CARL VON OSSIETZKY universität OLDENBURG

Ausarbeitung für das Modul "Einführung in die Mathematische Modellierung II"

Entwicklung von Ameisenstraßen mit virtuellen Pheromonen

Nina Marwede
12. Semester Diplom-Informatik

28. September 2007

Abstract

Ameisen kommunizieren unter anderem durch das Verteilen von Pheromonen in ihrer Umgebung zur Markierung von Wegen und Futterquellen. Ziel dieser Arbeit ist nicht die biologisch korrekte Simulation des Verhaltens der Ameisen, sondern eine anschauliche Reproduktion und Experimente mit der Auswirkung dieses Verhaltens: die Entstehung von Ameisenstraßen. Außerdem werden Möglichkeiten der technischen Nutzbarkeit dieser Strukturbildung betrachtet und eine Heuristik für das Problem des Handlungsreisenden implementiert.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

"Wer den Weg einer Ameise beobachtet, kann dem Insekt möglicherweise eine mehr oder weniger komplizierte Planungsfähigkeit zuschreiben. Aus der Perspektive der Ameise ergibt sich der Weg jedoch mehr aus den vor ihr liegenden Hindernissen plus einigen simplen Regeln." [Sti03]

1 Einleitung und Begriffsklärung

Ameisen zählen zu den erfolgreichsten Geschöpfen der Evolution: in 100 Millionen Jahren haben sich über 12.000 Arten mit über 10 Billionen Individuen entwickelt, die in einigen Biotopen 10% der lebenden Biomasse ausmachen. Es haben sich scheinbar komplexe Strukturen der Koordination und Kommunikation innerhalb eines Staates entwickelt, doch nicht immer muss der zugrunde liegende Mechanismus ebenfalls komplex sein.

Die grundlegende Orientierung und Nahrungssuche der Ameisen funktioniert mithilfe von Pheromonen. Pheromone sind Duftstoffe, die der biochemischen Kommunikation zwischen Lebewesen einer Spezies dienen. Andere Arten bleiben von dieser Kommunikation ausgeschlossen. Neben Sexuallockstoffen gibt es Pheromone zur Wegmarkierung oder als Alarmbotenstoff. [Wiki02]

Tatsächlich gibt es sogar Arten, die sich über große Strecken in der Wüste weitgehend *ohne* Pheromone orientieren, sondern über visuelle und motorische Reize: die Wüstenameisen Cataglyhpis finden den Heimweg mithilfe der Sonne und merken sich dafür die Anzahl ihrer Schritte. [WWW06]

In dieser Arbeit werden lediglich Pheromone zur Wegmarkierung betrachtet: Weder andere Pheromone noch andere Arten der inter-insektoiden Kommunikation sollen hier eine Rolle spielen.

Durch Verkettung von Wegmarkierungen können Straßen entstehen – eine "Straße" bedeutet in diesem Fall keinen speziellen Ausbau oder eine Befestigung im Voraus, sondern eine Entstehung aus der Benutzung heraus. Es handelt sich um ein statistisches Phänomen: Durch häufige Benutzung bilden sich (virtuelle) Trampelpfade. Im Fall der Ameisen werden diese Pfade natürlich nicht "ausgetreten" oder auf andere Weise optisch in Mitleidenschaft gezogen, sondern lediglich olfaktorisch markiert.

1.1 Zellulare Automaten

Zellulare Automaten bieten eine einfache Möglichkeit zur Modellierung räumlicher Prozesse. Ihr Prinzip wurde bereits Ende der 1940er Jahre durch John von Neumann beschrieben. Eine ihrer faszinierendsten Eigenheiten ist, dass einfache, lokal begrenzte Zustandsübergangsregeln komplexe globale Strukturen hervorbringen können. Ein berühmtes Anwendungsbeispiel ist Conway's "Game of Life". [Lov02]

Ein Zellularer Automat besteht aus einer **Menge von Zellen** in regelmäßiger **Anordnung**, meist in zwei räumlichen **Dimensionen**. Gegenüber liegende Ränder der so geformten Fläche können miteinander verbunden sein, um etwa eine Zylinder- oder Torus-förmige Konfiguration zu erzielen.
Jede einzelne Zelle ist ein aus der Theoretischen Informatik bekannter **endlicher Automat** und repräsentiert entweder einen Punkt oder eine Fläche im Raum. Die Zellen sind durch eine symmetrische **Nachbarschaftsrelation** miteinander verbunden, typisch ist die "Moore-Nachbarschaft" – auch "8-Nachbarschaft" genannt, weil bei quadratischer

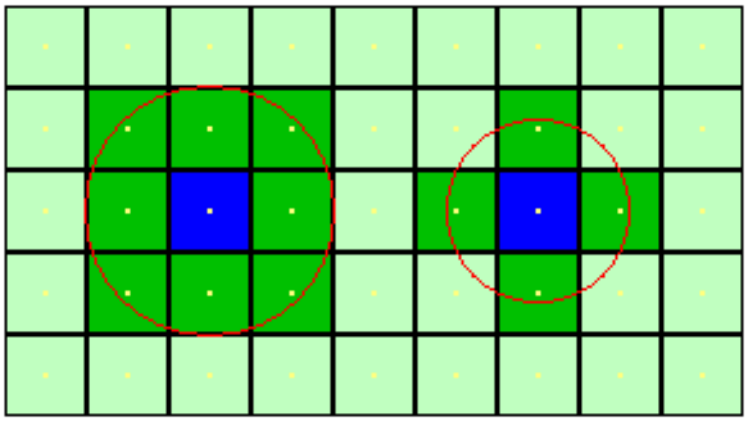

Abbildung 1: Nachbarschaftsbeziehungen: Moore bzw. Von-Neumann

Struktur genau die acht umliegenden Zellen als Nachbarn gelten, die mindestens eine Ecke mit der zentralen Zelle gemeinsam haben, wohingegen bei der Von-Neumann-Nachbarschaft nur jene vier Zellen als Nachbarn gelten, die eine Kante gemeinsam haben. Der Grad des Einflusses der Nachbarzellen ist meist binär (d.h. vorhanden oder nicht) oder mit zunehmender Entfernung mit einer Funktion (z.B. linear oder quadratisch) abnehmend. Die Nachbarschaftsregion muss allerdings nicht zwingend aus den direkt angrenzenden Zellen bestehen.

Der **Zustand** jeder Zelle ist durch ihre Attribute bestimmt, es kann sich im Prinzip um beliebige Größen wie Ausmaße, räumliche Orientierung, Energiegehalt, Wachstumsrate oder Farbe handeln. Die **Zustandsübergänge** finden in regelmäßigen Zeitschritten für alle Zellen synchron statt, dabei hängt die Entwicklung einzelner Zellen sowohl vom eigenen Zustand ab als auch von den Zuständen jener Zellen in der vorgegebenen Nachbarschaft, und zwar jeweils vom vorhergehenden Zeitschritt.

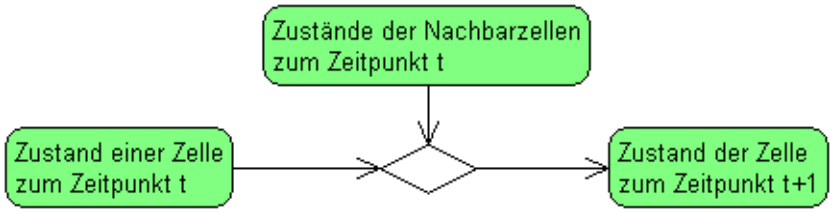

Abbildung 2: Zustandsabhängigkeiten beim Zellularen Automaten

Alle Eigenschaften des Automaten werden zur Modellierungszeit definiert und sind während der Laufzeit konstant, lediglich die Werte der Attribute können sich ändern, müssen dies aber nicht. Ergebnis einer Simulation kann sowohl allein der Endzustand abhängig vom Ausgangszustand nach einer festgelegten Laufzeit als auch die genaue Abfolge der Veränderungen sein.

Ihre **Grenzen** finden die klassischen Zellularen Automaten, wenn es darum geht, nicht mehr nur einfache lokale Prinzipien, sondern reale ökologische Systeme zu simulieren.

