Algorithmen

Ameisenalgorithmus: Idee und Anwendungen in der Tourenplanung

Prof. Dr. Julia Rieck, Michael Ganske

Abteilung Betriebswirtschaft und Unternehmensforschung
Abteilung Software Systems Engineering

Stichworte: Suchen, Optimieren, exakte und Näherungsverfahren

Hildesheim, 21.03.2022, it4school

Outline

- Algorithmen
- Finden eines kürzesten Weges mit einem Algorithmus
- Natürliche Ameisen und Ameisenalgorithmen
- Anwendungen des Ameisenalgorithmus:
  - zum Finden eines kürzesten Weges
  - in der Tourenplanung

Was sind Algorithmen?

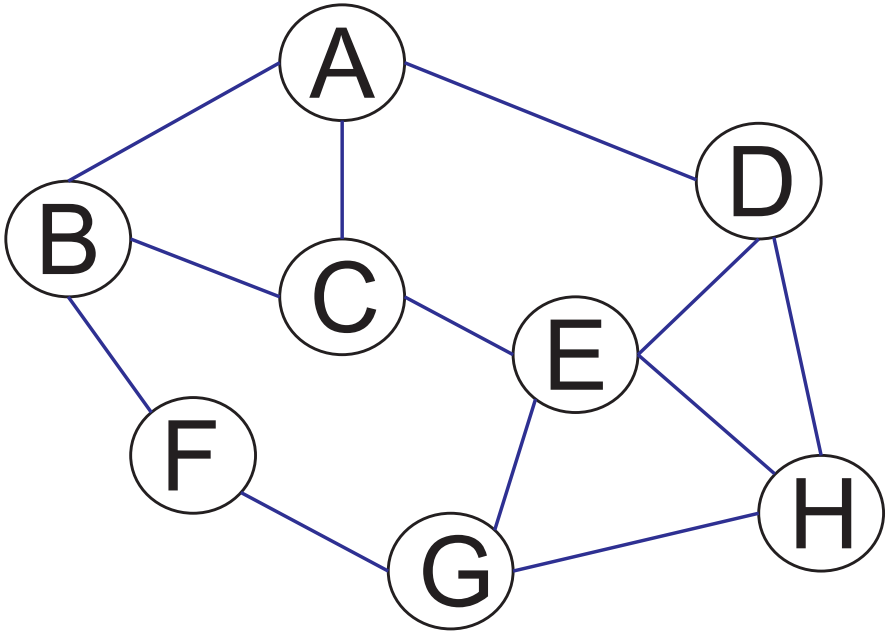
- Schritt für Schritt Anleitungen zur Lösung von Problemen
- Je mehr Schritte, desto höher die Laufzeit eines Algorithmus
- Schritte müssen geschickt nacheinander ausgeführt werden

- Algorithmen sind überall versteckt, z.B.:
  - Staubsaugerroboter
  - Smartphone
  - Waschmaschine
  - Navigationsgerät
  - u.v.m.

