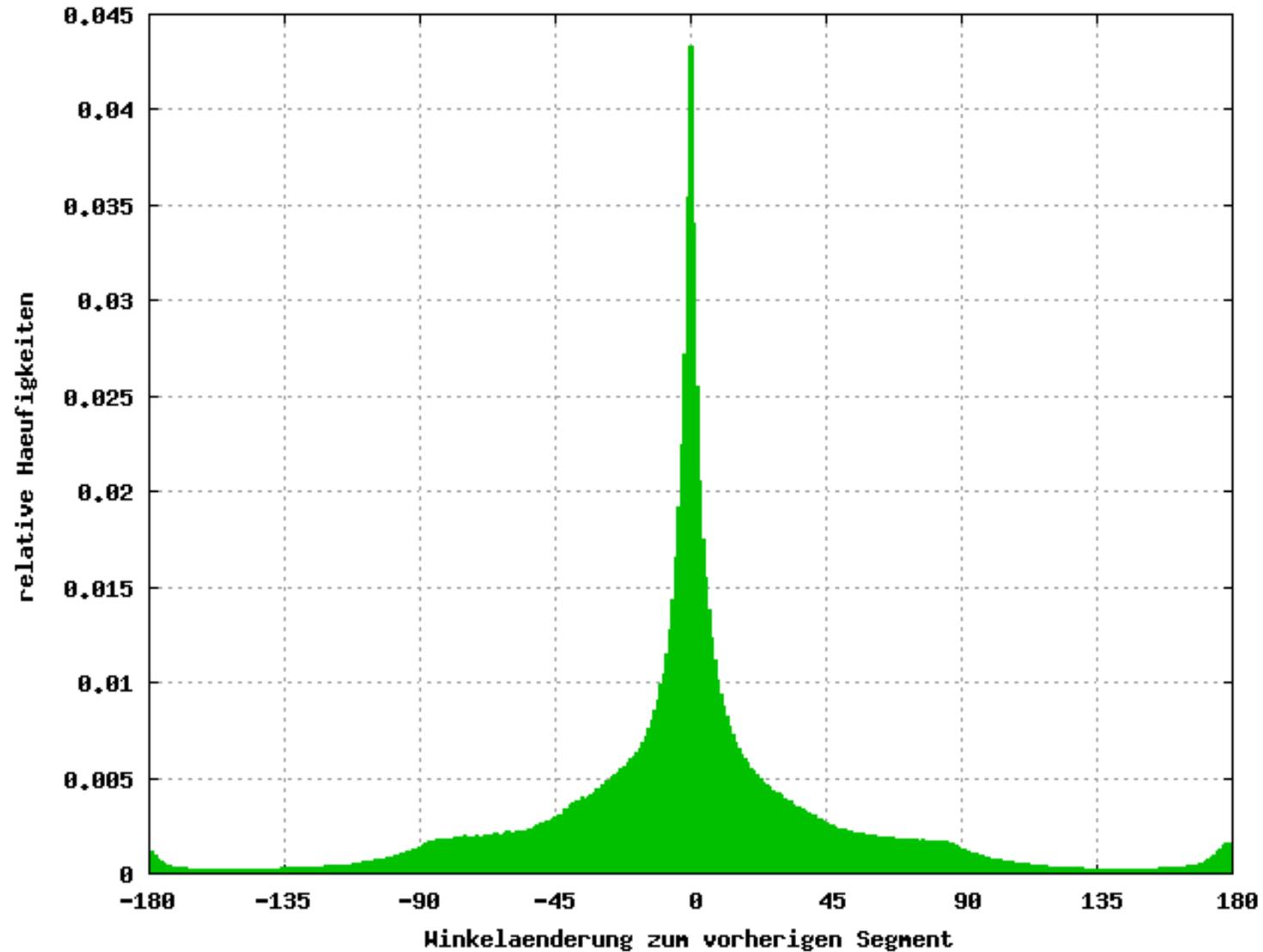
# Verkehrssimulation

## *Aufarbeiten vorhandener Daten*

- Taximeldungen innerhalb eines Beobachtungsfensters werden zu Trajektorien zusammengefaßt
- die Analyse dieser Daten dient als
  - ◇ Basis zur Simulation (z.B. Startzeitdifferenzen, Geschwindigkeitsmittelwerte, Anzahl der Segmente) und als
  - ◇ Vergleichsgrundlage (z.B. Geschwindigkeit in den Segmenten, Segmentlängen).