Da sich solche Modelle oft nur mit Mühe in Zellulare Automaten überführen lassen, wurden im Laufe der Zeit verschiedene Erweiterungen entwickelt. So bieten beispielsweise die "Hierarchischen Asymmetrischen Zellularen Automaten" (HAZA) [SV01] folgende Möglichkeiten:

- irreguläre Topologien
- asymmetrische Nachbarschaftsrelationen
- Prozesse auf verschiedenen räumlichen und zeitlichen Skalen
- globale Einflüsse

Auch wenn das vorliegende Modell auf den ersten Blick große Ähnlichkeit mit klassischen Zellularen Automaten aufweist, so handelt es sich im Endeffekt doch eher um einen Agenten-basierten Mechanismus auf zellularer Basis.

1.2 Software-Agenten

Ein Software-Agent ist ein Computerprogramm, das seine Aufgaben mit einer eigenen Initiative weitgehend unabhängig von der Steuerung durch einen Anwender bearbeitet und dabei auf die Änderung der Umgebung reagiert. Typischerweise kann auch Kommunikation mit anderen Agenten stattfinden. Der Agent lernt aufgrund zuvor getätigter Entscheidungen oder Beobachtungen. Manche Agenten können sich innerhalb ihrer Umwelt bewegen, andere zeichnen sich durch größeres Wissen oder verbesserte Lernfähigkeit und die Möglichkeit zu Verhaltensänderungen aus. [Wiki04]

In der "freien Wildbahn" werden Software-Agenten auch als "Bots" (Kurzform von Robots) bezeichnet und sind in vielen Formen anzutreffen, etwa als Webcrawler von Suchmaschinen, als Akteure in simulierten oder realen Spielen, als Dienstleister in Chat-Systemen, als Überwacher von Clustern, als Korrektoren in Wikis oder als Angreifer in kompromittierten virtuellen Netzen.

Im Gegensatz zu Expertensystemen, die ebenfalls Elemente der künstlichen Intelligenz enthalten, sind Software-Agenten meist nur scheinbar intelligent, sollen intelligent wirkendes Verhalten also lediglich vortäuschen. Im Falle von humanoiden Figuren in Unterhaltungsprogrammen kann ein "natürliches" Bewegungsmuster beispielsweise mithilfe einer Zufallskomponente umgesetzt werden. Gegnern und Konkurrenten des menschlichen Teilnehmers (etwa bei Autorennspielen) wird sogar absichtlich ein Teil der ihnen zugänglichen Informationen entzogen, um dem Menschen überhaupt eine Chance zu lassen, aber auch zur Abgrenzung untereinander. Häufig hilft dem Entwickler jedoch bereits der ohnehin eingeschränkte Lebensraum seiner Wesen.

Nicht zuletzt ist die Schonung der begrenzten Ressourcen insbesondere bei grafisch aufwändigen Spielen ein wichtiges Kriterium bei der Entwicklung einfacher Algorithmen zur Erzeugung scheinbar komplexen Verhaltens – "die goldene Regel der Unterhaltung lautet: Es muss nicht realistisch sein, sondern überzeugend." [Gle03]

2 Modellierung

Durch Verbreitung eines Pheromons in der Nähe eines Ameisenhügels sollen Ameisen ihre Heimat leichter wiederfinden können. Durch Verbreitung eines zweiten Pheromons sollen auf die gleiche Weise Nahrungsquellen markiert werden. Es wird erwartet, dass die Ameisen nach einer Orientierungsphase eine oder mehrere "Straßen" zwischen Ameisenhügel und Nahrungsquelle(n) aufbauen werden, d.h. eine relativ kurze Verbindung, auf der beide Pheromon-Varianten in relativ hoher Konzentration vorkommen. Der Mechanismus dabei sollte eine Rückkopplung sein: je mehr Ameisen den "besten" (direktesten) Weg gehen, desto deutlicher ist die Markierung, und je deutlicher die Markierung, desto mehr Ameisen werden ihr folgen.

Dieses Kapitel stellt die Entwicklung einer Experimentier-Umgebung für einen Ameisen-Algorithmus in *Matlab* dar.

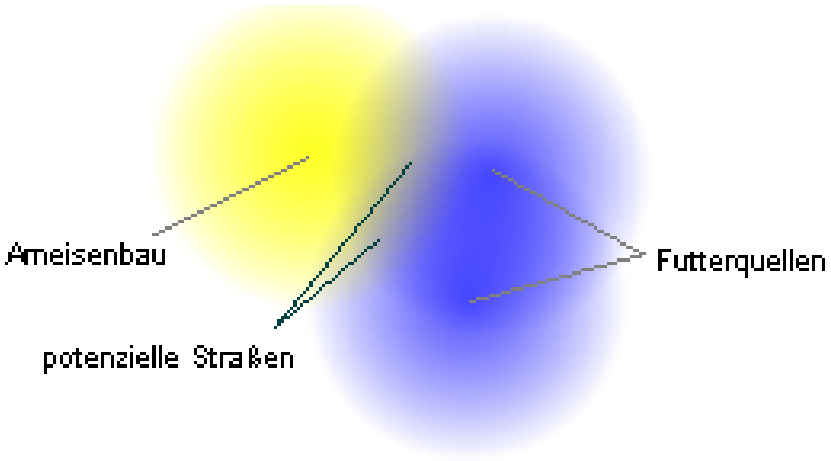

Abbildung 3: Ungefähr zu erwartende Verteilung der Pheromone

2.1 Modellbeschreibung

Jede Ameise im Modell ist ein Software-Agent, der in einer virtuellen Umwelt agiert. Er verfügt weder über eigenes Wissen noch über ein Gedächtnis oder Planungsvermögen, sondern erhält begrenzte Informationen aus seiner Umgebung und wählt seine Handlungen aufgrund einfacher Bedingungs-Aktions-Regeln.

Die Umwelt wird durch einen klassischen zweidimensionalen abgeschlossenen Zellularen Automaten gebildet. In jedem Zeitschritt bewegen sich alle Ameisen jeweils genau einen Schritt in eines der acht benachbarten Felder (Moore-Nachbarschaft).

Die wesentliche Aufgabenstellung ist die Wegfindung, d.h. die Berechnung für jede einzelne Ameise, für welches der ihr jeweils möglichen Zielfelder sie sich entscheidet. Sie agiert dabei für sich allein, unabhängig vom Zustand oder Verhalten anderer Ameisen, beeinflusst nur indirekt durch die Orientierung an den von allen Ameisen im Revier verbreiteten Pheromonen.

2.2 Anforderungsdefinition

Die Ameisen sollen zwei getrennte Vorräte mit Pheromonen (Heimat-anzeigend und Futter-anzeigend) mit sich führen können, die sie in der Umwelt verteilen. Sie bewegen sich mit einer Zufallskomponente (Random Walk) und orientieren sich dabei an einem der beiden Pheromone. An bestimmten Positionen (Ameisenhügel, Futterquellen) können sie ihre Pheromon-Vorräte auffüllen. Sterben können die Ameisen im Grundmodell nicht, noch können sie schlafen oder sich fortpflanzen.

Von der Umwelt sollen der Ameisenhügel sowie eine oder mehrere Futterquellen verwaltet werden. Die verteilten Pheromon-Markierungen sollen langsam verwittern, damit das Gesamtsystem schlechte Pfade vergessen und im Austausch neue erlernen kann.

Objekte	**Datentypen**	**Wertebereiche**
Größe der Umwelt	Ganzzahlen	$N \in [1\ \infty]$
Koordinaten für Bau und Nahrungsquellen	Ganzzahlen	$[1\ N]$
Heimat/Nahrung anzeigendes Pheromon	Felder: Fließkommazahlen	$[1\ N]\ [1\ N]$ $[0.0\ \infty]$
Ameise 2 Koordinaten Bewegungsrichtung Heimat/Nahrung anzeigende Pheromone Bewegungszustand	Struktur: Ganzzahlen Ganzzahl Fließkommazahlen Ganzzahl	 $[1\ N]$ $[1\ 8]$ $[0.0\ 1.0]$ $\{0, 1, 2\}$
Futterstelle 2 Koordinaten Anzahl Futterpakete	Struktur: Ganzzahlen Ganzzahl	 $[1\ N]$ $[0\ \infty]$

Tabelle 1: Datenstrukturen zur Modellierung der Ameisen

Wichtigste Nebenanforderung ist die anschauliche grafische Darstellung in unterscheidbaren Farbschattierungen für Ameisen, Ameisenhügel und Futterquellen sowie Pheromone. Zudem sollte die Geschwindigkeit der Simulation Berücksichtigung finden.