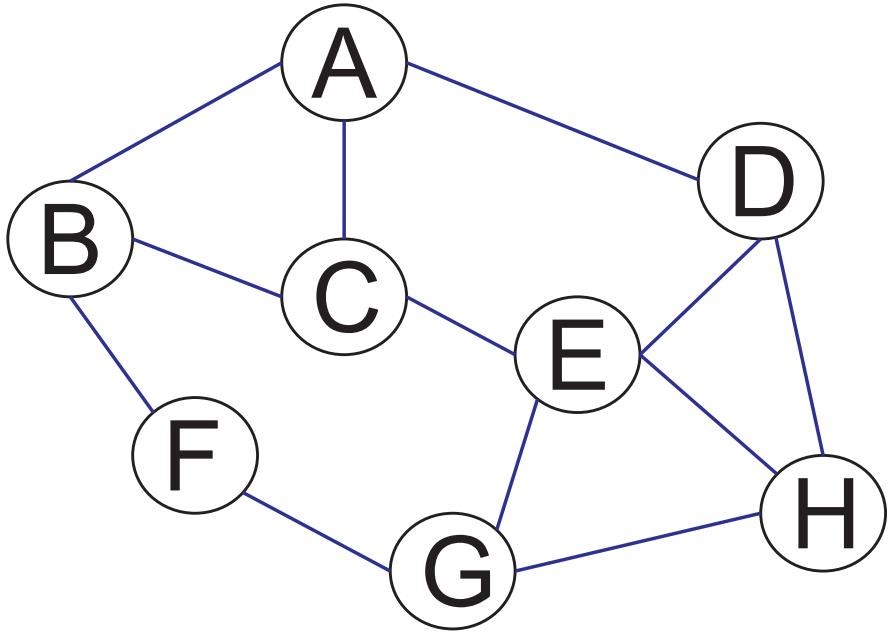
Baukasten zum Designen von Algorithmen

- Ein- und Ausgabe-Anweisungen
- Variablen
- Verzweigungen (if-Anweisung, if-else-Anweisung)
- Schleifen (for-Schleife, while-Schleife)
- Zeichen und Zeichenketten (Strings)
- Listen, Arrays
- Subroutinen (Prozeduren, Funktionen)

Beispiel: Finden eines kurzen Weges von A zu H



Beispiel: Finden eines kurzen Weges von A zu H



Möglicher Algorithmus fragt am aktuellen Knoten:

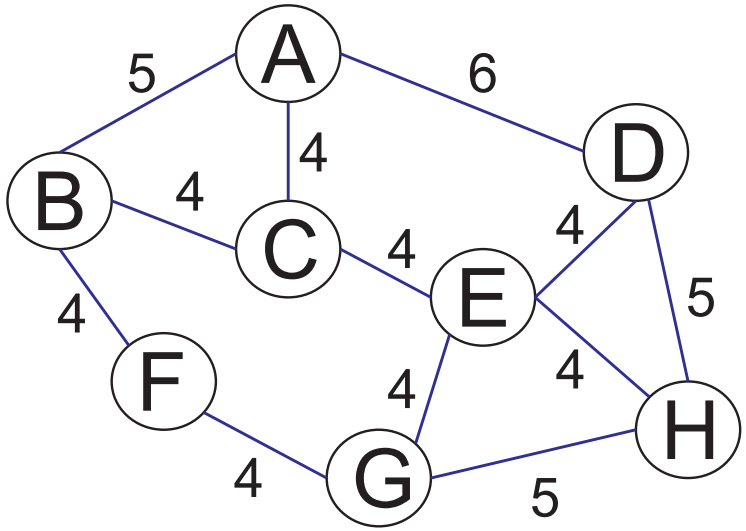
- Welche Straßen führen vom aktuellen Knoten zu anderen Knoten?
- Welche der Straßen hat die kürzeste Distanz zum nächsten Knoten?
- Wahl des Weges mit kürzester Distanz

Beispiel: Finden eines kurzen Weges von A zu H

- Ergebnis A–C–E–D–H ist nicht optimal

- Änderung des Algorithmus
- Systematisches Speichern des kürzesten Weges von A zu aktuellem Knoten
- Betrachtung einer Menge Q von noch zu untersuchenden Knoten

Beispiel: Finden eines kürzesten Weges von A zu H



- Start bei Knoten A
- $d_{\mathsf{AB}} = 5$, $d_{\mathsf{AC}} = 4$, $d_{\mathsf{AD}} = 6$, $Q = \{\mathsf{B}, \mathsf{C}, \mathsf{D}\}$
- Entnehme B aus Q und fahre bei Knoten B fort
- d_{AC} (über B) $= 9$, d_{AF} (über B) $= 9$, $Q = \{\mathsf{C}, \mathsf{D}, \mathsf{F}\}$
- Entnehme C aus Q und fahre bei Knoten C fort
- d_{AE} (über C) $= 8$, $Q = \{\mathsf{D}, \mathsf{F}, \mathsf{E}\}$
- Entnehme D aus Q und fahre bei Knoten D fort
- d_{AH} (über D) $= 11 \ldots$
- Keine weiteren Updates mehr bis $Q = \emptyset$

