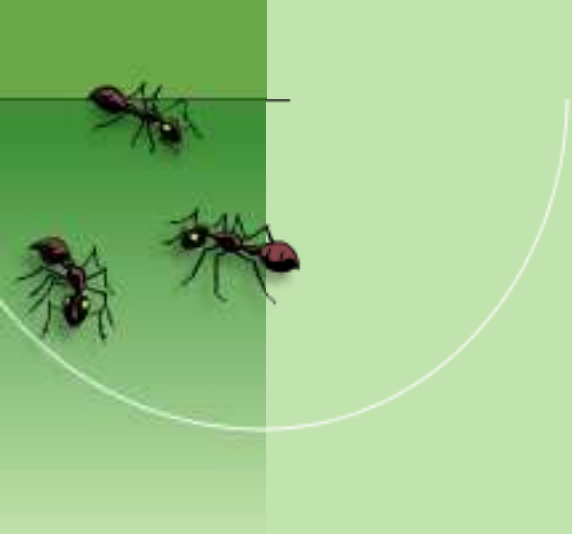


# Ant Colonies

Ameisen lösen das Travelling Salesman Problem

Jan Lümkemann
18.12.2003

# Überblick

- Biologisches Vorbild
- Modellierung künstlicher Ameisen
- Ant System
- Ant Colony System
- Konvergenzverhalten
- Zusammmenfassung

# Biologisches Vorbild

- Ameisenschwärme lösen komplexe Probleme
  - Futtersuche
  - Nestbau
  - Wegsuche zwischen Nestern

„Ameisenbrücke“
(Holldobler & Wilson 1990)

„Teamwork“
(Holldobler & Wilson 1990)

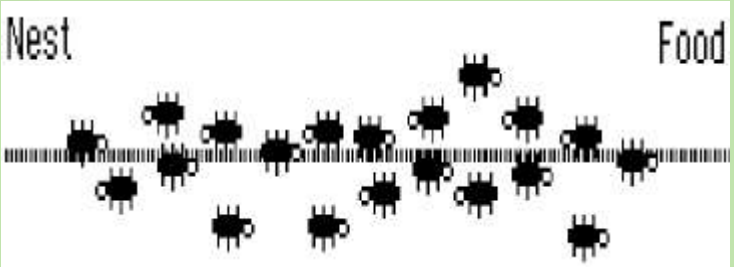
# Biologisches Vorbild

- Was steuert die Handlung ?
  - Eine einzelne Ameise kann nicht die gesamte Handlung steuern oder planen
  - Selbstorganisation
    - Entstehen von makroskopischen Mustern durch die Interaktion von mikroskopischen Teilchen (Physik, Chemie)
    - Positives Feedback
      - Belohnt „gute“ Strukturen
    - Negatives Feedback
      - Bestimmt durch Randbedingungen
    - Zufällige Fluktuation
      - Fördert die Erzeugung neuer Strukturen

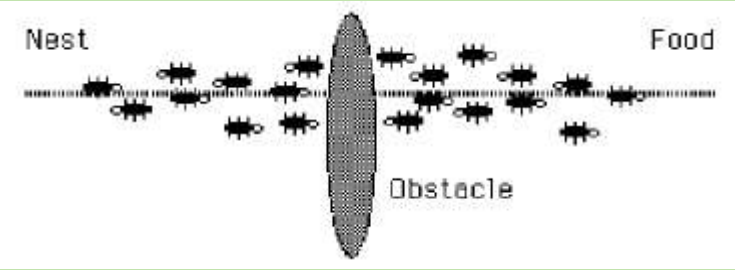
# Biologisches Vorbild

- Selbstorganisation bei Ameisen
  - Positives Feedback
    - Zurückgelegte Wege werden mit Pheromonen markiert
    - Wege mit hoher Pheromonkonzentration werden bevorzugt
  - Negatives Feedback
    - Pheromon zerfällt mit der Zeit
  - Zufällige Fluktuation
    - Verhalten der Ameisen lässt sich stochastisch beschreiben