Objekte	**Datentypen**	**Wertebereiche**
Ausgabematrix für die Anzeige der Umwelt	Feld: Fließkommazahlen	$[1\ N]\ [1\ N]$ $[0.0\ \infty]$
Farbschema für die Falschfarben-Kodierung	Tabelle: Fließkommazahlen	$\{rot, gruen, blau\}$ $[0.0\ 1.0]$

Tabelle 2: Datenstrukturen zur Darstellung der Umwelt

2.3 Entwurfsentscheidungen

Randbetrachtung Die Ameisen bewegen sich in einem kleinen, räumlich abgeschlossenen Gebiet. Die Ränder werden nicht miteinander verbunden. Damit sie ihre Umwelt nicht verlassen, wird eine Grenze implementiert, so als ob die Ameisen dort nicht hin wollten.

Futterpakete statt Fließkommazahlen Eine exaktere Aufteilung des Futters ist im vorliegenden Modell nicht erforderlich, da weder dessen Struktur noch die persönliche Leistungsfähigkeit oder Appetit der Ameisen betrachtet wird.

Speicherung der Bewegungsrichtung ... ist nicht zwingend erforderlich, erscheint jedoch sinnvoll zur realistischeren Gestaltung insofern, dass die Ameisen bevorzugt Orte aufsuchen sollen, an denen sie noch nicht waren. Aus dem gleichen Grund wird eine Bevorzugung der Geradeausbewegung implementiert. Dies unterstützt außerdem die Einschränkung von festgefahrenen Hin-und-Her-Bewegungen sowie kurzen Schleifen.

Speicherung des Zustandes Die Information, ob ein Nahrungspaket geladen ist, kann implizit aus der Größe des Nahrung anzeigenden Pheromons abgelesen werden, indem dieses nur auf Null gesetzt wird, wenn das Nahrungspaket im Ameisenhügel abgeliefert wird. Da jedoch Rundungsfehler bei langer Laufzeit der Simulation nicht ausgeschlossen werden können und zusätzlich die Übersichtlichkeit im Quelltext leidet, wird eine separate Variable zur Speicherung des Bewegungszustandes jeder Ameise verwendet.

Zusätzliche Orientierung am jeweils anderen Pheromon? ... ergibt wenig Sinn, denn bei positiver Orientierung würde die Suche in die falsche Richtung gehen, und bei negativer Orientierung würden eventuell vorhandene “Straßen” (Orte, an denen zuvor relativ viele Ameisen unterwegs waren) verfehlt.

2.3.1 Zustände und Verhalten einer Ameise

Der Gesamtzustand einer Ameise definiert sich durch wenige Variablen:

1. Heimat anzeigendes Pheromon – Konzentration als Fließkommazahl
2. Nahrungsquelle anzeigendes Pheromon – Konzentration als Fließkommazahl
3. Suchzustand – {0,1,2} – Orientierung an einem Pheromon, Futterpaket geladen

Da die exakte Konzentration der Pheromone nur eine untergeordnete Rolle spielt, sondern ihre An- bzw. Abwesenheit über die Aktion entscheidet, ergeben sich durch Reduzierung der Konzentrationen auf *Null* bzw. *größer Null* acht theoretisch mögliche Zustände, wie in Abbildung 4 dargestellt.

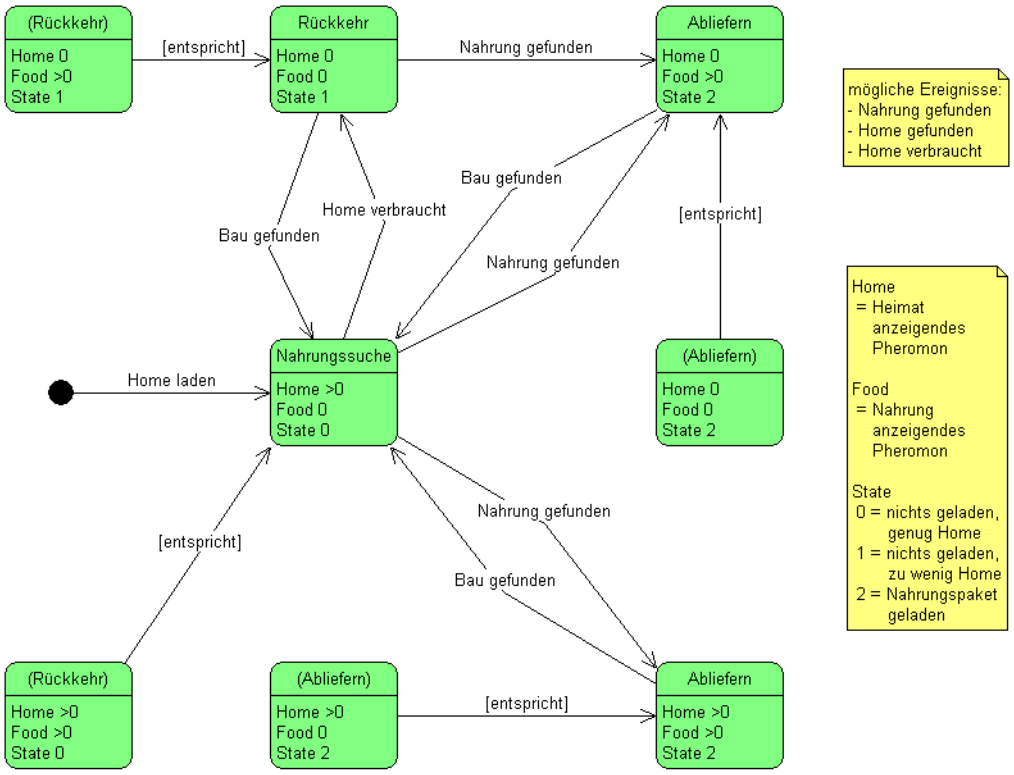

Abbildung 4: Zustände und Zustandsübergänge einer einzelnen Ameise

Von den acht Zuständen können jene vier vernachlässigt werden, die in der Praxis nicht erreicht werden können:

- Zu den Zuständen *(Food >0, State 0 oder 1)* (links oben und links unten in der Abbildung) führen keine Übergänge, und sie ergeben auch keinen Sinn – State wird nur 0, wenn Nahrung in der Basis abgeladen wird, und Food wird mit Inhalt gefüllt, wenn Nahrung gefunden, also State 2 wird.
- Die Zustände *(Food 0, State 2)* (unten und rechts) werden im Grundmodell nicht erreicht, weil zwar Food-Pheromon bei der Rückkehr zum Ameisenhügel verbraucht wird, dessen Wert jedoch niemals Null erreicht, abgesehen von möglichen Rundungsfehlern.

Die Anzahl der grundlegenden Aktionen beschränkt sich auf zwei:

1. Nahrungsquelle gefunden “nein” UND Heimat anzeigendes Pheromon nicht leer → **Nahrungssuche**: zufällige Bewegung, orientiert am Nahrungsquelle anzeigenden Pheromon, dabei wenn möglich das Heimat anzeigende Pheromon hinterlassen. (State = 0)

2. Nahrungsquelle gefunden "ja" ODER Heimat anzeigendes Pheromon leer
   → **Rückkehr**: zufällige Bewegung, orientiert am Heimat anzeigenden Pheromon, dabei wenn möglich das Nahrung anzeigende Pheromon hinterlassen. (State = 1)
   Optional ist hierbei der Transport eines Nahrungspaketes. (State = 2)

Weiterhin nehmen die Position der Ameise und die nähere Umgebung, die sie im Modell sensorisch erfassen kann, Einfluss wie folgt:

- Wenn Ameisenhügel erreicht
  → Heimat anzeigendes Pheromon auftanken und ggf. Nahrungspaket abladen
- Wenn Nahrungsquelle erreicht
  → Nahrung anzeigendes Pheromon auftanken und Nahrungspaket aufladen

2.3.2 Modellierung der Umwelt

Das Verwittern der Pheromone erfolgt durch eine Multiplikation des jeweiligen Feldes mit einem Abschwächungsfaktor zu jedem Zeitschritt. Optional sorgt eine Glättungsfunktion für das Verwischen der Werte, was dem Einfluss von Luftzirkulationen in der Natur entsprechen könnte.