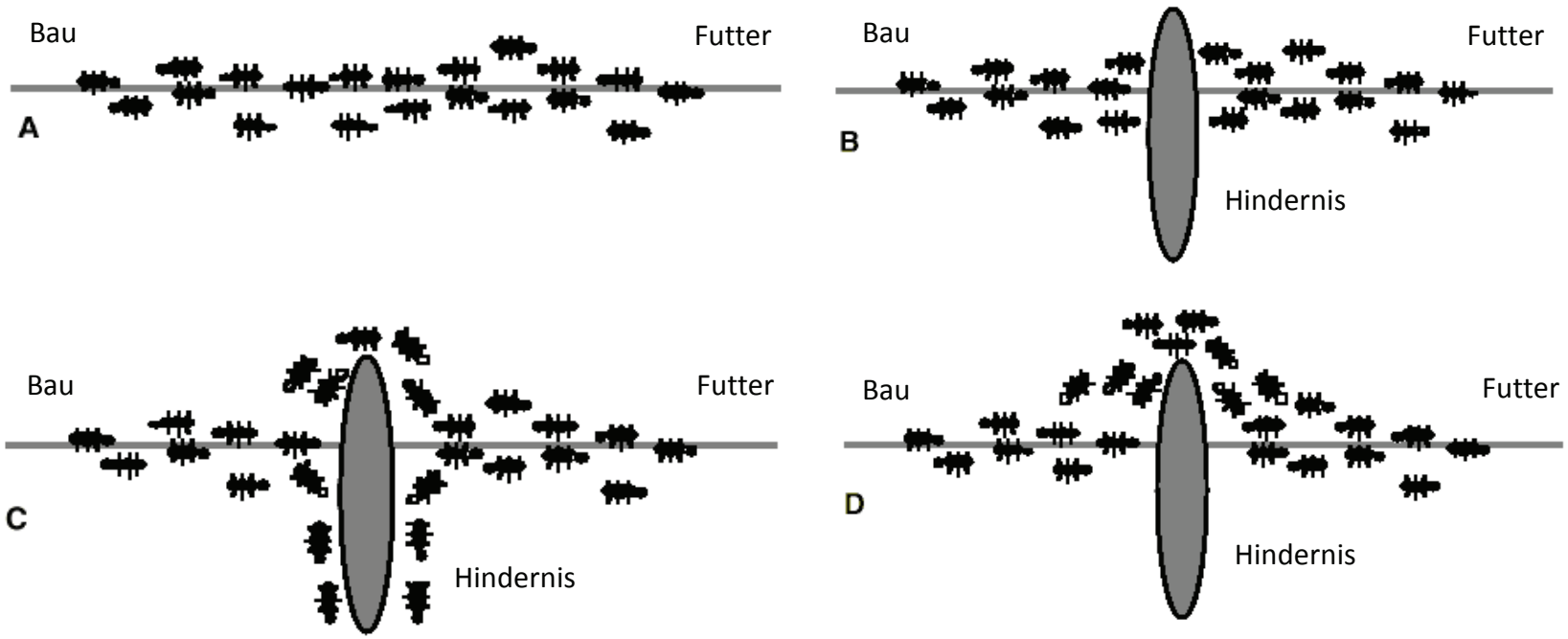
Kreativität

- Um Algorithmen zu designen kann man kreativ werden
- Man sollte auf die Anzahl der erforderlichen Schritte, d.h. auf die Laufzeit, achten
- Das zugrunde liegende Problem sollte bestmöglich gelöst werden
- Man kann sich z.B. die Natur zum Vorbild nehmen
- Ameisenalgorithmen imitieren natürliche Ameisen

