

# Ameisenalgorithmen

## Dynamische Fahrzeugnavigation

Seminar SS 2005 – Künstliche Intelligenz

Thomas Walther





1. Einleitung
2. Reale Ameisen und künstliche Ameisen
3. Ant Colonisation Optimisation
4. Ant Based Control - Routing System
5. Zusammenfassung Ausblick
6. Quellen

# Einleitung



# Einleitung – Von der Natur lernen

- Schwarmintelligenz
  - Vögel, Bienen, Fische, Ameisen, ...
  - Einzelne Individuen sind nur „bedingt“ intelligent
  - Kollektive Zusammenarbeit ermöglicht beachtliche Leistungen
  - Verhalten auf das Überleben der Kolonie ausgerichtet



# Einleitung – Von der Natur lernen

- Idee: Kollektive Zusammenarbeit von Fahrzeugen
  - Geschwindigkeiten, Verkehrsdichte, Reisezeiten
  - Allgemeine Datenweiterleitung, Internet auf der Straße
  - Staus/Unfälle erkennen und nachfolgenden Verkehr informieren
  - ....





# Agenda

---

## 2. Reale Ameisen

- Was können Ameisen?
- Wie koordinieren sich Ameisen?
- Ameisen und die Suche von Wegen
- Künstliche Ameisen

# Reale Ameisen – Was können Ameisen?

---

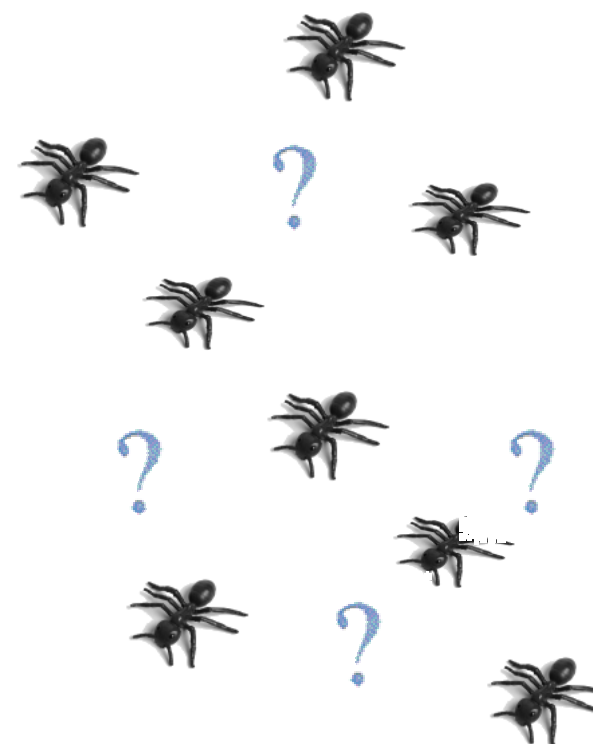
- Ameisen finden kürzeste Wege
  - Koordination der Futtersuche
  - Anpassung an Veränderungen der Umgebung
- Wege in Verkehrsnetzen
  - Kürzeste Route
  - Schnellste Route
  - Komfortabelste Route
  - Dynamische Informationen nutzen



# Reale Ameisen - Futtersuche

Wie können sich die Ameisen koordinieren?

- Indirekte Kommunikation mittels Pheromonen
- Probabilistische Entscheidungsfindung





# Reale Ameisen – Kommunikation

---

- **Pheromone**
  - Chemische Substanz
  - Permanente, konstante Ausschüttung
  - Verändern die lokale Umgebung
  - Stimuli für einzelne Ameisen
- **Stigmergie**
  - Indirekte Kommunikation über Pheromone
  - Pheromone nur lokal ablesbar
  - Verteilte Information

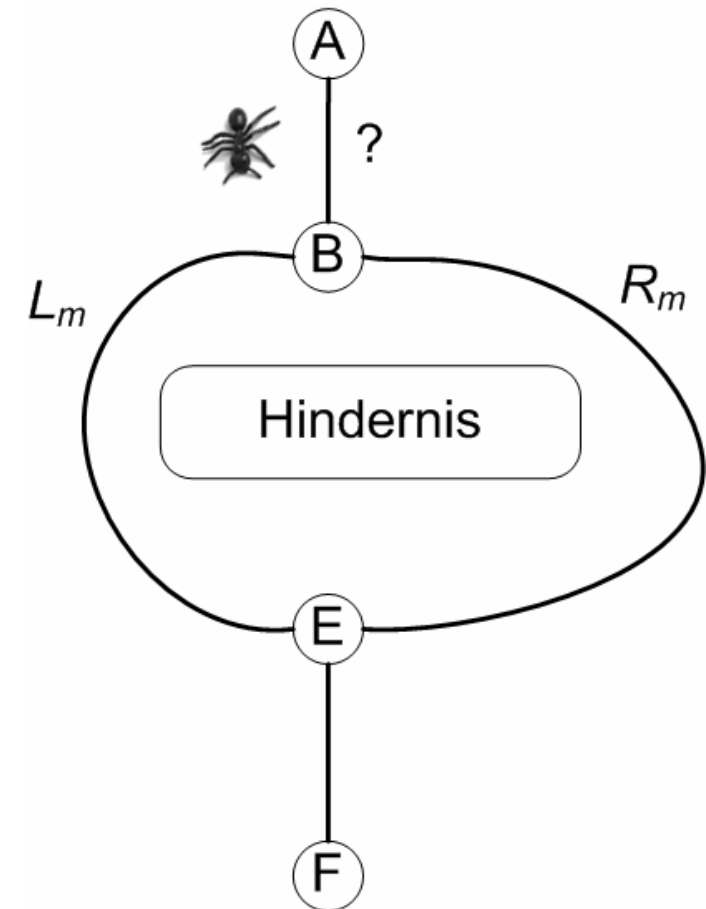


- Experiment: Binäre Brücke
  - Annahme: keine Verdunstung
  - $m$  ... Anzahl der Ameisen
  - $L_m$  ... Ameisen auf linken Weg
  - $R_m$  ... Ameisen auf rechten Weg