Eine Begrenzung der Umwelt ist nötig, damit die Ameisen keine ungültigen Koordinaten ansteuern. Aufgrund ihrer nichtdeterministischen Bewegung genügt es nicht, die Konzentration der Pheromone am Rand künstlich niedrig zu halten. Es bieten sich diese beiden Möglichkeiten des "Abprallens" an:

1. "1 Feld drum herum frei lassen"
   Vorteil: Vermeidung ungültiger Koordinaten – die Ameisen dürfen sich in das Randgebiet hinein bewegen, müssen aber sofort wieder entfernt werden.
   Nachteil: Verschiebung des Koordinatensystems – im Quellcode muss bei allen Koordinaten +1 hinzugerechnet werden.
2. "exakte Grenzen"
   Vorteil: Es ist keine Umrechnung der Koordinaten nötig.
   Nachteil: Die Prüfung auf ungültige Koordinaten muss bereits während der Wegsuche erfolgen, dies bedeutet erhöhten algorithmischen Aufwand.

Ein kurzer Vergleichstest nach Implementierung beider Varianten lässt keine signifikanten Laufzeitunterschiede erkennen. Die Entscheidung fällt zugunsten der Koordinatenverschiebung aufgrund des einfacheren Algorithmus.

2.4 Entwurf

Nach der Initialisierung der Ameisen, der Umwelt, der Parameter sowie einiger Hilfsvariablen besteht das Programm (siehe Datei `ameisenstrassen.m`) im Wesentlichen aus einer großen Schleife über alle Zeitschritte – im Folgenden als Pseudocode dargestellt.

```
für jeden Zeitschritt:

    für jede Ameise:

        Bewegung
            abhängig vom eigenen Zustand an einem Pheromon orientieren
            Geradeausbewegung bevorzugen, Rückwärtsbewegung vermeiden
            auf den Rand der Umwelt achten
            ggf. Kollision mit anderen Ameisen beachten

        Ameisenhügel gefunden?
            Nahrungspaket geladen? -> abliefern
            Heimat anzeigendes Pheromon auftanken

        Futterstelle gefunden?
            kein Nahrungspaket geladen? -> aufladen

        Unterwegs
            auf Futtersuche?
                Heimat-Pheromon vorhanden? -> verteilen
            Nahrungspaket geladen? -> Futter-Pheromon verteilen

    Verwitterung und ggf. Glätten der verteilten Pheromone

Unterfunktion - Wegsuche:

    für jede mögliche Bewegungsrichtung:
        Wahrscheinlichkeit bestimmen

    Vektor aus den Wahrscheinlichkeiten für jede Richtung bilden
        Bsp: [ 3 1 2 4 ]
    Überhöhung aller Werte durch Potenzbildung (Verringerung des Zufalls)
        Bsp: [ 9 1 4 16 ]
    Bildung der kumulierten Summe über den Vektor
        Bsp: [ 9 10 14 30 ]
    Zufallswert in [0 Maximalwert] bestimmen, auf den Vektor anwenden
        Bsp: rand(30) = 12 -> [ 0 0  1  1 ]
    Position der ersten Eins bestimmen, als gewählte Richtung zurückgeben
        Bsp: 3
```

Listing 1: Pseudocode für den Ameisenalgorithmus

2.5 Potenzielle Probleme und Lösungsansätze

Allgemein ist bei der Programmierung mit Matlab im Vergleich zu "richtigen" Programmiersprachen zu beachten, dass Call-by-Reference nicht unterstützt wird. Call-by-Value kostet (für die Übergabe von Datenstrukturen anstatt eines einzelnen Datenwortes) mehr Ressourcen, mit der Konsequenz, dass zur höheren Geschwindigkeit bevorzugt mehr spezialisierter Code eingesetzt wird: z.B. mehrere fast-identische Funktionen, die sich aufgrund unterschiedlicher Bezeichner nur schwer oder unschön zusammenfassen lassen. Anstatt etwa Call-by-Value mit großen Matrizen zu nutzen, kommen bevorzugt globale Variablen und Seiteneffekte zum Einsatz, auch wenn der Stil wegen Redundanz darunter leidet.

2.5.1 Wie wird vermieden, dass mehrere Ameisen gleichzeitig Nahrung aufnehmen?

Das typische Verhalten von Zellularen Automaten sieht vor, dass trotz serieller Verarbeitung im Rechner eine Gleichzeitigkeit für alle Aktionen in einem Zeitschritt simuliert wird, wobei lediglich die *vergangenen* Zustände der Nachbarn zur Verfügung stehen, nicht jedoch die aktuellen, von denen unklar ist, ob sie in diesem Zeitschritt bereits verarbeitet worden sind oder nicht.

Die einfachste Lösung ist daher die Vermeidung echter Gleichzeitigkeit, indem die Ameisen in jedem Zeitschritt einzeln nacheinander bewegt werden und einander beachten können. Dabei wird die Verfügbarkeit der Nahrung für jede Ameise geprüft, und im Falle eines negativen Befundes passiert einfach gar nichts.

Für eine Parallelisierung des Algorithmus' stellen die Prozeduren der Nahrungsaufname und -abgabe kritische Abschnitte dar.

2.5.2 Was passiert, wenn mehrere Ameisen das gleiche Feld betreten wollen?

Es wäre möglich, die Kollision von Ameisen miteinander *gar nicht* zu prüfen, demnach könnten sich zu einem Zeitpunkt mehrere Ameisen auf dem gleichen Feld befinden, ohne voneinander überhaupt Kenntnis zu erlangen. Ein nicht zu missachtendes Problem dabei ist jedoch, dass bei "fast deterministischer" Richtungsbestimmung – d.h. weit gehender Orientierung am jeweiligen Pheromon und wenig Einfluss der Zufallskomponente – die Wahrscheinlichkeit groß ist, dass mehrere Ameisen in einem Zeitschritt ein Feld mit relativ hoher lokaler Konzentration betreten, und in den folgenden Zeitschritten sich ihre Richtungsbestimmungsfunktion aufgrund gleicher Ausgangsbedingungen stets für den gleichen Pfad entscheidet. Dies ist für das Ziel der Simulation – die Ausprägung einer Ameisenstraße – kaum relevant, doch wird die Visualisierung etwas... langweiliger: Es scheinen Ameisen zu verschwinden, weil für mehrere von ihnen über einen größeren Zeitraum nur noch ein gemeinsamer Punkt auf der Karte angezeigt wird.

Lösungsansätze

- Eine Ameise auf einem potenziellen Zielfeld wirkt als Subtrahend in Bezug auf die Wirkung des Pheromons – interpretierbar als ein weiteres, kleinräumig wirkendes Pheromon.
- Eine Ameise auf einem potentiellen Zielfeld verringert direkt die Wahrscheinlichkeit des Betretens weiterer Ameisen – Vorteil der wahrscheinlich einfacheren Modellierung.

Hierdurch würden Kollisionen zwar nicht ausgeschlossen, aber die Wahrscheinlichkeit ihres Auftretens verringert und ihre nachträgliche Auflösung ermöglicht.

Die Kollisionserkennung sollte jedoch behutsam ansetzen: es wäre kontraproduktiv, wenn eine Ameise ein Feld nicht betritt, weil sich dort bereits eine Ameise befindet, die sich noch im gleichen Zeitschritt zum Verlassen entscheidet.