# Verkehrssimulation

## *Winkeländerungen zwischen den Segmenten I*



# Verkehrssimulation

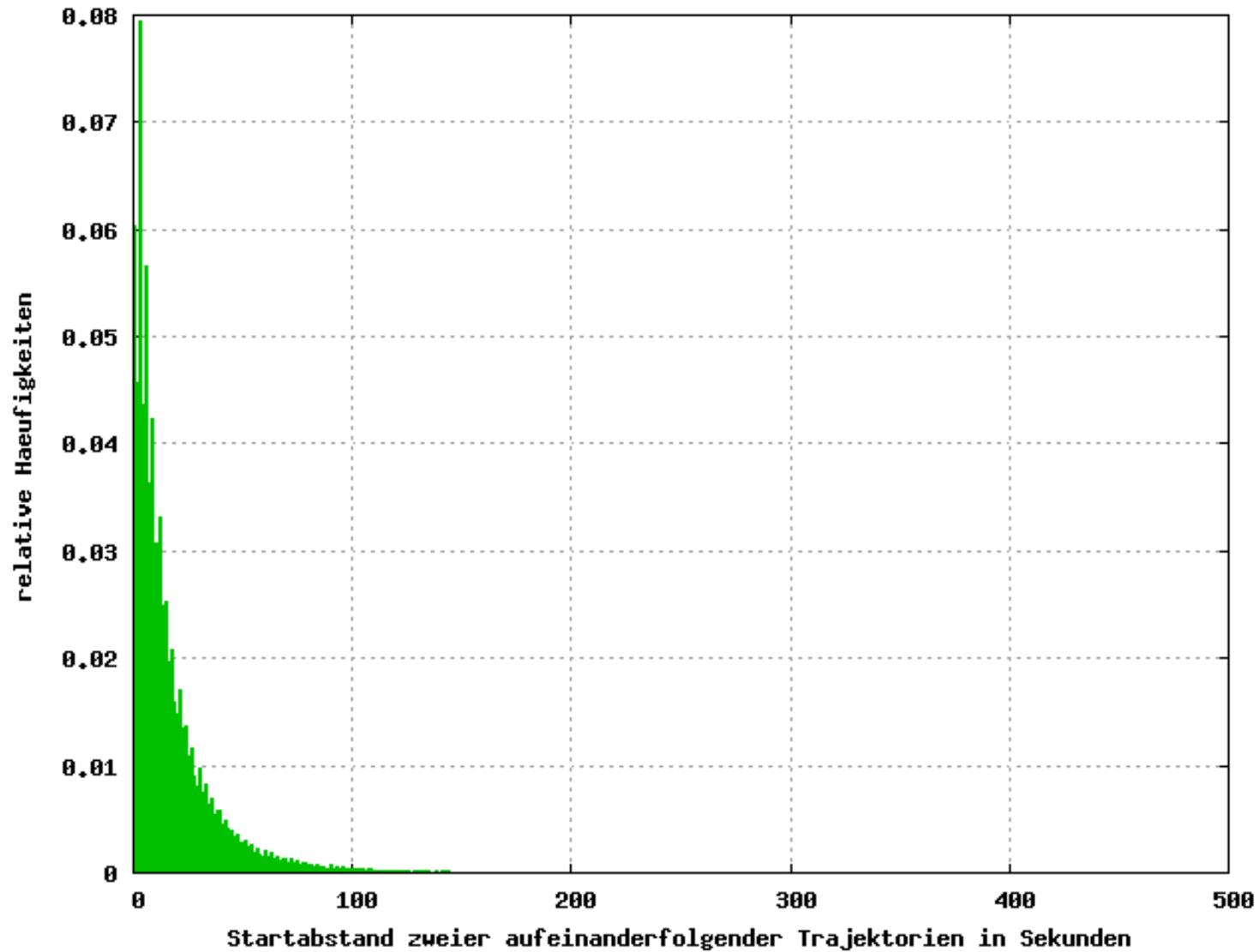
## *Winkeländerungen zwischen den Segmenten II*

- Mit hohen Wahrscheinlichkeiten fahren die Taxis in den beobachteten Daten tendenziell geradeaus (auf ihr Ziel zu)
- Überstumpfe Winkel sind sehr selten
- Es existiert ein kleiner Peak bei  $180^\circ$