$$P_L(m) = \frac{(L_m + k)^h}{(L_m + k)^h + (R_m + k)^h}$$

$$P_R(m) = 1 - P_L(m)$$

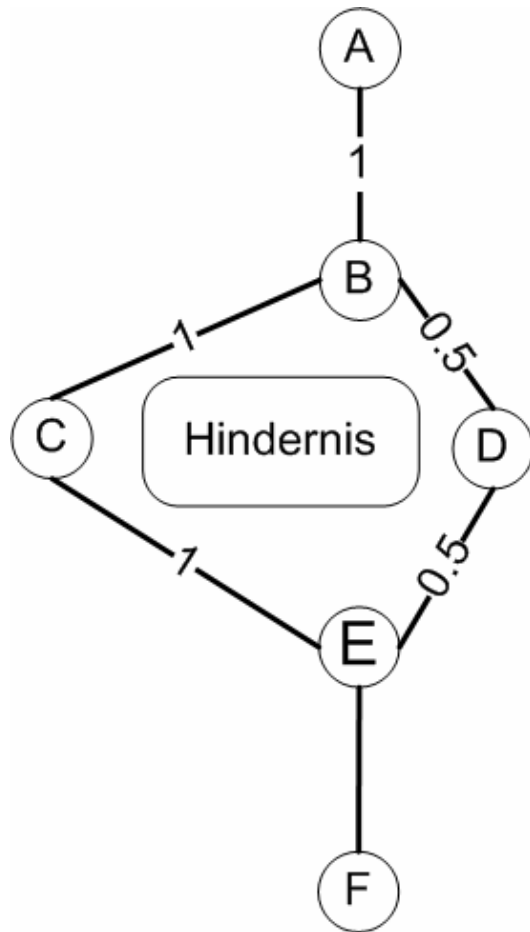
$$k = 20; h = 2$$



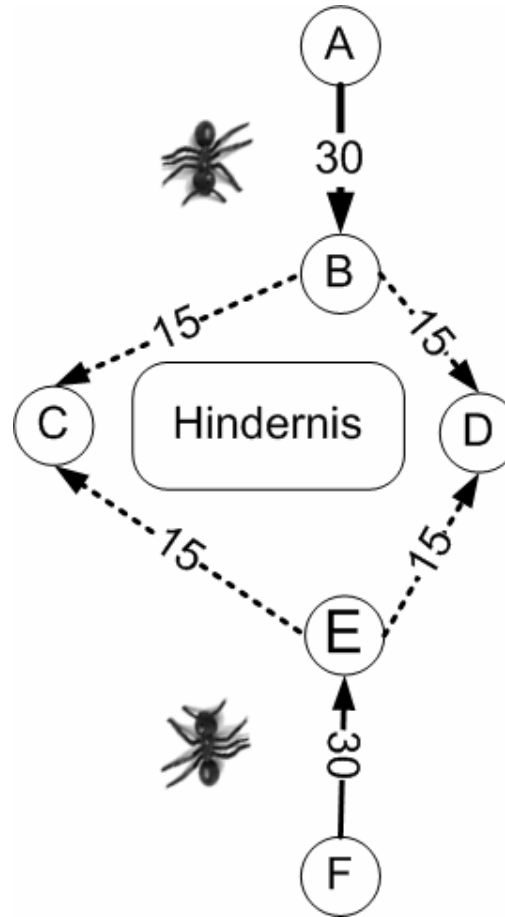
# Reale Ameisen – Kürzester Weg



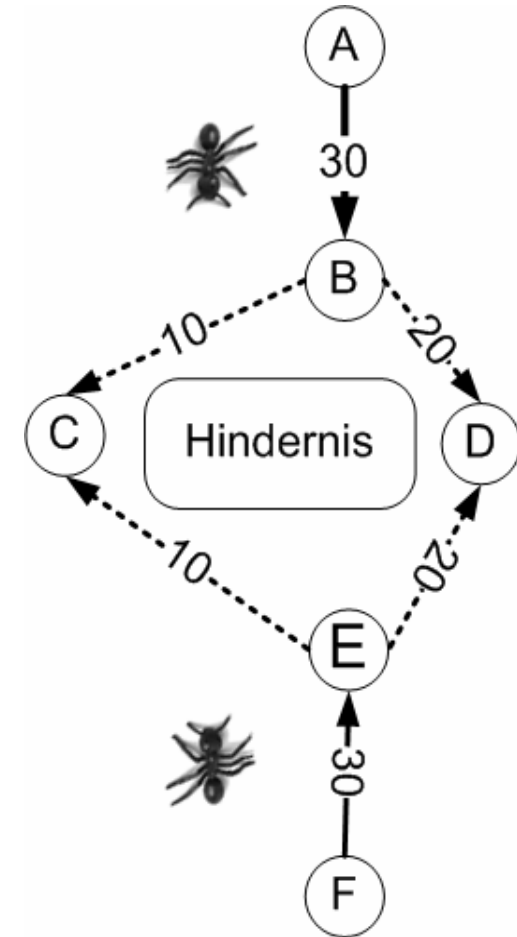
Entfernungen



T=0



T=1





# Reale Ameisen – Kürzester Weg

---

- **Autokatalyse**
  - Positives Feedback durch Pheromone
  - Je höher die Pheromonkonzentration ist, desto mehr Ameisen werden diesen Weg wählen.
- **Implizierte Problemlösung**
  - Je kürzer ein Weg ist, desto mehr Ameisen durchlaufen diesen in gleicher Zeit, desto schneller steigt die Pheromonkonzentration
- **Kombination beider Eigenschaften**
  - Je kürzer ein Weg ist, desto mehr Ameisen werden diesen Weg gehen.



# Reale Ameisen - Verdunstung

---

- **Problem: Stagnation**
  - Gute Lösungswege wirken anziehend
  - Gefahr: schlechte Ameisen finden durchschnittliche Lösung
  - Algorithmus konvergiert vorschnell
- **Lösung: Pheromonverdampfung**
  - Differenz der Pheromonkonzentration wird begrenzt
  - Kompromiss zwischen Verstärkung und Erforschung



# Reale Ameisen und Künstliche Ameisen

---

## Gemeinsamkeiten

- Kolonie von kooperierenden Individuen
- Pheromonspur und Stigmergie
- Kürzeste Wege – Kostenminimierung
- Kontinuierliche Bewegungen
- Stochastische und „kurzsichtige“ Entscheidungsfaktoren

## Unterschiede

- Diskrete Welt, diskrete Zustandsübergänge
- Zustandsvariable, Gedächtnis
- Pheromonausschüttung korreliert mit Lösungsgüte
- Oftmals verzögerte Pheromonausschüttung
- Problemspezifische Fähigkeiten