Lösungsansätze

- Es wäre möglich, die Bewegungswünsche jeder Ameise zwischenzuspeichern und erst am Ende des jeweiligen Zeitschrittes auszuwerten. Auf ähnliche Weise wird bei den Hierarchischen Asymmetrischen Zellularen Automaten [SV01] eine virtuelle Schicht zwischen den Ebenen eingefügt.
- Die Reihenfolge der Bewegung könnte bei drohender Kollision umsortiert werden, etwa durch ein Anreihen ans Ende einer Warteschlange, sobald ein Zielfeld belegt ist, zur späteren Berechnung. Somit können jedoch Endlosschleifen entstehen, wenn zwei Ameisen den Platz tauschen wollen.
- Relativ simpel scheint auf den ersten Blick das Ignorieren einer auf dem Zielfeld vorhandenen Ameise, wenn diese einen höheren Index hat, also erst später in diesem Zeitschritt handelt. Das Problem hierbei ist jedoch der hohe Rechenaufwand zum Herausfinden des Indexes der anderen Ameise – bei den vorhandenen Datenstrukturen ist schnell zu sehen, *ob* dort jemand ist, aber nur langsam zu berechnen, *wer* dort ist.

3 Simulation

Schon früh stellt sich die Frage: Wie kann bewertet werden, ob überhaupt und zu welchem Zeitpunkt eine Ameisenstraße existiert? Eine Lösung dieses Problems – etwa ein Algorithmus, der zu einer gegebenen Pheromon-Verteilung eine Liste mit exakten Positionsangaben aller enthaltenen Ameisenstraßen berechnet – geht in den Bereich der Bildanalyse und kann an dieser Stelle leider nicht behandelt werden.

Für unsere Zwecke soll der scharfe Blick des Beobachters genügen, um zu erkennen, ob sich relativ viel Pheromon in einer mehr oder weniger linearen Anordnung gesammelt hat und/oder ob sich auffällig viele Ameisen über einen gewissen Zeitraum in entsprechender Form bewegen.

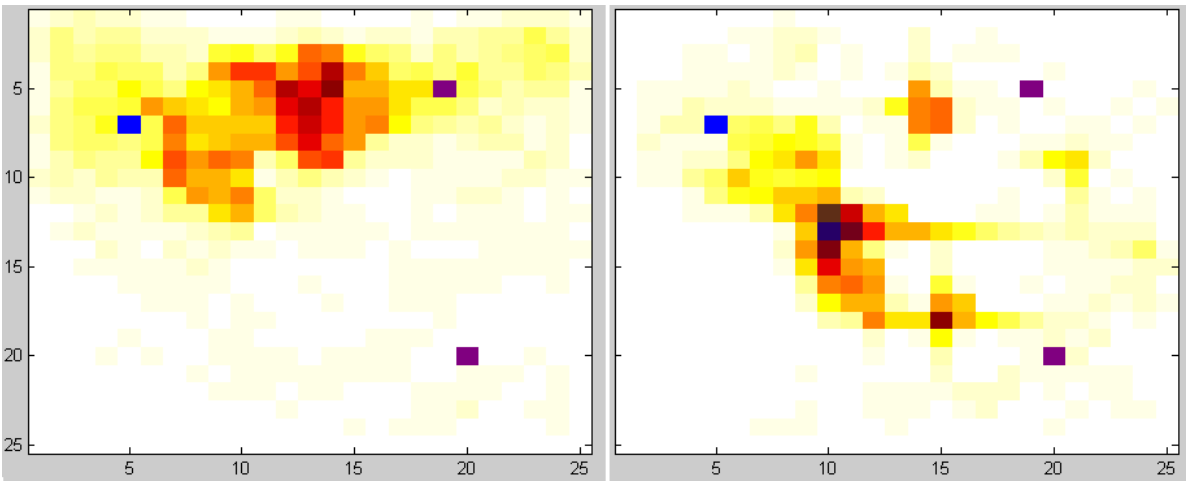

Abbildung 5: Ameisenstraßen – Beispiele / Kodierung der Pheromone in gelb bis rot

3.1 Beobachtungen

Abhängig davon, welche der Zusatzfunktionen – Glättung, Kollisionsvermeidung – aktiviert bzw. welche Parameter – insbesondere die Bevorzugung der Geradeausbewegung – gesetzt sind, bilden sich verschiedene charakteristische Strukturen aus. Es kann vorkommen, dass sich zu Beginn des Straßenbaus, bei ca. 500 bis 1000 Zeitschritten, mehrere "Hotspots" bilden, die sich (wenn sie günstig liegen) im Laufe der Zeit zu Straßen verbinden oder (eher abseits gelegen) verwittern – es braucht also eine gewisse Zeit zur Ausprägung von Straßen.

Bei mehreren Futterstellen etabliert sich meist nur zu einer Stelle eine Straße, die anderen bleiben wenig beachtet. Straßen zwischen zwei Futterquellen sind möglich, aber selten so kräftig markiert wie jene zum Ameisenhügel. Selten bilden sich zwei annähernd gleichberechtigte Straßen zu *einer* Futterstelle.

Schachbrett-artige Strukturen und diagonale Straßen ... entwickeln sich schon bei Experimenten mit frühen Prototypen und lassen eine "unrealistische" Bevorzugung der diagonalen Bewegungsrichtungen vermuten. Zur Kompensation scheint es sinnvoll, die Wahrscheinlichkeit jeder diagonalen Bewegung um einen Faktor entsprechend der Entfernung ($\sqrt{2}$) zum jeweiligen Ausgangspunkt zu verringern. Die Experimente zeigen jedoch, dass dieser Wert zu groß ist, indem nun fast ausschließlich horizontale und vertikale Straßen mit entsprechend rechtwinkligen Abzweigungen entstehen. Ein Faktor in der Größenordnung $1,2$ erweist sich als brauchbar.

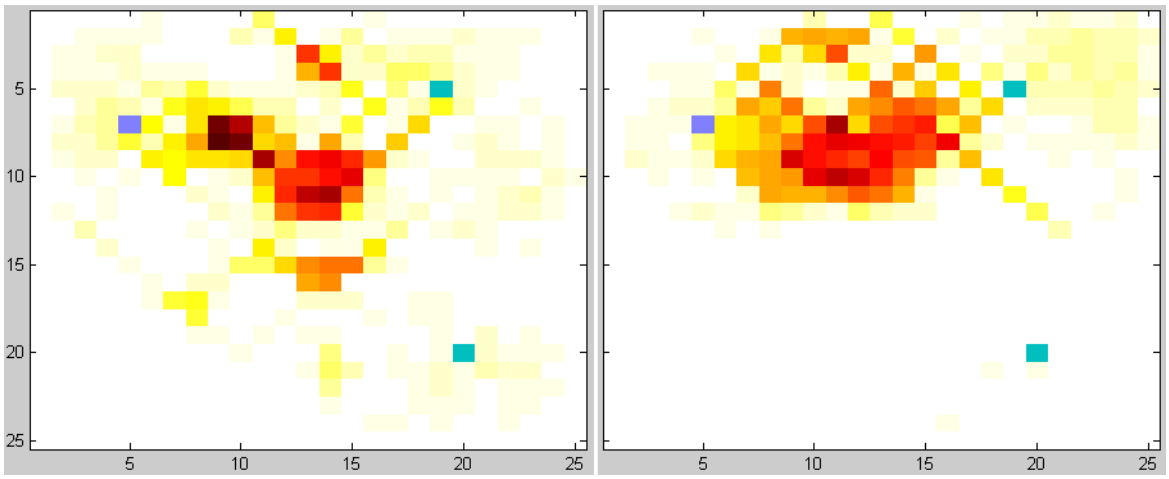

Abbildung 6: Ameisenstraßen – Diagonale Strukturen

Inseln ... seien Bereiche der Umwelt, in die sich Ameisen aufgrund hoher lokaler Pheromonkonzentration "verirrt" haben. Sie finden den Weg dort schwer heraus, sondern scheinen im Kreis zu laufen, obwohl sich in ihrer Nähe kein sinnvolles Ziel befindet. Es kann beispielsweise passieren, dass eine Ameise, die gerade ein Nahrungspaket aufgeladen hat, sich zu einer anderen Futterstelle verirrt, oder zu einem Kreuzungspunkt, wo sich viele andere Ameisen versammelt und dort dementsprechend viel des Heimat anzeigenden Pheromons verbreitet haben – so dass sie länger als nötig dort bleibt, anstatt sich zügig zum Ameisenhügel zu bewegen, und dabei wiederum (fälschlicherweise) relativ viel des Nahrung anzeigenden Pheromons hinterlässt, so dass weitere Ameisen angelockt werden. Im Extremfall halten sich so *alle* im System vorhandenen Ameisen auf nur zwei Feldern auf und bewegen sich immer zwischen diesen beiden hin und her.