Natürliche Ameisen

- Ameisen können sich in der Gruppe so gut organisieren, dass sie komplexe Probleme wie z. B. das Suchen nach Futter schnell lösen können
- Bei der Futtersuche gibt es keine Ameise, die einer anderen Befehle erteilt, sondern jede Ameise entscheidet selbst
- Die Kommunikation läuft über eine chemische Substanz, so genannte Pheromone, ab
- Jede Ameise markiert ihren Weg mit Pheromonen
- Andere Ameisen erkennen die Pheromonspur und wählen die Wege mit einer Wahrscheinlichkeit proportional zu der darauf befindlichen Pheromonmenge
- Pheromone können nicht nur verstärkt werden, sie verdunsten auch mit der Zeit
- Dieser Prozess führt dazu, dass die Ameisen den kürzesten Weg zum Futter finden

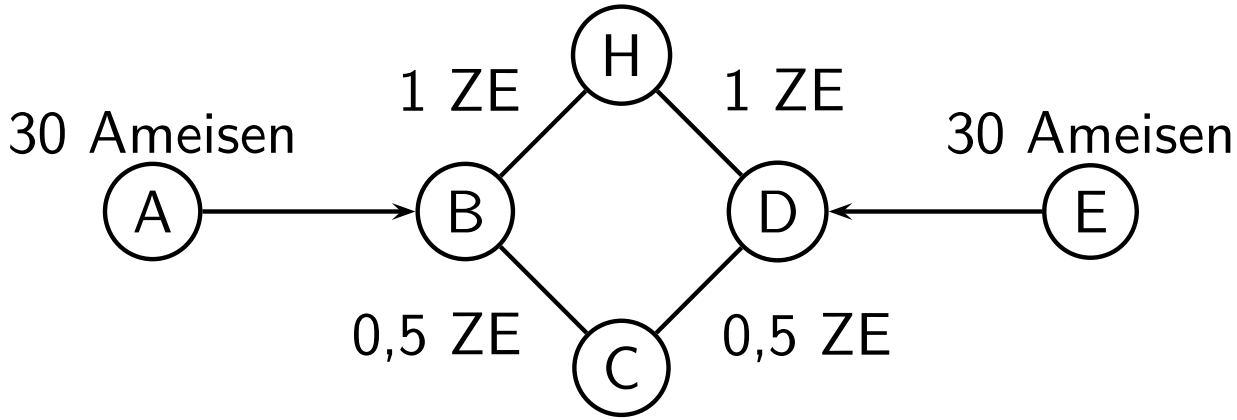
Natürliche Ameisen II



Quelle: M. Dorigo und L. M. Gambardelle (1997) Ant Colony System: A Cooperative Learning Approach to the Taveling Salesman Problem. In: IEEE Transactions 1.1