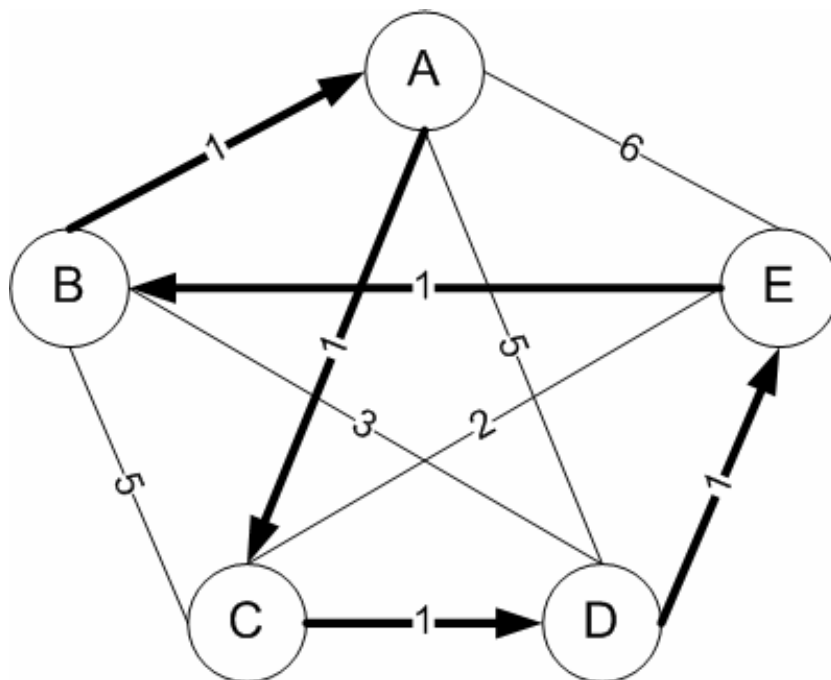
# Agenda

---

3. Ant Colonisation Optimisation
  1. AntSystem und TSP
  2. Ant Based Control

# ACO – AntSystem und TSP

- Dorigo et al. (1991), erster ACO – Algorithmus
- Grundlage für viele weitere Algorithmen
- Problemstellung: Travelling Sales Person (TSP)



# ACO – AntSystem und TSP

Relative Entscheidungsbasis im Knoten i

$$a_{ij}(t) = \frac{\tau_{ij}^\alpha n_{ij}^\beta}{\sum_{l \in N_i} \tau_{il}^\alpha n_{il}^\beta}$$

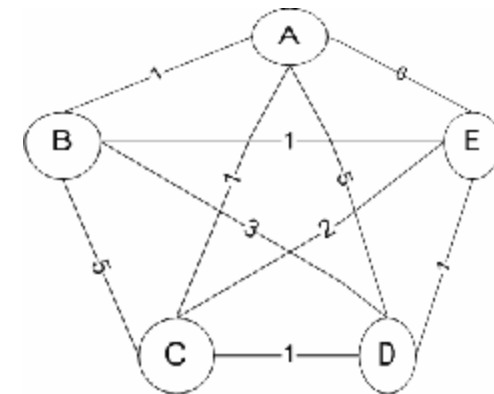
$\tau_{ij}$  ... Pheromonkonzentration der Kante  $e(i,j)$

$d_{ij}$  ... Länge der Kante  $e(i,j)$

$n_{ij} = \frac{1}{d_{ij}}$  ... Heuristik

Entscheidungsfindung im Knoten i

$$P_{ij}^k(t) = \frac{a_{ij}(t)}{\sum_{l \in N_i^k} a_{il}(t)}$$



Pheromonausschüttung (nachträglich) im Knoten i

$$\Delta\tau_{ij} = \begin{cases} \frac{1}{L_k} & \text{falls } (i,j) \in T_k \\ 0 & \text{sonst} \end{cases}$$

$L_k$  ... Tourlänge der Lösung von Ameise k

$p$  ... Verdampfungsfaktor

$$\tau_{ij} = (1-p)\tau_{ij}(t) + \Delta\tau_{ij}$$



# ACO – Ant Based Control

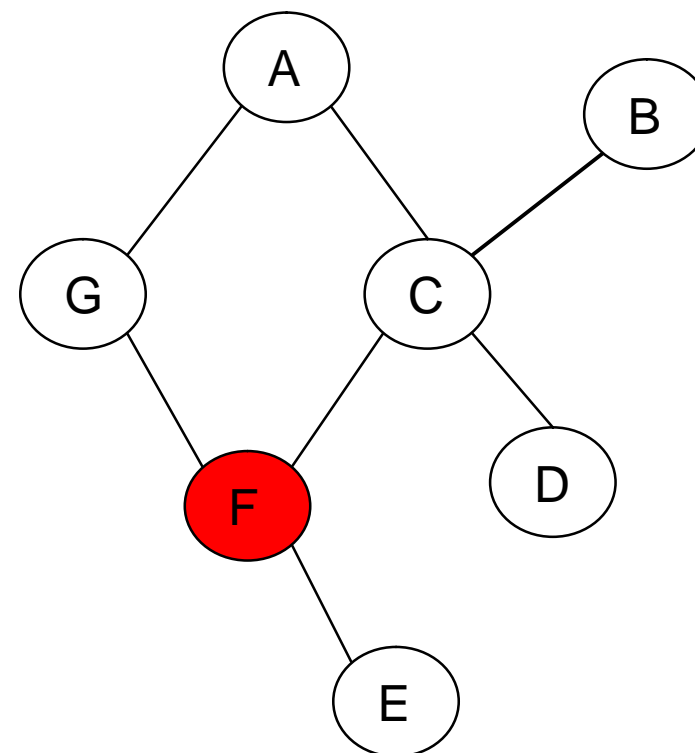
---

- Van der Put and Rothkrantz (1998)
- Routingtabellen in Telefonnetzen
- Eigenschaften:
  - Verzögerte Pheromonausschüttung
  - Pheromonverdampfung
  - Heuristik wird nicht verwendet
- 2 Ameisentypen
  - Vorwärts-Ameise: Routesuche, Informationsgewinnung
  - Rückwärts-Ameise: Routingtabellen aktualisieren