Tatsächlich ist dies zwar genau das Prinzip der Entstehung einer Ameisenstraße – und somit ist eine Ameisenstraße "nur" eine spezielle Form der Insel – allerdings sollte die Straße bevorzugt eine größere, möglichst lineare Strecke umfassen und nicht einen kleinen Fleck abseits der besten Route sowie sinnvolle Endpunkte.

Befindet sich eine Insel in einiger Distanz vom Geschehen, so kann sie sich im Laufe der Zeit von selbst auflösen, sobald die Pheromon-Vorräte der betroffenen Ameisen aufgebraucht sind und die Spuren allmählich verwischen. Glücklicherweise können die

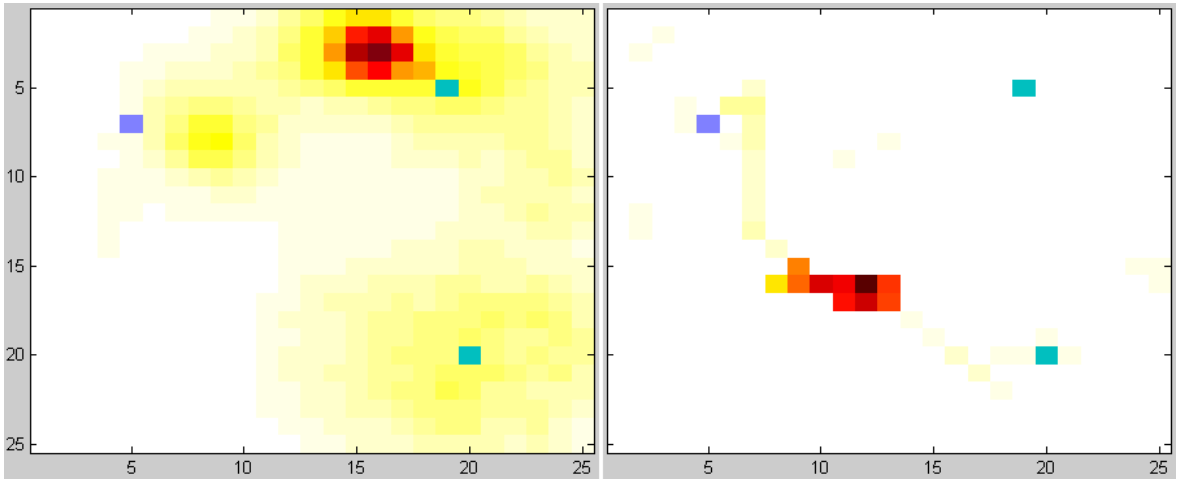

Abbildung 7: Insel nahe einer Futterstelle / Beinahe-Ameisenstraße

Modell-Ameisen nicht verhungern – und in der Natur wird (soweit der Autorin bekannt) bei staatbildenden Insekten wenig Rücksicht auf die Gesundheit der Individuen genommen. Im ungünstigen Fall jedoch kann eine Straßenverbindung blockiert werden, weil effektiv die Anzahl der markierenden Ameisen im System verringert wird.

Um die Entstehung solcher Inseln von vornherein zu vermeiden, dürfte eine mäßige Förderung der *Geradeausbewegung* sowie Vermeidung der Rückwärtsbewegung hilfreich sein. Ebenso sollte eine *Glättungsfunktion* für die Pheromonkonzentration die Wahrscheinlichkeit ihres Auftretens senken können, und die *Kollisionsvermeidung* der Ameisen vermindert die Bindung einer zu großen Zahl von Ameisen auf engem Raum. Schließlich könnte die Einführung eines *Schwellwertes* für die Pheromone helfen, oberhalb dessen sich die Ameisen von ihrer Position entfernen, anstatt weiter das gleiche Pheromon zu verbreiten.

3.2 Untersuchungen

Nun, da das Modell an sich erfolgreich simuliert werden kann und erste funktionierende Parameter gefunden worden sind, mit denen in ungezielten Experimenten relativ häufig "schöne" Straßen entstehen, können erweiterte Fragestellungen betrachtet werden.

Als ein Maß für die **Effizienz** einer Straße könnte die Anzahl der über sie transportierten Nahrungspakete pro Zeiteinheit gelten. Da es jedoch schwer zu bestimmen ist, welche Ströme abseits der Straße fließen, muss die "Transportleistung" des Gesamtsystems als Indikator genügen. Dafür soll im Folgenden die Zeit gelten, die das System unter festen Randbedingungen benötigt, um eine bestimmte Anzahl Nahrungspakete von zwei Futterstellen zu einem Ameisenhügel an willkürlich gewählten (aber konstanten) Positionen in der Umwelt zu transportieren. Die Aussagekraft leidet unter dem großen stochastischen Einfluss auf die Bewegung, weswegen eine Mittelung aus mehreren Simulationsläufen sinnvoll ist.

Es werden nun beispielhaft einige auf den ersten Blick interessante Parameter systematisch variiert, während andere zur Einschränkung der Vielfalt für jeden Lauf festgehalten werden, ihre Startwerte abgeschätzt aus unzähligen Tests mit Prototypen während der Entwicklung. Die variablen Parameter erhalten Standardwerte, während sich jeweils einer von ihnen verändert. Lediglich die Parameter *Anzahl der Ameisen* und *Ausdehnung der Umwelt* werden in Kombination miteinander betrachtet. Als Kompromiss aus Simulationsdauer und Genauigkeit werden die Ergebnisse von jeweils 5 Simulationsläufen (Stichproben) werden gemittelt.

Technisch ausgedrückt wird für jeden Parameter das Minimum einer Funktion gesucht, welche die Effizienz des Gesamtsystems in Abhängigkeit jenes Parameters berechnet.

Parameter	**Wert**
Nachbarschaftsgröße	8
Präsenz-Konstante für Ameisenhügel und Futterquellen	2
Minimalwert des Heimat-anzeigenden Pheromonvorrats	0.01
Abschwächungsfaktor gegen Diagonalbewegung	0.8
Abschwächungsfaktor gegen Rückwärtsbewegung	0.01
Pheromon-Verteilung pro Zeitschritt	0.01
zu transportierende Nahrungspakete	$2 * 20$
verzögerte Öffnung der Futterstellen	$false$

Tabelle 3: Startwerte der festen Parameter während der Untersuchungen

Parameter	**Wert**
Anzahl der Ameisen	50
Ausdehnung der Umwelt	$25 * 25$
Verwitterungsfaktor	0.01
Grenzwert für die Pheromone in der Umwelt	0.2
Faktor zur Bevorzugung der Geradeausbewegung	1.7
Überhöhungsfaktor zur Verminderung des Zufalls	1.5
Faktor zur Kollisionsvermeidung	0.3
Glättung der Pheromone (aktiv/inaktiv)	$false$

Tabelle 4: In den Untersuchungen zu verändernde Parameter

3.2.1 Variation der Ameisendichte

Erwartungsgemäß ist eine Abhängigkeit der Transportleistung des Systems von der Dichte der Ameisen in der Umwelt deutlich erkennbar (Tabelle 5). Überraschend scheint jedoch ein kleines Optimum bei einer Anzahl von 40 Ameisen zu sein; eine größere Anzahl in einer relativ kleinen Umwelt bringt keinen signifikanten Vorteil.

Umwelt	Ameisen					
	25	30	35	40	45	50
$15 * 15$	569	396	460	421	430	382
$20 * 20$	1151	1114	981	770	832	834
$25 * 25$	1887	1538	1287	1239	1294	1317
$30 * 30$	–	2570	2542	1992	1664	1566
$35 * 35$	–	–	–	–	2475	2659

Tabelle 5: Untersuchung: Ameisendichte

Es gibt einen Schwellwert der Dichte, unterhalb dessen die Transportleistung des Systems stark abfällt: er liegt bei etwa 0,05 Ameisen pro Feld. Ab einer Dichte von etwa 0,1 Ameisen pro Feld sind die Unterschiede bei verschiedenen Feldgrößen im Bereich der Messtoleranz. Die kürzeste absolute Rechenzeit wurde bei vielen Ameisen auf kleiner Fläche erreicht – 14 bis 21 Sekunden für 40 Ameisen auf 15x15 Zellen im Vergleich zu 90 bis 135 Sekunden für 50 Ameisen auf 25x25 Zellen.