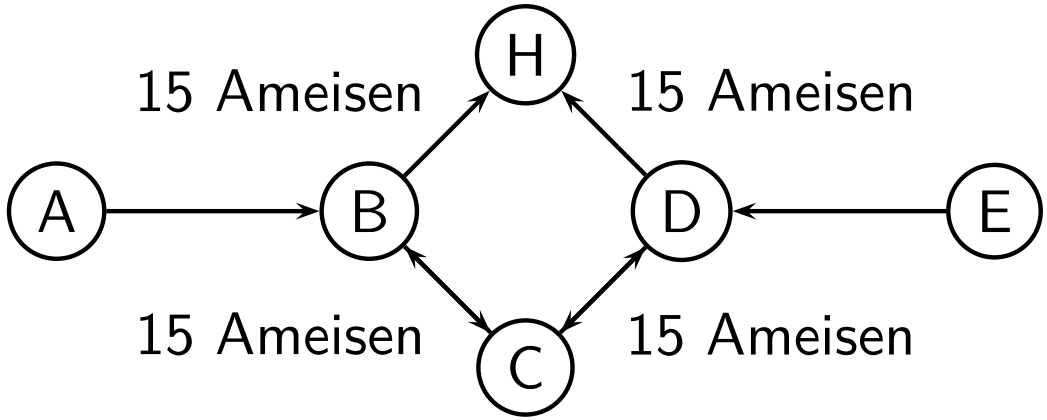
Fallbeispiel: Ausgangssituation

- Eine Ameise läuft mit der Geschwindigkeit von 1 cm pro ZE
- Während sie läuft hinterlässt sie zum Zeitpunkt t eine Pheromonspur der Stärke 1
- Nehmen wir an, dass pro ZE 30 Ameisen von Ort E kommend in D ankommen und 30 Ameisen erreichen von Ort A kommend den Punkt B



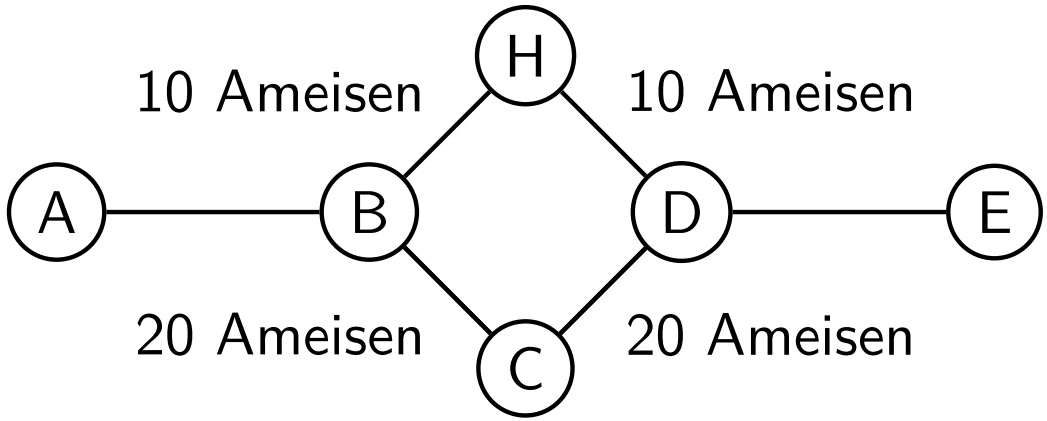
Fallbeispiel: $t = 0$

- Zum Zeitpunkt t=0 sind jeweils 30 Ameisen in B und D
- Dort finden sie keine Pheromonspur vor und entscheiden sich zufällig für einen der beiden Wege vor sich
- Im Durchschnitt werden sowohl von B als auch von D aus jeweils 15 Ameisen nach H und 15 Ameisen nach C laufen



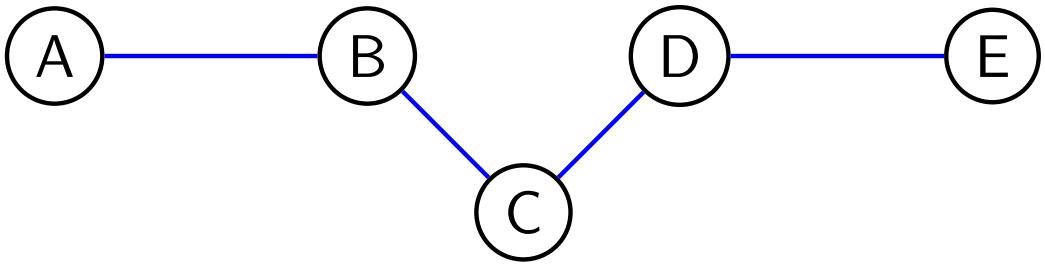
Fallbeispiel: $t = 1$

- Zur Zeit t=1 sind 30 neue Ameisen in Punkt B
- Auf dem Weg von B nach H finden sie eine Pheromonspur der Stärke 15 und auf dem Weg B-C finden sie eine Spur der Stärke 30 (15 Ameisen, die in t=0 von B über C nach D gelaufen sind und 15 Ameisen, die von D über C nach B gelaufen sind)
- Somit werden doppelt so viele Ameisen in Richtung C laufen (20) als Ameisen, die in Richtung H (10) laufen
- Dasselbe gilt für die 30 neuen Ameisen, die in Punkt D ankommen



Fallbeispiel: kürzester Weg

- Nach einiger Zeit werden schließlich alle Ameisen auf dem kürzeren Weg laufen
- Der kürzeste Weg ist A–B–C–D–E



Ameisenalgorithmen . . .