# ACO – Ant Based Control

## Routingtable basierend auf Pheromonen

Next \ Dest	C	G	E
A	0.3	0.65	0.05
B	0.6	0.35	0.05
C	0.9	0.05	0.05
D	0.9	0.05	0.05
E	0.9	0.05	0.05
G	0.6	0.35	0.05





# ACO – Ant Based Control

Erhöhung der Pheromonkonzentration

$$P_{d,f} = \frac{P_{d,f} + \Delta P}{1 + \Delta P}$$

$$\Delta P = \frac{A}{t} + B$$

d ... Zielknoten der Vorwärtsameise  
f ... Knoten über den die Rückwärtsameise den  
aktuellen Knoten erreicht hat

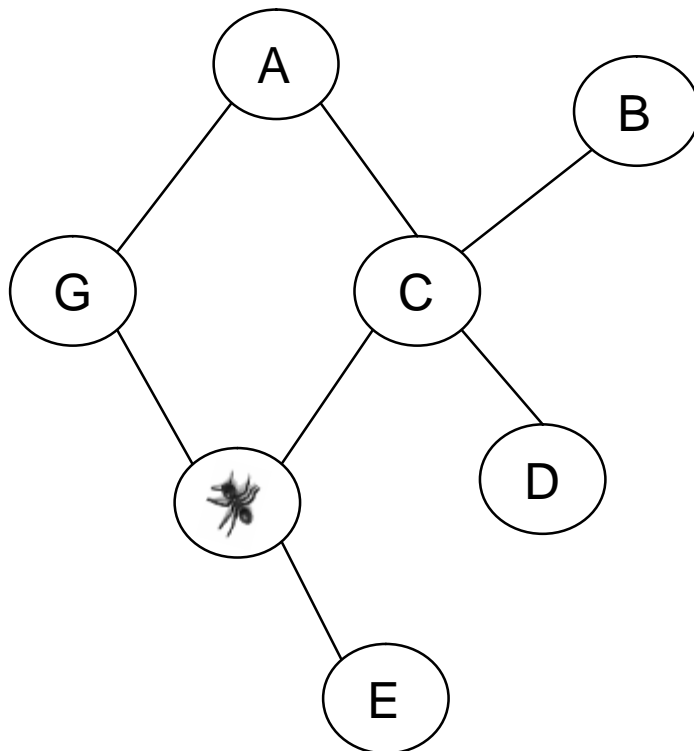
Pheromon – Verdampfung

$$P_{d,i} = \frac{P_{d,i}}{1 + \Delta P} \quad \forall i \neq f$$

Next \ Dest	C	G	E
A	0.3	0.65	0.05
B	0.6	0.35	0.05
C	0.9	0.05	0.05
D	0.9	0.05	0.05
E	0.9	0.05	0.05
G	0.6	0.35	0.05

# ACO – Ant Based Control

## Ablauf des Algorithmus - Vorwärtsameise



### Gedächtnis

$\{s,d\} = \{F,B\}$

$\{k,t_k\} = \{F,0.5\} \{C,1.5\}$

Tabelle F (verkürzt)

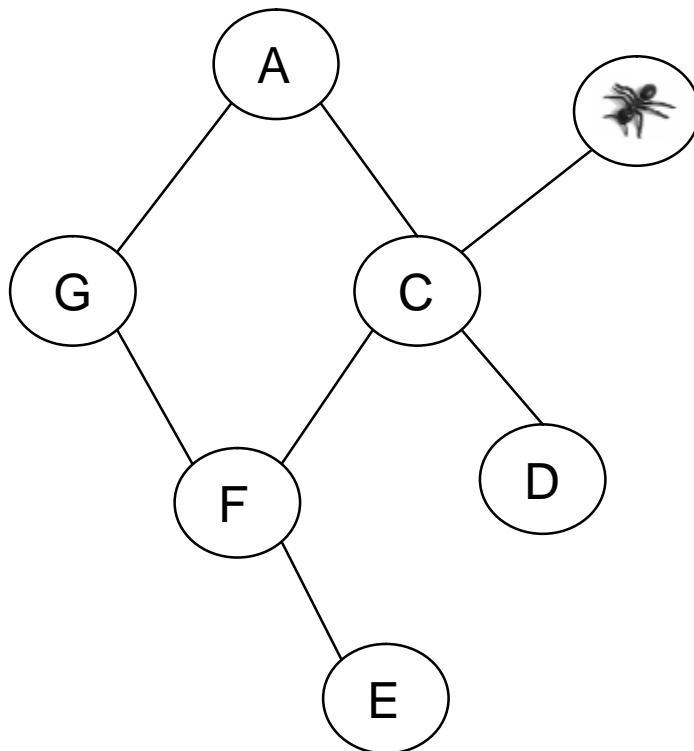
Next \ Dest	C	G	E
B	0.5	0.35	0.15

Tabelle C (verkürzt)

Next \ Dest	A	B	D	F
B	0.05	0.85	0.05	0.05



## Ablauf des Algorithmus - Rückwärtsameise



### Gedächtnis

$$\{s,d\} = \{F,B\}$$

$$\{k,t_k\} = \{F,0.5\} \quad \{C,1.5\}$$

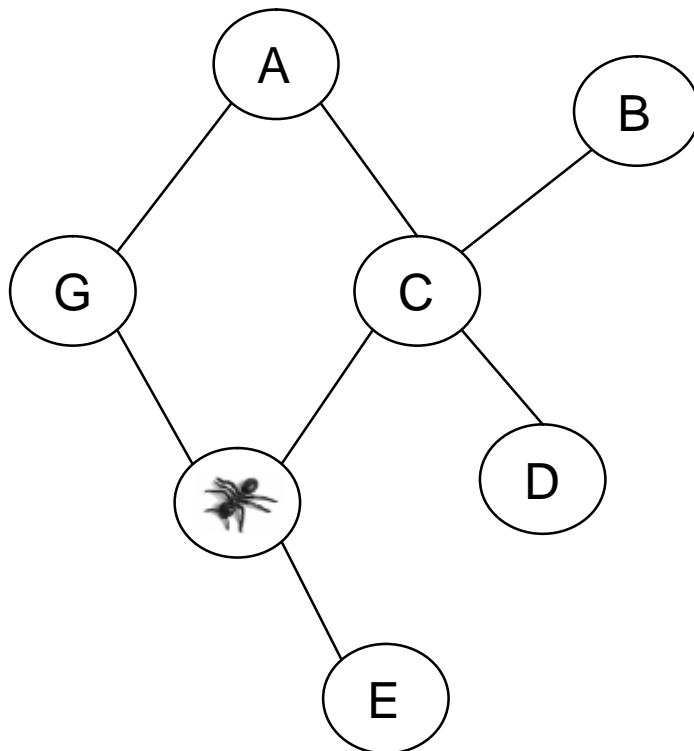
Tabelle C (verkürzt)

Next \ Dest	A	B	D	F
B	0.05	0.85	0.05	0.05

Pheromonupdate verändert die Tabelle des Knotens C nicht, da als minimaler Wert für jeden Eintrag 0.05 erhalten bleiben soll.