Möglicherweise ist die Auswahl der Bewertungszahl schlechter geeignet als zunächst gedacht, da in der grafischen Ausgabe gerade kurz vor Abschluss der Simulation häufig einzelne Ameisen abseits der Straßen unterwegs waren und für den Rückweg entsprechend länger brauchten. (Subjektiv betrachtet scheint am Ende immer genau eine “blinde” Ameise übrig zu bleiben, die den Weg partout nicht findet.)
Eine Alternative ist die Beschränkung auf einen gewissen Anteil, etwa 80% der verfügbaren Nahrung, um die Irrläufer zu egalisieren – dies wird zwecks Zeitersparnis für die folgenden Untersuchungen genutzt, d.h. es müssen zur Bewertung der Effizienz nur 32 von 40 Nahrungspakete ihr Ziel erreichen.

3.2.2 Variation des Verwitterungsfaktors

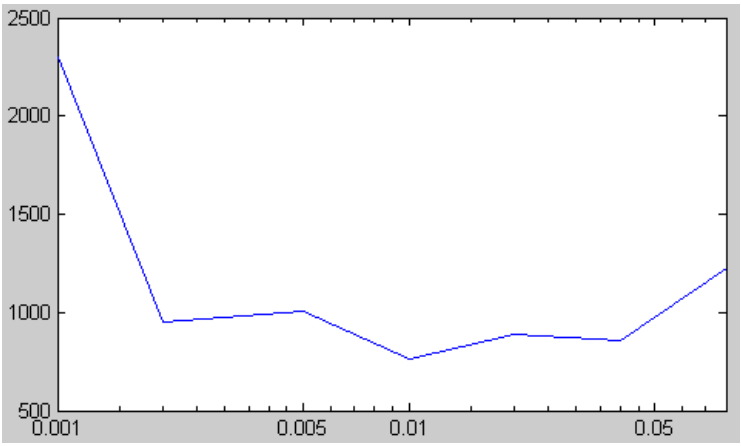

Abbildung 8: Untersuchung: Verwitterungsfaktor

Das Modell wird mit sechs verschiedenen Werten für den Verwitterungsfaktor jeweils fünf Mal berechnet. Abbildung 8 zeigt das Ergebnis: der Wert 0.01 ist gut gewählt, mit dem das Transportziel im Mittel nach 760 Zeitschritten erreicht wird.

3.2.3 Variation des Grenzwertes für die Pheromon-Verteilung

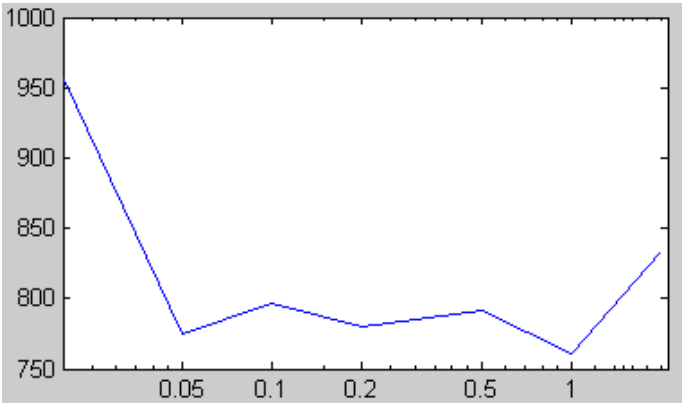

Abbildung 9: Untersuchung: Grenzwert für die Pheromon-Verteilung

Es werden sieben verschiedene Werte als Grenzwert für die Pheromon-Verteilung herangezogen – jene Schwelle, oberhalb der die Wahrscheinlichkeit verringert wird, dass die Ameisen das betroffene Feld betreten. Wiederum sieht es so aus (Abbildung 9), dass der vorab gewählte Wert von 0.2 im grünen Bereich liegt.

3.2.4 Variation des Faktors zur Bevorzugung der Geradeausbewegung

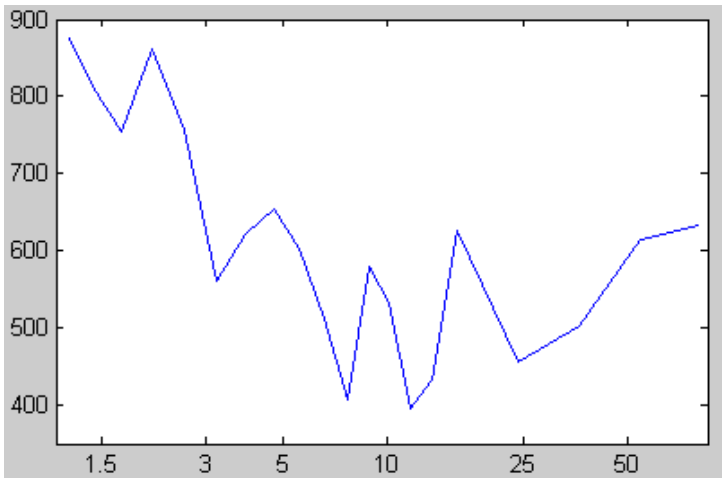

Abbildung 10: Untersuchung: Faktor zur Bevorzugung der Geradeausbewegung

Bei diesem Experiment erweist sich der Startwert von 1.7 als eher schlecht gewählt, denn offenbar bleibt noch einiger Spielraum nach oben (Abbildung 10). Ein Wert von etwa 10.0 scheint die Leistung des Systems annähernd zu verdoppeln – mit dem Schönheitsfehler, dass die grafische Ausgabe der Umwelt durch schnurgerade Verbindungen und Schachbrettmuster nun weniger "realistisch" aussieht (Abbildung 11).
Eine Vermutung geht in die Richtung, dass diese Art der Strukturen die Streuung vergrößert: es kann durch Zufall eine sehr gute Straße entstehen, es können aber auch leichter Verbindungen abseits ihrer optimalen Position gebildet werden, als dies bei konservativerer Parameterwahl der Fall wäre. So kommt es vor, dass eine Straße zu einer der beiden Futterstellen so kräftig markiert ist, dass es sehr viel Zeit braucht, um die andere Futterstelle überhaupt zu entdecken.
Als neuer Standardwert für die weiteren Untersuchungen wird nun 4.0 festgelegt – ein Kompromiss zwischen Effizienz und Natur.

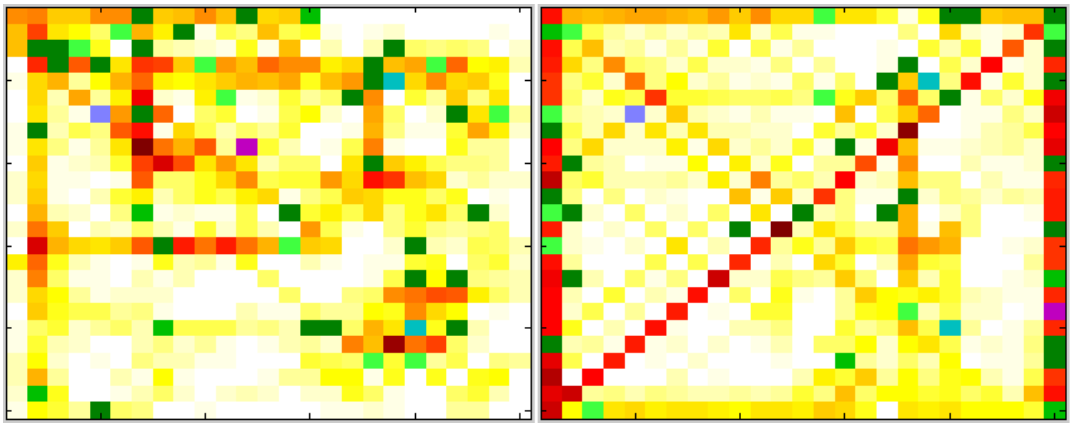

Abbildung 11: Geradeausbewegung extrem verstärkt – typische Strukturen

3.2.5 Überhöhungsfaktor zur Verminderung des Zufalls

Erneut erweist sich der vorab gewählt Wert von 1.5 als relativ effizient (Abbildung 12). Eine geringfügige Anpassung auf 1.8 scheint dennoch sinnvoll.