Aus dieser Beobachtung läßt sich eine Heuristik ableiten: eine typische Trajektorie führt vergleichsweise direkt auf das Ziel zu

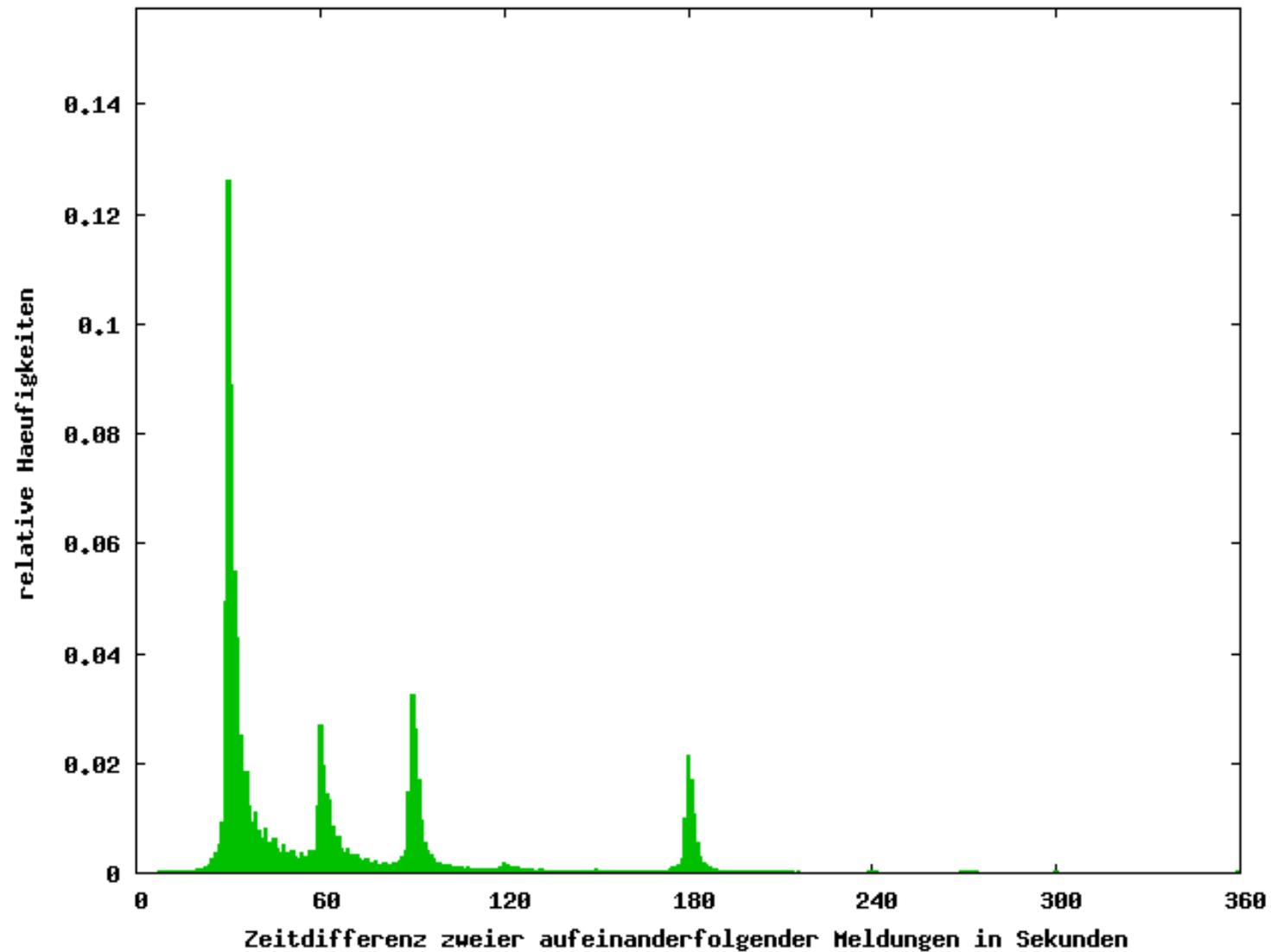
# Verkehrssimulation

## Startzeitdifferenzen



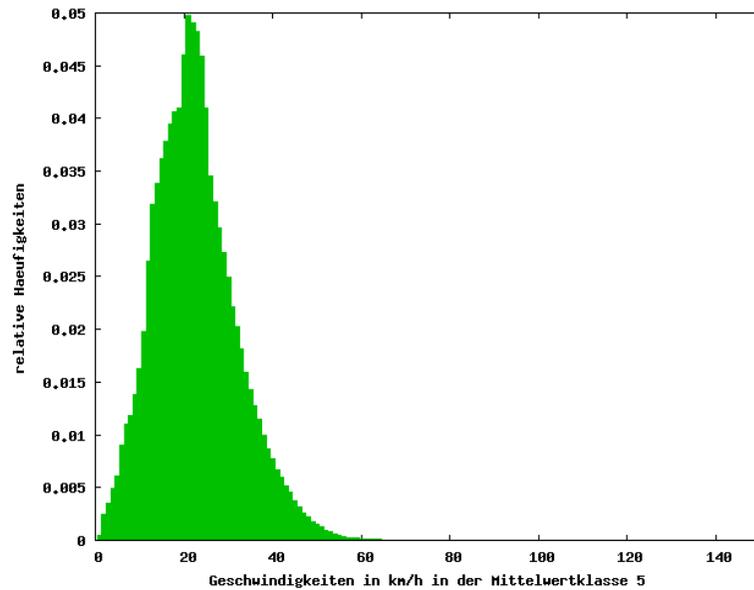
# Verkehrssimulation

## *Meldezeitpunktdifferenzen innerhalb einer Trajektorie*

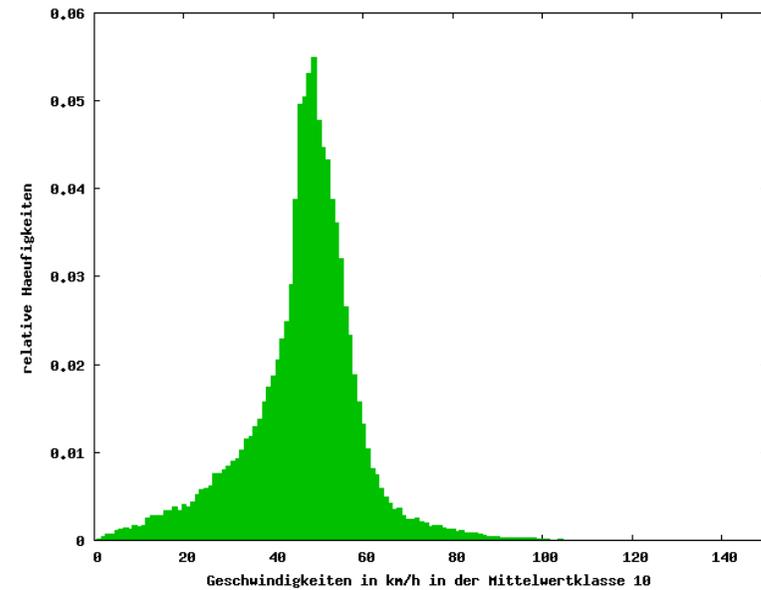


# Verkehrssimulation

## Mittelwertklassen



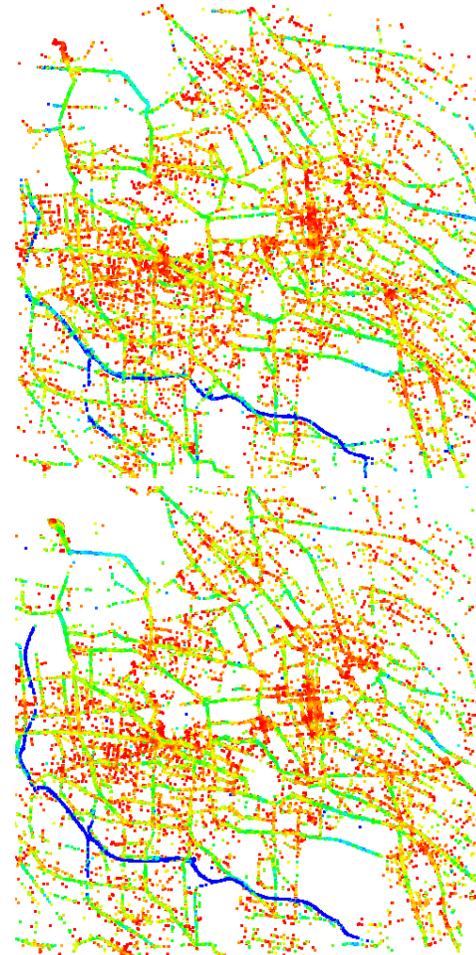
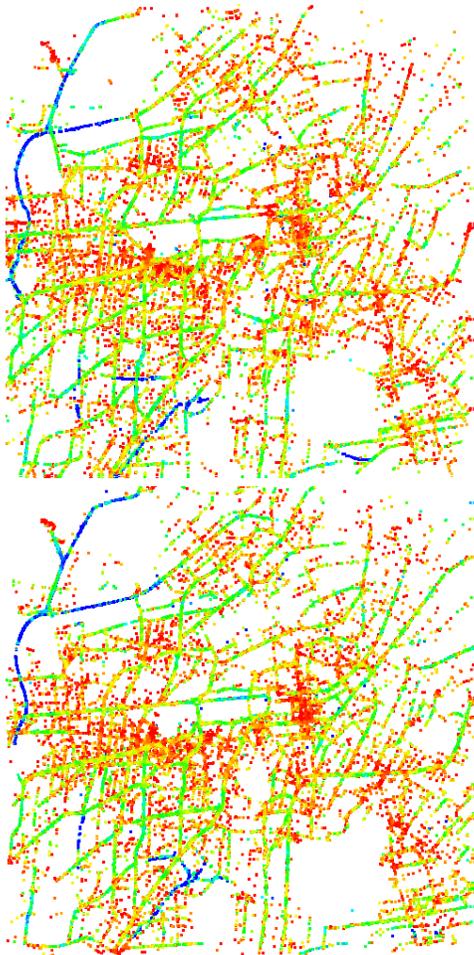
Mittelwertklasse 5: 20 bis 25  
km/h



Mittelwertklasse 10: 45 bis  