# ACO – Ant Based Control

## Ablauf des Algorithmus - Rückwärtsameise



Gedächtnis  
 $\{s,d\} = \{F,B\}$   
 $\{k,t_k\} = \{F,0.5\} \{C,1.5\}$

Tabelle F (verkürzt)			
Next \ Dest	C	G	E
B	0.5	0.35	0.15

Tabelle F (verkürzt)			
Next \ Dest	C	G	E
B	0.83	0.12	0.05

$$\Delta P = \frac{2}{2} + 1$$

$$P_{\text{new},C} = \frac{0.5 + 2}{1 + 2} = 0,83$$

$$P_{\text{new},G} = \frac{0.35}{1 + 2} = 0,12$$

$$P_{\text{new},E} = \frac{0.15}{1 + 2} = 0,05$$



# Agenda

---

4. ABC – System zur Fahrzeugnavigation
  1. Globales System Design
  2. Timetable Update System
  3. Route Finding System
  4. Verteiltes Routing System

Quelle:

Kroon R., *Dynamic vehicle routing using Ant Based Control*,  
Master's thesis, Delft University of Technology, 2002.

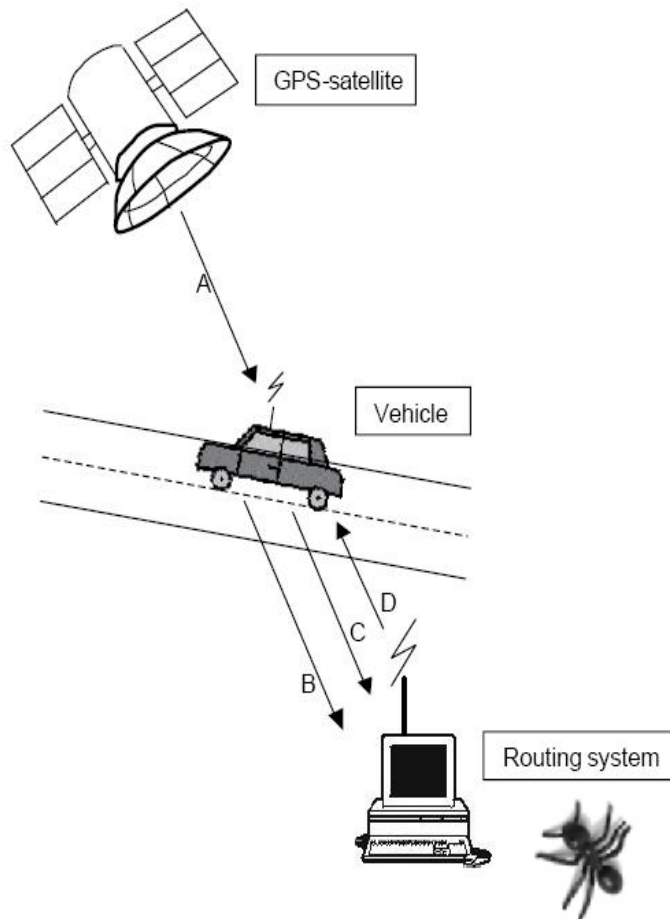


# ABC – System

---

- Ziele
  - Qualität: momentan schnellste Route anbieten
  - Geschwindigkeit: Echtzeitanfragen beantworten
  - Erweiterbarkeit
  - Adaptiv
- Optimierungsziel
  - Fahrzeit, weitere Kriterien werden nicht berücksichtigt
- Ant Based Control
  - Verteiltes, dynamisches System
  - Optimierung durch kollektives Verhalten