- orientieren sich (abstrakt) an der Nahrungssuche von Ameisen
- simulieren die dynamische Verhaltensoptimierung mit Spuren als Informationsträger

- Im Algorithmus bestimmen drei Kriterien die Laufrichtung einer Ameise auf der Suche nach Nahrung:
  (1) bereits besuchte Orte
  (2) Entfernung zu den noch nicht besuchten Orten
  (3) Intensität der Pheromon-Spuren auf den jeweiligen Wegen

- Je stärker die Pheromonspur und je kürzer der Weg, desto eher wird eine Ameise den Weg einschlagen

Schema eines Ameisenalgorithmus

Algorithmus 1:

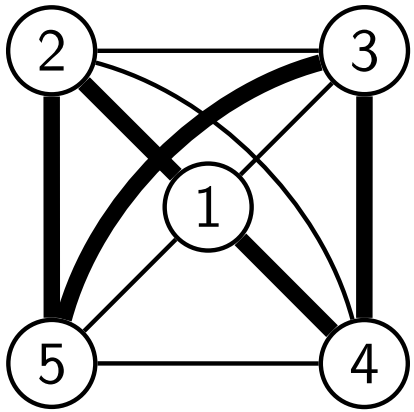
Initialisiere die Pheromonwerte

1: **repeat**
2: **for** Ameise $k \in \{1, \ldots, m\}$ **do**
3: Konstruiere eine Lösung
4: **for** alle Pheromonwerte **do**
5: Setze die Werte um einen bestimmten Prozentsatz herab
6: (Verdunstung)
7: **for** Pheromonwerte, die Bestandteil guter Lösungen sind **do**
8: Erhöhe den Wert (Intensivierung)
9: **until** das Stoppkriterium erreicht ist

Fallstudie zu Ameisenalgorithmen

- Wir betrachten das so genannte Traveling Salesman Problem (Problem des Handlungsreisenden)
- Problemstellung aus der Tourenplanung

- Gegeben seien n Kunden sowie symmetrische Distanzen $d_{ij} = d_{ji}$ zwischen allen Kunden i und j Es ist eine Rundreise mit minimaler Länge zu finden, auf der jeder Kunde einmal angefahren wird
- Mögliche Rundreise:



Ameisenalgorithmen: Anwendung auf das TSP

Initialisierung: Verteilung der Ameisen auf die Städte. Jede Stadt erhält z.B. eine Ameise.

Hauptschritt: Die Ameisen bewegen sich unabhängig voneinander von einer Stadt i zur nächsten Stadt j. Die Auswahl der als nächstes zu besuchenden Stadt j erfolgt stochastisch anhand der drei Kriterien:

(a) Welche Orte bereits besucht wurden Stadt j darf nicht bereits in der Tabuliste der betreffenden Ameise enthalten sein.
(b) „Sichtbarkeit“ (d_{ij}, Distanz zwischen i und j)
(c) „Stärke der Pheromonspur“

Ameisenalgorithmen: Anwendung auf das TSP II

- Nachdem alle Ameisen ihre Tour beendet haben, wird die Gesamtlänge jeder Tour berechnet
- Auf jede Kante einer Tour wird nun Pheromon gestreut; die Menge des Pheromons ist proportional zur Güte der Tour (kurze Touren $\rightarrow$ viel Pheromon)
- Alle Ameisen werden auf ihre Ausgangsstädte zurückgesetzt
- Das Verfahren beginnt erneut, bis ein Stoppkriterium (bspw. die maximale Anzahl von Iterationen) erreicht wird