50 km/h

# Verkehrssimulation

## *Mittelwertbilder*



# Verkehrssimulation

## *Simulation I*

- Beobachtungsfenster ist ca. 12 km × 12 km groß
- der gerichtete Straßengraph liegt komplett im Beobachtungsfenster
- Startpunkt und Zielwinkel einer simulierten Trajektorie wird einer vorhandenen Trajektorie entnommen
- Schnitt der Geraden ausgehend vom Startpunkt unter dem Zielwinkel mit dem Beobachtungsfenster ist der (eigentlich unbekannt) Zielpunkt für die Simulation

# Verkehrssimulation

## *Simulation II*

- per Ameisenalgorithmus wird ein kurzer Weg vom Start zum Zielpunkt unter den bekannten Geschwindigkeitsmittelwerten gesucht (Heuristik der Kantenwahl: bevorzuge direkte Richtungen)
- der von den Ameisen gefundene Weg dient als Basis einer Trajektorie: Geschwindigkeiten und Meldezeitpunkte entlang des Weges werden zufällig ermittelt

# Verkehrssimulation

## *Beispiel einer Simulation*



# Verkehrssimulation

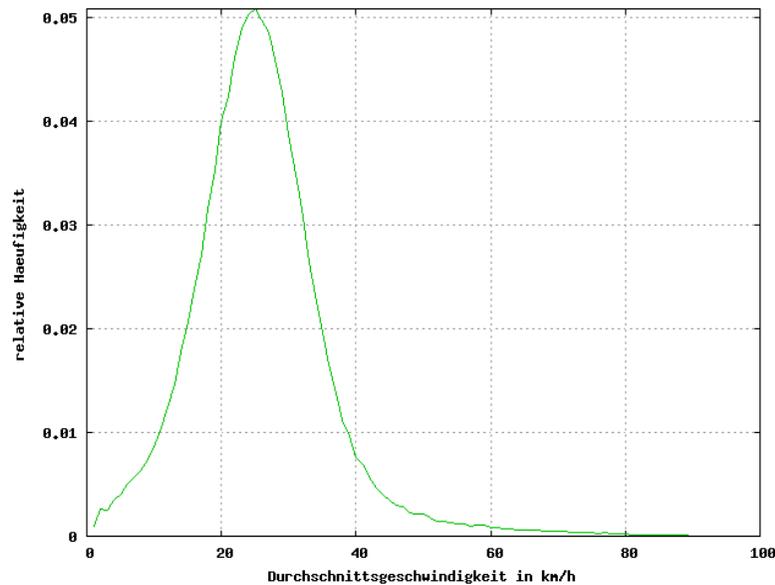
## *Auswertung der Simulation*

- die simulierten Daten werden exakt auf die gleiche Art ausgewertet wie die realen Datensätze
- ein Vergleich verschiedener Parameter zeigt gute Übereinstimmungen