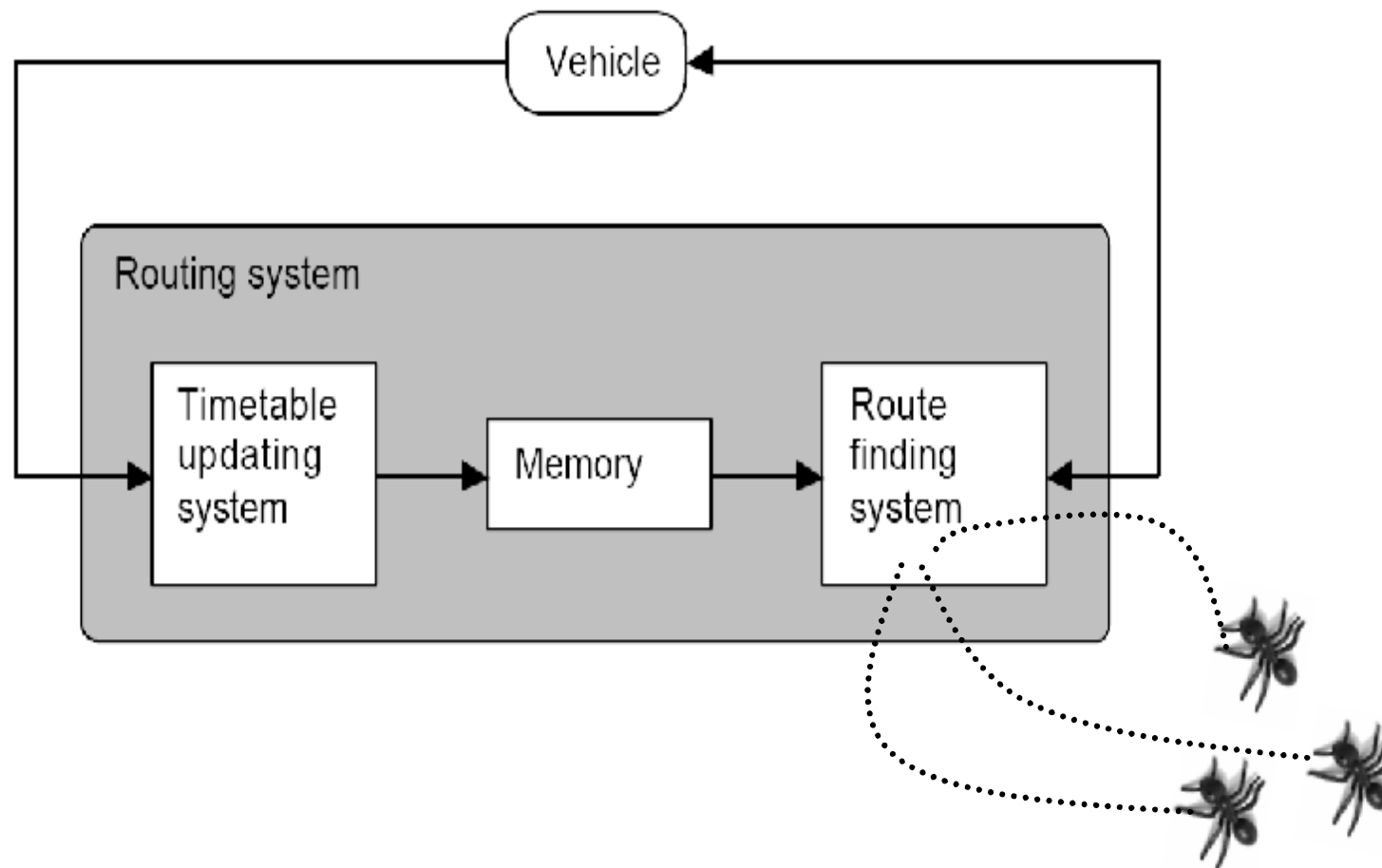
# ABC – System Global Design



	From	To	Data
Arrow A	GPS-satellite	Vehicle	SEND_POSITION, latitude/longitude co-ordinates
Arrow B	Vehicle	Routing system	UPDATE, previous time/position, covered road A, covered road B, covered road C, ..., current time/position
Arrow C	Vehicle	Routing system	REQUEST_ROUTE, current position, destination
Arrow D	Routing system	Vehicle	ANSWER_ROUTE, road A, road B, road C, ...

# ABC – System Global Design

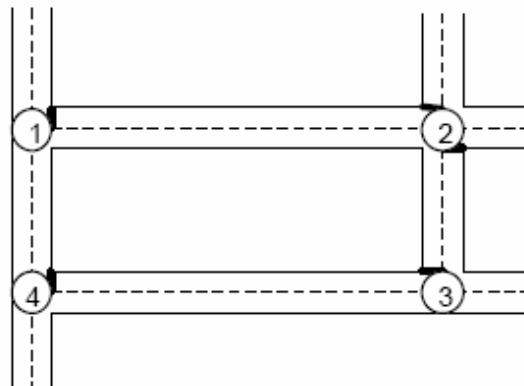
## Routingssystem Struktur



# ABC – System Global Design

## Timetable Update System

- Erstellung einer Zeittabelle als Basis für Routensuche
- Erfassung der aktuellen Verkehrslage
- Informationsquellen: Fahrzeuge selbst
- Erfassung von gemittelten Fahrtzeiten
  - Fahrzeuge senden in regelmäßigen Abständen Informationen über die gefahrene Strecke und die Fahrzeit



From:	To:	Intersection 1	Intersection 2	Intersection 3	Intersection 4
Intersection 1			25 s		13 s
Intersection 2		26 s		18 s	
Intersection 3			18 s		27 s
Intersection 4		12 s		25 s	