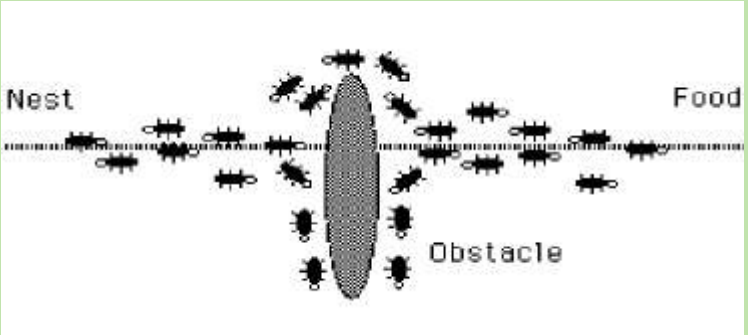
# Biologisches Vorbild



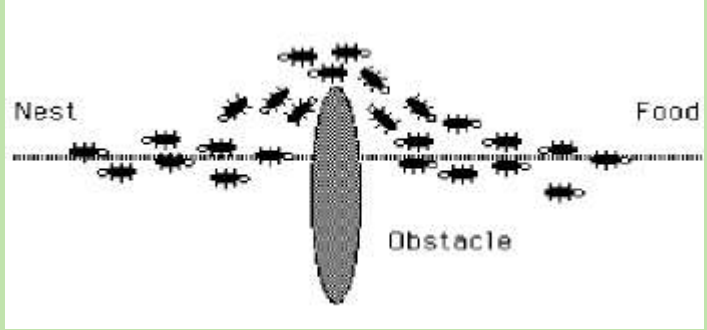

1) Ameisen wandern zwischen Nest und Futterquelle




2) Ein Hindernis versperrt den Weg

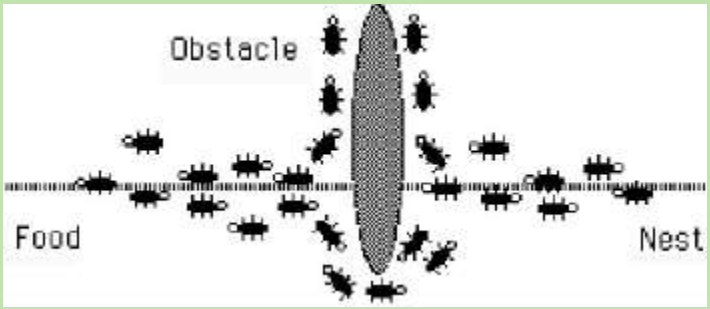



3) Ameisen wandern nach rechts und links und markieren die Wege mit Pheromonen

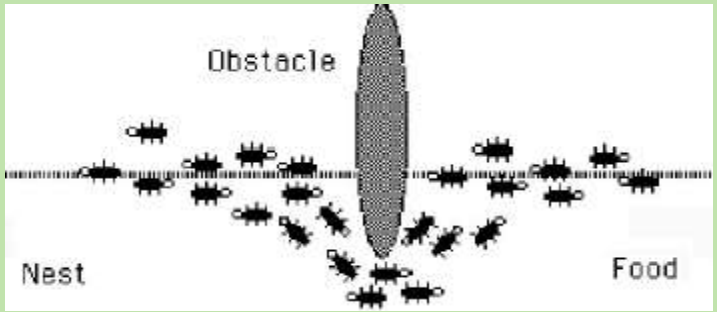



4) Ameisen finden den optimalen Weg

# Biologisches Vorbild




5) Das Hindernis verändert sich

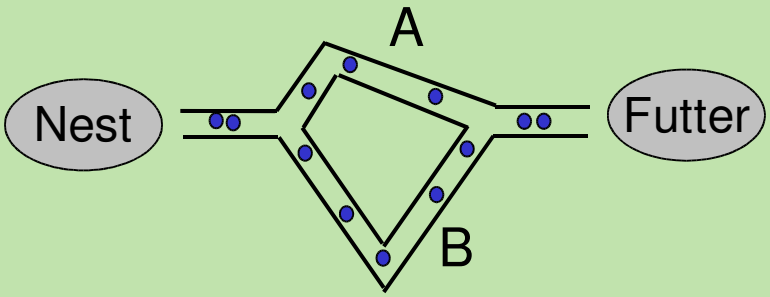



5) Ameisen finden schnell eine neue optimale Lösung

# Biologisches Vorbild

- Experimente von Deneubourg
- Stochastischer Zusammenhang:
  - Wahrscheinlichkeit, dass die (i+1)te Ameise Pfad A wählt:

$$P_A = \frac{(k + A_i)^n}{(k + A_i)^n + (k + B_i)^n} = 1 - P_B$$



# Modellierung künstlicher Ameisen

- Allgemeines Verfahren:
  - Beschreibung des Problems als Wegsuche auf Graphen
  - Modellierung der Selbstorganisation:
    - Statistische Funktion für die Auswahl des nächsten Knoten im Suchraum
    - Imitierung der Pheromonspuren als positives Feedback -> Belohnung guter Lösungen
    - Negative Feedbackmechanismen um Stagnation zu vermeiden (z.B. Pheromonzerfall)

# Modellierung künstlicher Ameisen

- Modellierung für das TSP:
  - Tabuliste
    - Enthält die bereits besuchten Städte
  - Lokale Heuristik
    - Kehrwert der Distanz $d_{ij}$ zweier Städte i,j: $\eta_{ij} = \frac{1}{d_{ij}}$
    - $\eta_{ij}$ heißt Sichtbarkeit
  - Globale Heuristik
    - Pheromonkonzentration der Kanten nach jeder Iteration aktualisiert
    - Pheromonkonzentration der Kante (ij) in Iteration t: $\tau_{ij}(t)$

# Ant System (AS)

- Jede Ameise erzeugt in jeder Iteration eine Lösung
- Die Wahrscheinlichkeit, dass Ameise k in Iteration i von Stadt i zu Stadt j wandert:

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \bullet [\eta_{ij}]^{\beta}}{\sum_{l \in J_i^k} [\tau_{il}(t)]^{\alpha} \bullet [\eta_{il}]^{\beta}}$$

*Statistische Fluktuation*

# Ant System

- Nach jeder Iteration aktualisiert jede Ameise k die Pheromonkonzentration auf den Kanten (ij) ihrer Lösung:

$$\Delta\tau_{ij}^{k}(t) = \frac{Q}{L^{k}(t)}$$

Q: Geschätzte Länge der optimalen Lösung
$L^{k}(t)$ : Länge der von Ameise k in Iteration t gefundenen Lösung

*Positives Feedback*

# Ant System

- Um Stagnation in lokalen Optima zu vermeiden zerfällt ein Teil des Pheromons nach jeder Iteration t:

$$\tau_{ij}(t) \leftarrow (1-\rho) \bullet \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

$\rho$ : Zerfallskoeffizient

$\Delta\tau_{ij}(t)$ : Summe aller Pheromonaktualisierungen auf Kante (ij) in Iteration t

*Negatives Feedback*

# Ant System

| | Beste Lösung | Mittlere Lösung | Standaradabweichung |
|---|---|---|---|
| **Ant System** | 420 | 420,4 | 1,3 |
| **Tabu Search** | 420 | 420,6 | 1,5 |
| **Simulated Annealing** | 422 | 459,8 | 25,1 |

(Vergleich für TSP Oliver30)

- Löst TSPs mit 30-70 Städten zuverlässig
- Zeitkomplexität: $O(t \bullet n^3)$
- Aber:
  - Suche nicht sehr zielgerichtet
  - Kovergiert für große TSPs sehr langsam
  - Lokale Optimierung erforderlich

# Ant Colony System (ACS)

- Verbesserung des Ant Systems durch Dorigo, um Konvergenzgeschwindigkeit zu erhöhen
- Lokale Optimierung leitet Suche schneller in Richtung der optimalen Lösung
- 4 wesentliche Änderungen