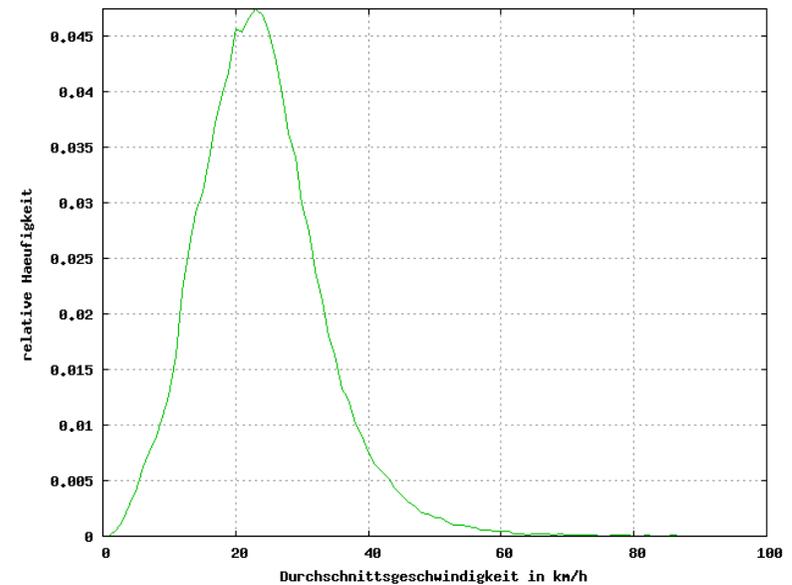
# Verkehrssimulation

## Vergleich I

### Vergleich der Durchschnittsgeschwindigkeiten je Trajektorie



reale Daten

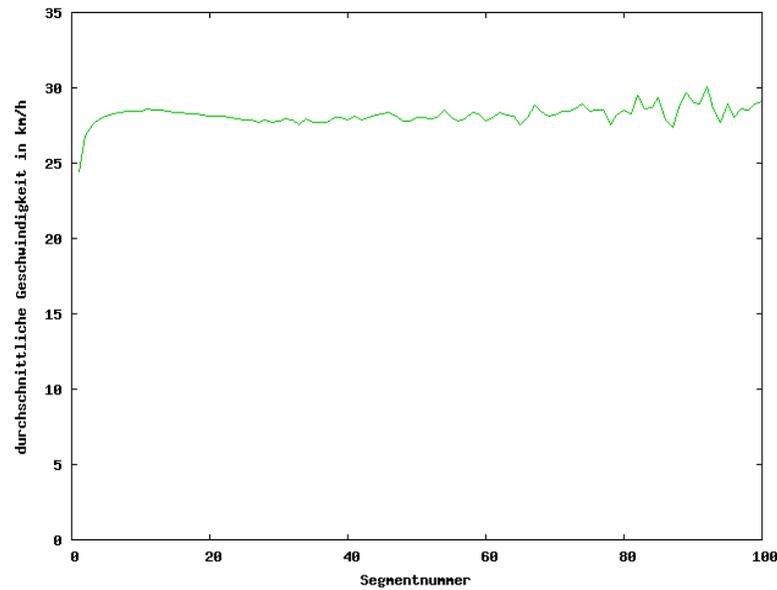


simulierte Daten

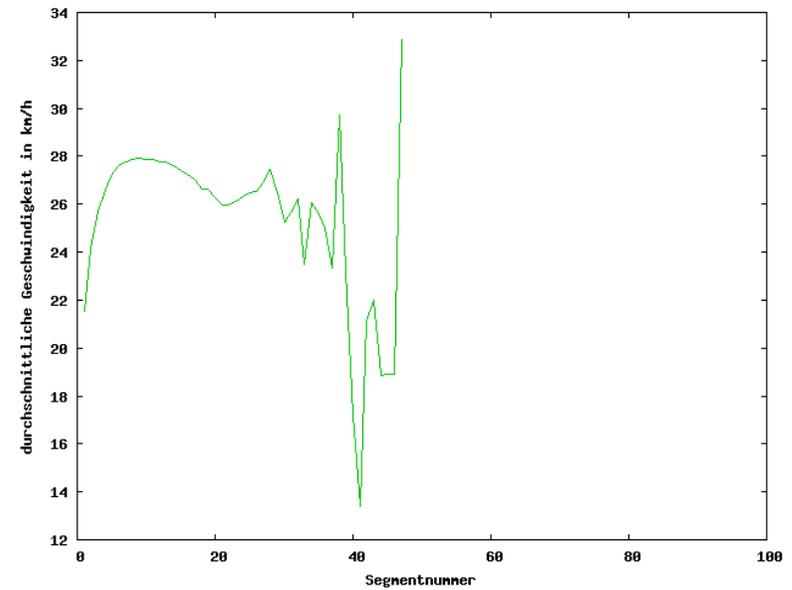
# Verkehrssimulation

## Vergleich II

### Vergleich der Durchschnittsgeschwindigkeiten je Segment

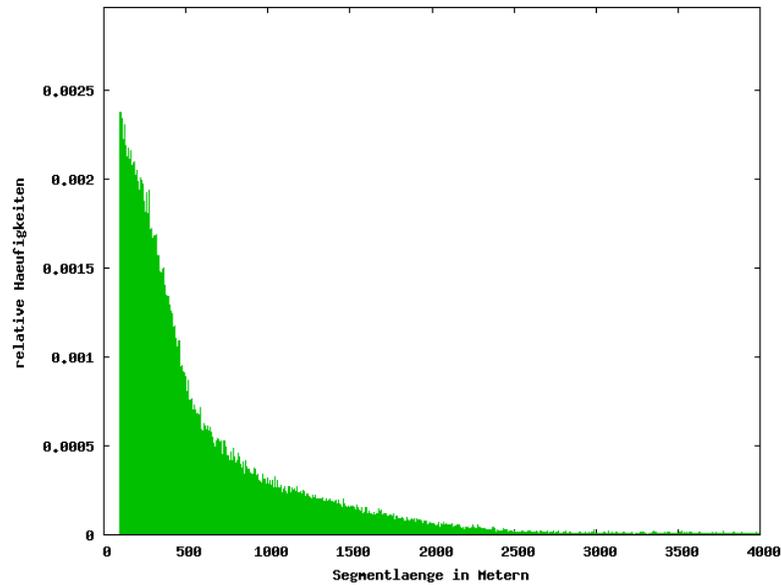


reale Daten

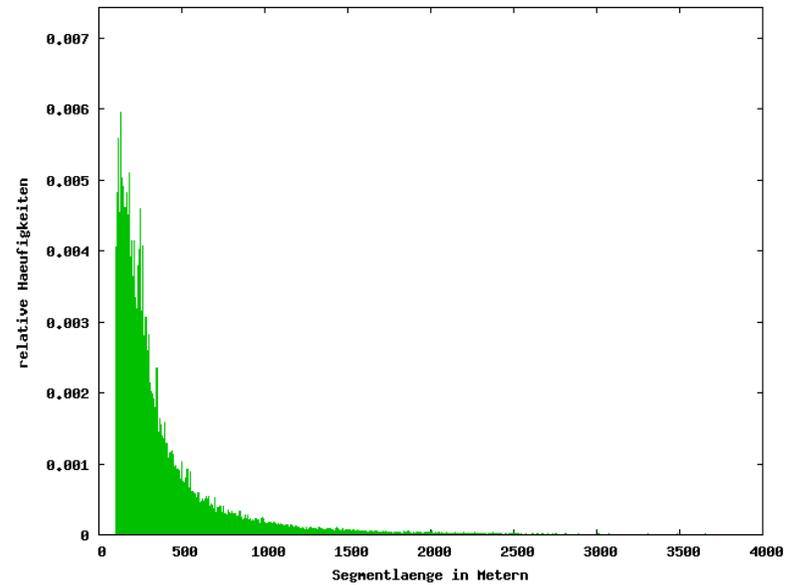


simulierte Daten

### Vergleich der Segmentlängen

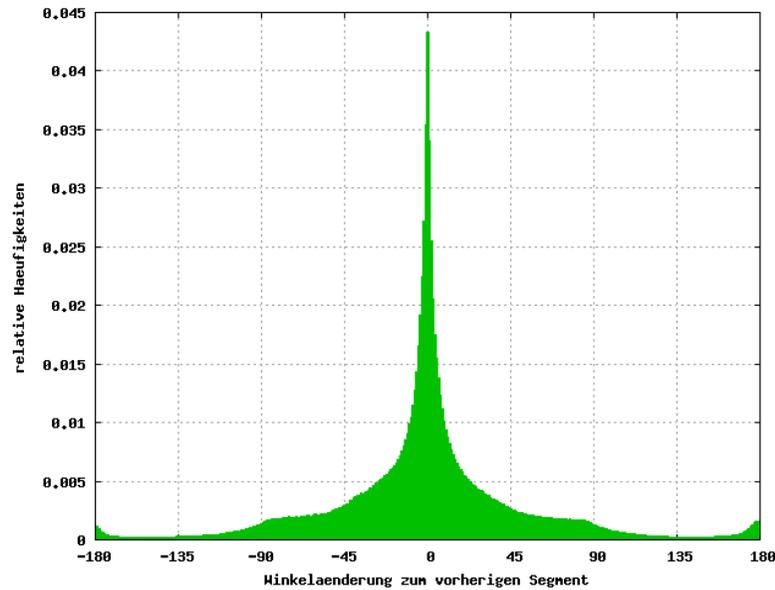


reale Daten

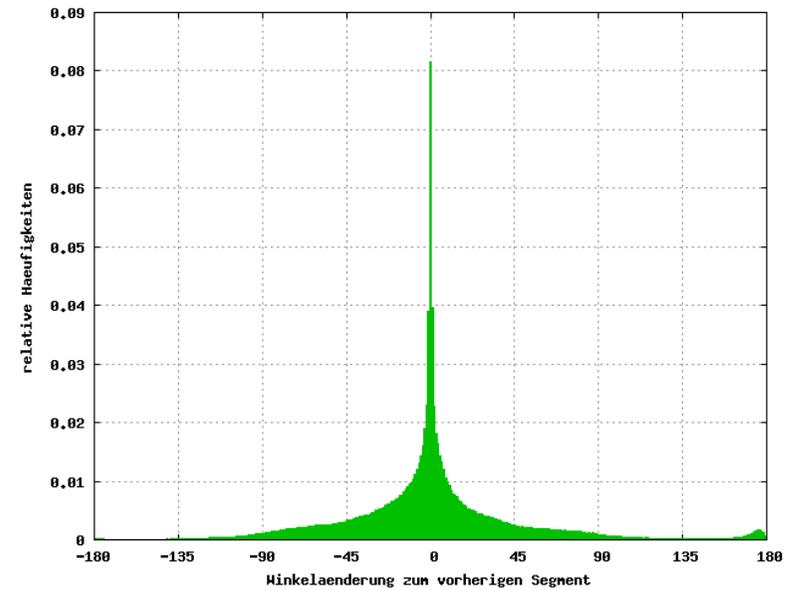


simulierte Daten

### Vergleich der Winkeldifferenzen



reale Daten