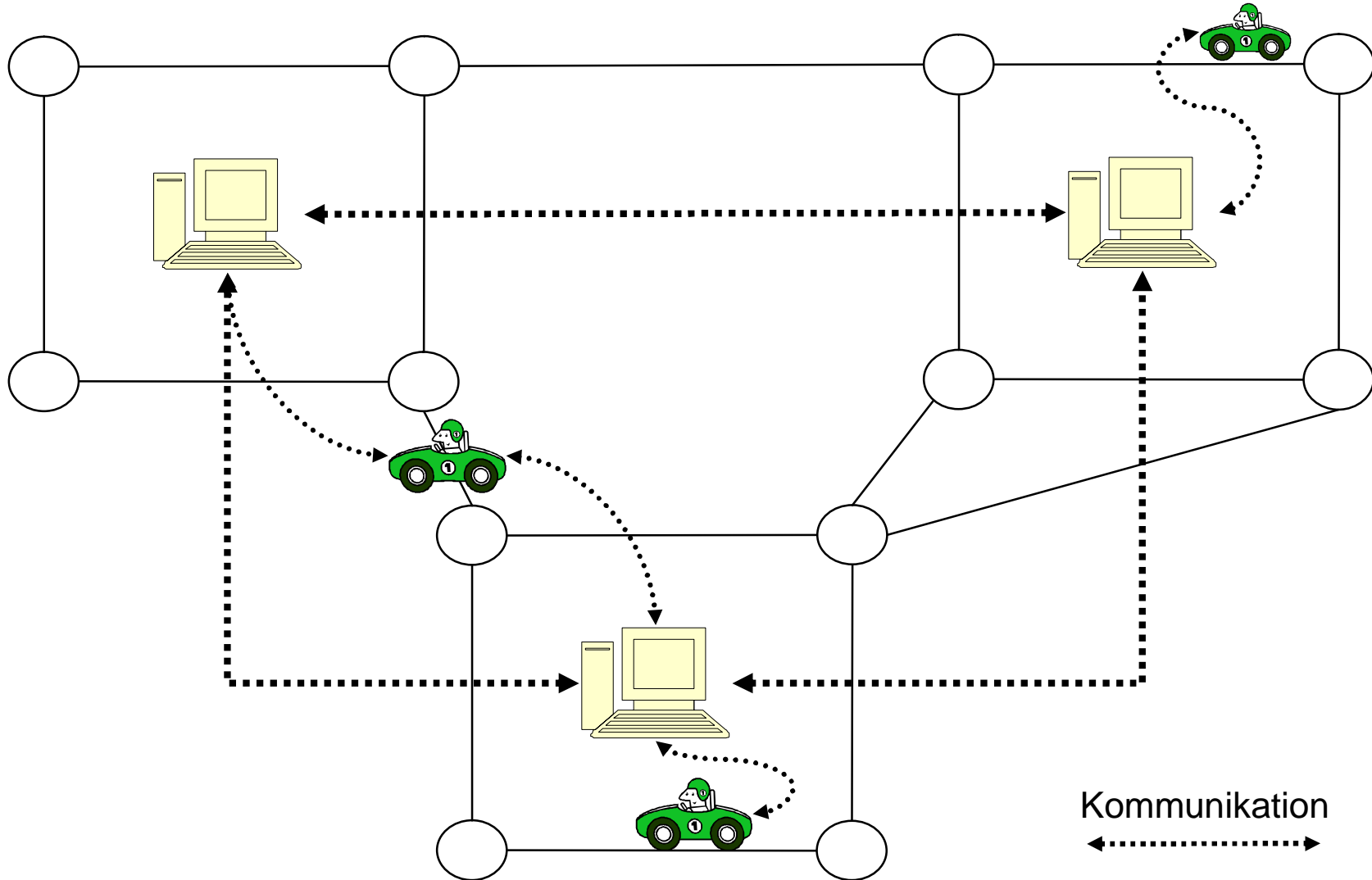


## Route Finding System

- Ermittlung der schnellsten Wege mittels ABC
  - Ständige Generation von Ameisen mit zufälligen Zielen
  - Triggerung durch das Timetable System
  - „Reisezeiten“ der Ameisen werden aus Zeittabellen errechnet
  - Pheromonbasierte Routingtabellen in jedem Verkehrsknoten
- Empfangen und Beantworten von Routenanfragen
  - Fahrzeugen wird immer der Weg mit der höchsten Wahrscheinlichkeit empfohlen
  - Rückgabe von Teilrouten, Fahrzeuge erfragen Aktualisierung



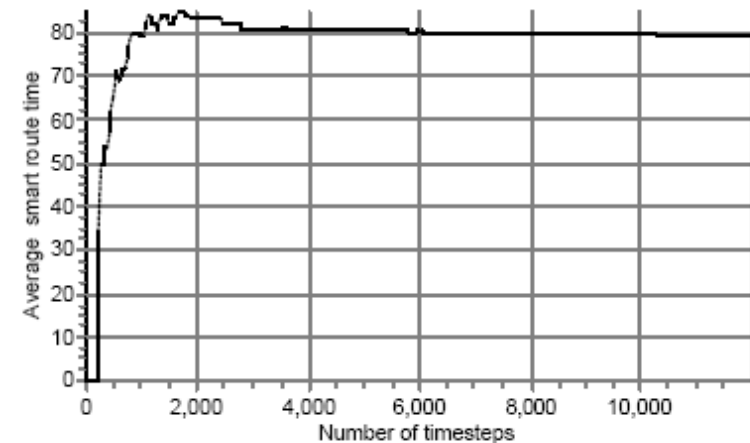
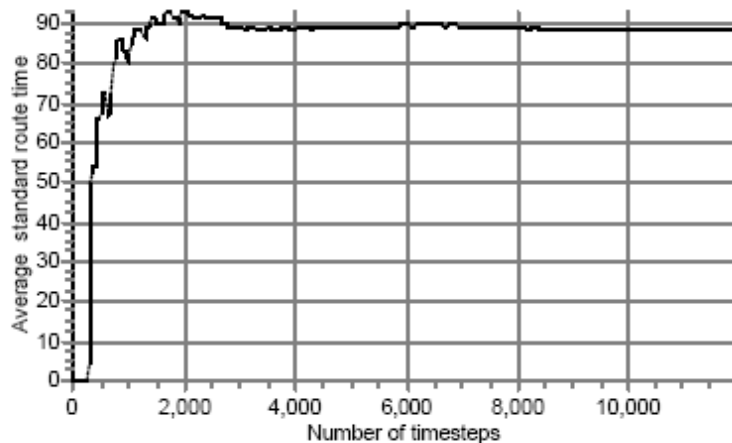
# Verteiltes ABC Routing System



# ABC – System Ergebnisse

## Effektivität

- Zeitgewinn für intelligent geleitete Fahrzeuge
- Positive Rückwirkung auf statisch geleitete Fahrzeuge
  - Wird nur statisch gerechnet beträgt die mittlere Fahrzeit 128 Simulationsschritte



# ABC – System Ergebnisse



## Stabilität vs. Adaptivität

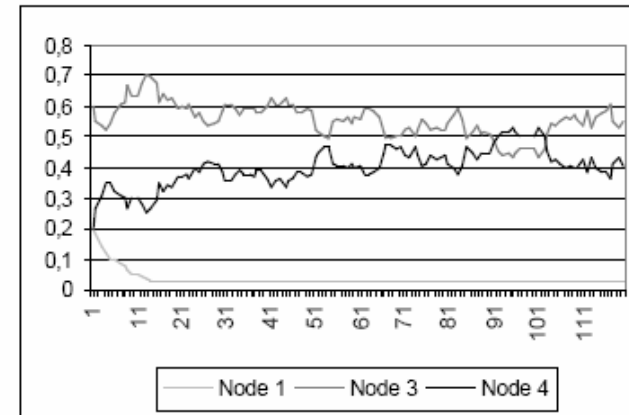
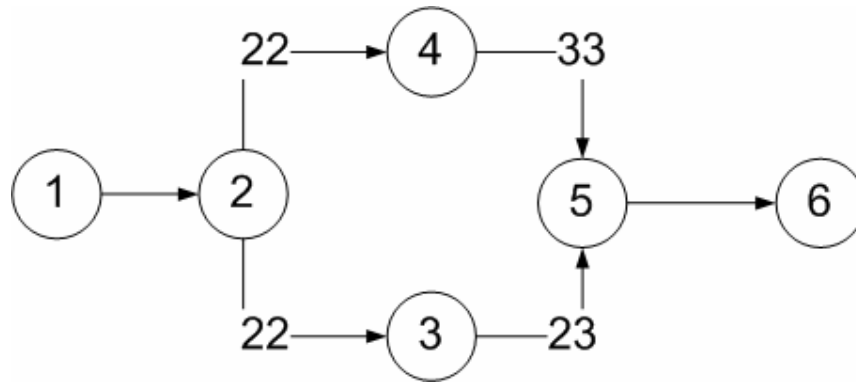


Figure 83: Probabilities for destination 6 ( $A = 1, B = 0.01$ )