# Ant Colony System

1. Übergangsfunktion

- Eine Ameise k, die sich in Iteration t in Stadt i befindet, wählt als nächstes Stadt j gemäß:

$$j = \arg\max_{u \in J_i^k} \{ [\tau_{iu}(t)] \bullet [\eta_{iu}]^{\beta} \}, \text{wenn } q \leq q_0 \text{ und}$$

$$j = J, \text{wenn } q > q_0.$$

q: gleichverteilte Zufallsvariable auf dem Intervall [0;1]
$q_0$: einstellbarer Parameter
J: zufällig ausgewählte Stadt gemäß der Wahrscheinlichkeitsfkt. des AS

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)] \bullet [\eta_{ij}]^{\beta}}{\sum_{l \in J_i^k} [\tau_{il}(t)] \bullet [\eta_{il}]^{\beta}}$$

# Ant Colony System

2. Globale Aktualisierung der Pheromonspuren

- Nur die Ameise, die die bisher beste Lösung $T^+$ findet aktualisiert die Pheromonkonzentration auf den Kanten (ij) ihrer Lösung:

$$\tau_{ij}(t) \leftarrow (1-\rho) \bullet \tau_{ij}(t) + \rho \bullet \Delta\tau_{ij}(t), \text{wobei } \Delta\tau_{ij}(t) = \frac{1}{L^+}$$

$\rho$ : Zerfallskoeffizient

# Ant Colony System

3. Lokale Aktualisierung der Pheromonspuren

- Um Stagnation zu vermeiden verringert jede Ameise während einer Iteration die Phero-monkonzentrationen aller benutzten Kanten:

$$\tau_{ij}(t) \leftarrow (1-\rho)\bullet\tau_{ij}(t) + \rho\bullet\tau_0, \text{wobei}\,\tau_0 \approx \frac{1}{n\bullet L_{nn}}$$

n : Anzahl der Städte

$L_{nn}$ : Länge der optimalen Lösung nach der Nearest -Neighbor-Heuristik

# Ant Colony System

4. Kandidatenliste

- ACS speichert für jede Stadt die *c* nächsten Nachbarn in einer Kandidaten-Liste
- Ameisen wählen eine Stadt aus der Kandidaten-liste gemäß den obigen Formeln
- Wurden bereits alle Städte der Kandidatenliste besucht, wählt die Ameise als nächstes die nächste noch nicht besuchte Stadt

# Ant Colony System

| | ACS | ACS | GA | GA | EP | EP | SA | SA |
|---|---|---|---|---|---|---|---|---|
| | best | # iter. | best | # iter. | best | # iter. | best | # iter. |
| **Eil50** | **425** | **1830** | 428 | 25000 | 426 | 100000 | 443 | 68512 |
| **Eil75** | **535** | **3480** | 545 | 80000 | 542 | 325000 | 480 | 173250 |
| **KroA100** | **21282** | **4820** | 21761 | 103000 | N/A | N/A | N/A | N/A |

- GA: Genetische Algorithmen
- EP: Evolutionäres Programmieren
- SA: Simulated Annealing

- ACS findet bei dieser Problemgröße die besten Lösungen
- Bei noch größeren Problemen (200 – 1000 Städte) finden Genetische Algorithmen bessere Lösungen

# Konvergenzverhalten

Stützle und Dorigo haben gezeigt:

- Wenn die Pheromonkonzentration nach unten begrenzt ist und nur die Ameise, die die beste Lösung findet, die Pheromonkonzentration aktualisiert, gilt:

$$\forall \varepsilon > 0 \; \exists t : P^*(t) \geq 1 - \varepsilon,$$

$P^*$: Wahrscheinlichkeit, dass mindestens eine optimale Lösung gefunden wird

- D.h.: Im Grenzfall (für unendlich viele Iterationen) konvergieren solche Ameisenalgorithmen gegen eine optimale Lösung

# Zusammenfassung

- Prinzip der Selbstorganisation:
  - Positives Feedback
  - Negatives Feedback
  - Statistische Fluktuation
- Verfahren konvergiert gegen die optimale Lösung
- Tolerant gegenüber dynamischen Verhältnissen