simulierte Daten

# Verkehrssimulation

## *Implementierung*

- Java und MySQL
- Objektorientierung ermöglicht durch Ableiten von Klassen einfache Erweiterungen
- geschlossenes System: Übertragung des Systems auf andere Städte oder Gebiete ohne weiteres möglich
- Simulation ist parallelisiert, die Anzahl der maximalen Threads kann vorgegeben werden
- Alle für die Simulation relevanten Parameter werden an zentraler Stelle vorgegeben

- Verknüpfung simulierter Daten und Geschwindigkeitsmittelwertkarten mittels Variogrammen
- Führt die Simulation durch Ameisenalgorithmen zu *besseren* Ergebnisse als andere Verfahren?

- Einbringen der Straßenhierarchie in den Graphen
- Optimierung der konstruierten Routen z.B. per Lokaler Suche
- Vergleich mit Daten aus anderen Großstädten
- Wegsuche nicht nur in Großstädten sondern allgemein

# Bibliographie

- Hromkovič, J.: Algorithmics for Hard Problems. Introduction to Combinatorial Optimization, Randomization, Approximation and Heuristics, Springer, Berlin, Heidelberg, New York et. al., 2001.
- Dorigo, M., Stützle, T., Ant Colony Optimization, MIT Press, Cambridge, London, 2004.
- Braxmeier, H. et al., Kriged road traffic maps, in: Springer Series: Advances in Spatial Science, S. 39 - 52, Berlin, 2005.

# Ende

Vielen Dank für die Aufmerksamkeit!