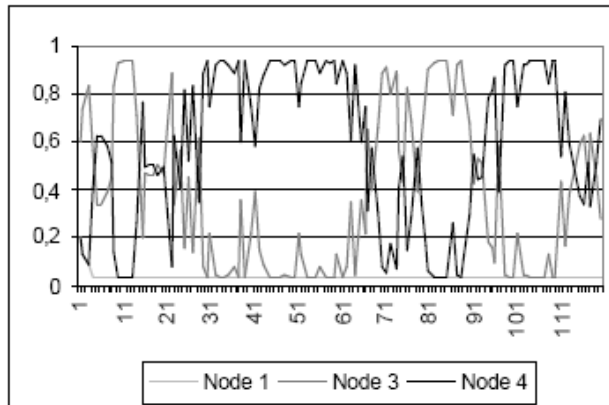


Figure 85: Probabilities for destination 6 ( $A = 32, B = 0.01$ )

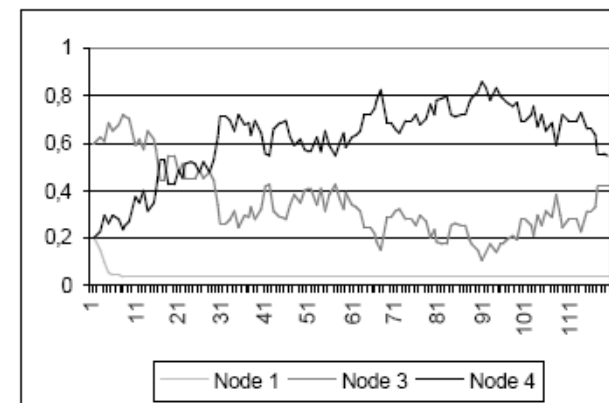


Figure 84: Probabilities for destination 6 ( $A = 4, B = 0.01$ )



# Zusammenfassung

---

- Ameisenalgorithmen
  - Einfache einzelne Einheiten bilden eine Kolonie
  - Kollektives Verhalten ermöglicht die Lösung komplexer Probleme
  - Indirekte Kommunikation – Stigmergie
  - Verschiede statische und dynamische Probleme mit ACO gelöst
- ABC System zur Fahrzeugnavigation
  - Reisezeiten werden direkt von Fahrzeugen erfasst und zur dynamischen Routenberechnung genutzt
  - Rechnersystem simuliert Ameisenkolonie zur Berechnung der momentan schnellsten Wege

# Ausblick

- Fahrzeugklassen berücksichtigen
- Integration/Kombination verschiedener Optimierungsziele
- Integration von historischen Daten
- Anbindung an einen komplexeren Simulator
- Simulation/Implementierung einer mobilen Anwendung
  - Adhoc Netzwerke zur Kommunikation
  - Fahrzeuge bieten verschiedene Services an





# Quellen

---

- Dorigo M., G. Di Caro & L. M. Gambardella (1999). Ant Algorithms for Discrete Optimization. *Artificial Life*, 5(2):137-172.
- Kroon R., *Dynamic vehicle routing using Ant Based Control*, Master's thesis, Delft University of Technology, 2002.
- R. Schoonderwoerd, O. Holland, and J. Bruten. Ant-like agents for load balancing in telecommunications networks. In *Proceedings of the First International Conference on Autonomous Agents*, pages 209–216. ACM Press, 1997.
- <http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html>
- Gianni Di Caro, An Introduction to Swarm Intelligence and Metaheuristics for Combinatorial Optimization: lecture slides [http://www.idsia.ch/~gianni/my\\_lectures.html](http://www.idsia.ch/~gianni/my_lectures.html)



Vielen Dank für Ihre Aufmerksamkeit  
Fragen?



# Agenda

---

3. Ant Colonisation Optimisation
  1. ACO meta-heuristic
  2. AntSystem und TSP
  3. Ant Based Control



# ACO meta-heuristic

---

- Algorithmische Struktur auf Basis kollektiven Verhaltens
- Lösen von dynamischen und statischen Problemen
- Lösung ist ein „kürzester“ Weg durch den Lösungsraum
- ACO ist geeignet wenn:
  - Verteiltes Problem
  - Dynamische Problemstellung
- ACO weniger geeignet wenn:
  - Sehr große Wertebereich der einzelnen Lösungsvariablen
  - Große Anzahl möglicher Schritte führt zu wirkungsloser Pheromonausschüttung



# ACO – Algorithmische Struktur

---

```
procedure ACO_Meta_heuristic()  
  while (! termination_criterion_satisfied)  
    ants_generation_and_activity();  
    pheromone_evaporation();  
    daemon_actions(); /* optionale globale Optimierung */  
  endwhile  
end procedure  
  
procedure ants_generation_and_activity()  
  while (available_resources) /* Anzahl der Ameisen pro Iteration */  
    new_active_ant();  
  end while  
end procedure
```



# ACO – Algorithmische Struktur

---

```
procedure new_active_ant()  
  while (current_state != target_state) /* Lösung gefunden */  
    move_ant() /* Zustandsübergang, Lösungsvariable belegen */  
    if (online_step_by_step_pheromone_update)  
      deposit_pheromone_on_the_visited_arc();  
      update_ant_routing_table();  
    end if  
  end while  
  if (online_delayed_pheromone_update)  
    evaluate_solution();  
    deposit_pheromone_on_all_visited_arcs();  
    update_ant_routing_table();  
  end if  
end procedure
```



# ACO – Algorithmische Struktur

---

```
procedure move_ant()  
  A = read_local_ant_routing_table();  
  P = compute_transition_probabilities(A;M;problem_constraints);  
  next_state = apply_ant_decision_policy(P; problem_constraints);  
  move_to_next_state(next_state);  
end procedure
```



# Einleitung – Von der Natur lernen

---

- Soziale Insekten bilden Staaten
  - Gemeinsamer Nutzen und Arbeitsteilung
  - Verhalten auf das Überleben der Kolonie ausgerichtet
  - Kollektives Verhalten ermöglicht komplexe Leistungen
- Ameisen
  - Brückenbau
  - Nestbau und Pflege des Nestes
  - Gemeinsame Futtersuche – kürzeste Wege
  - ...



# ACO - AntSystem und TSP

---

## Problem:

- AntSystem konvergiert bei vielen Städten zu schnell
- Schlechtere Lösung als Vergleichsalgorithmen

## Lösung:

- Problemspezifisch angepasste Algorithmen
- ACS, Max-Min AS, Rank-Based AS, ASGA, ...