3.2.6 Faktor zur Kollisionsvermeidung

Obwohl es in der Natur keine Staus oder Kollisionen an Engpässen gibt, unabhängig von der Anzahl der Ameisen [Wiki01], ist die Vermeidung im Modell sinnvoll.
Eine geringe Kollisionswahrscheinlichkeit (durch aktives Ausweichen) sorgt dafür, dass vorhandene Straßen weniger intensiv genutzt, sondern Alternativen gesucht werden. Außerdem kann, ebenso wie mithilfe der bevorzugten Geradeausbewegung, die Bildung von Inseln (Kapitel 3.1) vermieden werden, indem das Bedürfnis der Ameisen nach Plätzen mit viel Pheromon bzw. vielen Artgenossen gering gehalten wird.

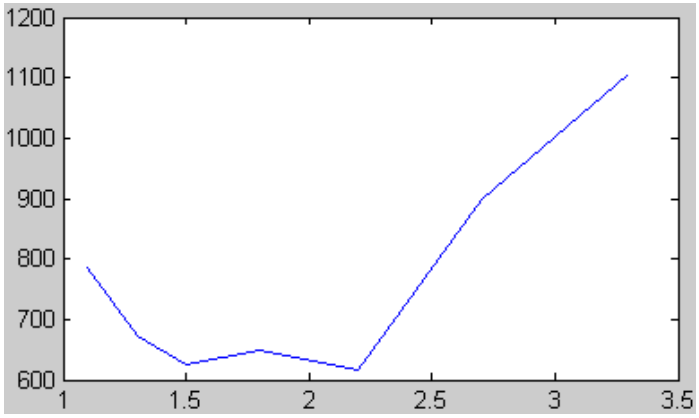

Abbildung 12: Untersuchung: Überhöhungsfaktor zur Verminderung des Zufalls

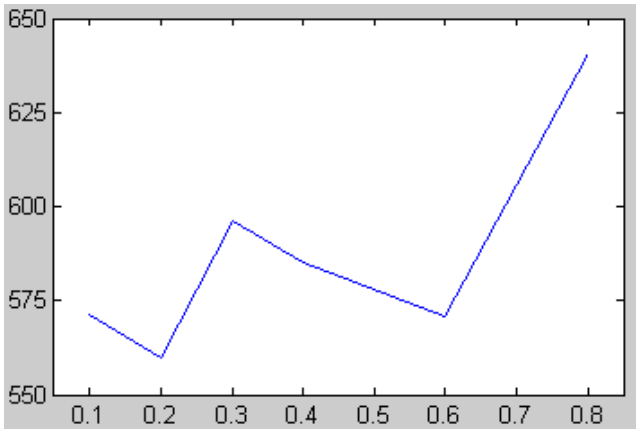

Abbildung 13: Untersuchung: Faktor zur Kollisionsvermeidung

Der Faktor zur Verringerung der Wahrscheinlichkeit des Betretens eines Zielfeldes, falls sich dort bereits eine andere Ameise aufhält, von etwa 0.3 erweist sich als angemessen und wird beibehalten.

3.2.7 Glättung der Pheromone

Auch wenn die Visualisierung der Pheromone schon bei schwachem Einfluss der Nachbarfelder hübsch anzuschauen ist (Abbildung 14), ergibt sich im ersten Versuch bei aktivierter Glättungsfunktion mit einer Abweichung von 0,5% keine signifikante Änderung in der Transportleistung des System.
Erst bei deutlicher Vergrößerung des Einflusses, so dass entstehende Strukturen schnell großflächig verwischen, wirkt sich der Algorithmus geringfügig positiv auf die Leistung aus – die Ameisen brauchten im Schnitt nur 560 statt 575 Zeitschritte, um ihre Aufgabe zu erfüllen.
Ausnahmsweise wurde für dieses Experiment die Berechnung für jede Parameterkombination zehn statt nur fünf Mal durchgeführt, um exaktere Mittelwerte zu bekommen.

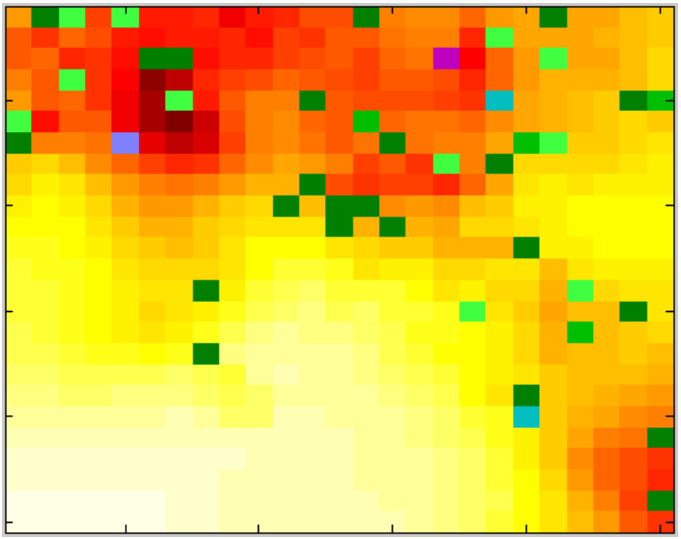

Abbildung 14: Untersuchung: Glättungsfunktion aktiviert – typische Verteilung

3.2.8 Ergebnis

Die Aussagekraft der Untersuchungen ist eingeschränkt aufgrund der begrenzten Anzahl der Wiederholungen. Idealerweise müsste jedes Experiment nicht nur 5, sondern besser 20 oder vielleicht 100 Mal wiederholt und aus diesen Messwerten der Mittelwert gebildet werden, da während der Durchführung teilweise starke Schwankungen zu beobachten waren. Dennoch sind in den meisten Fällen klare Tendenzen erkennbar, und es scheint, dass das Optimierungspotenzial durch die analytische Betrachtung im Vergleich zur menschlichen durch "scharf angucken und damit spielen" relativ gering ist: in fast allen Fällen liegen die vorab gewählten Parameter bereits in der Nähe der jeweils optimalen Bereiche.
Weiterhin unklar bleiben vorerst die möglichen Abhängigkeiten der Parameter untereinander – beispielsweise könnte bei einer anderen Ameisendichte in veränderter Umwelt bei anderen Positionen der Futterstellen theoretisch eine völlig andere Parameterkombination zur effizientesten Transportleistung des Systems führen. Außerdem könnte, über die Messung der Zeitdauer für einen bestimmten Transfer hinaus, die Geschwindigkeit des Transports zu jedem Zeitpunkt näher betrachtet werden.

3.3 Optimierungen

... können an dieser Stelle lediglich Ideen sein – dafür unterscheidet sich Matlab zu stark von "echten" Programmiersprachen wie C++, bei denen die Belegung von Prozessor und Speicher direkter zu beobachten und zu beeinflussen ist.

Vermutlich aufgrund Matlab-interner Optimierungen für Matrizen und der Anpassung an die Matlab-Besonderheiten (Stichwort Call-by-Value) wirkt sich eine Vergrößerung

der Umwelt nur minimal auf die Rechenzeit des Modells aus.[1] Die Anzahl der Ameisen hingegen wirkt sich fast proportional auf die Rechenzeit aus. Mäßigen, aber signifikanten Einfluss auf die Rechenzeit hat die grafische Darstellung per `imagesc`. Eine Maßnahme ist daher, auf die Anzeige jedes einzelnen Zeitschrittes zu verzichten, sondern über einen "Teiler" nur z.B. jedes zehnte Bild auf den Monitor zu bringen. Sinnvoll sind Werte bis zu etwa 10 – darüber ist der Gewinn an Rechenzeit in einem Testlauf kaum messbar.[2]

Eine weitere, echte Optimierung auf Geschwindigkeit könnte unter günstigen Umständen (relativ wenige Ameisen in einer sehr großen Umwelt) sein, die Berechnung der Abschwächung der Pheromone bei jedem Zeitschritt nicht mehr für alle Felder zu berechnen, sondern nur einzeln für die betroffenen Felder, etwa beim Betreten durch eine Ameise – das wäre ein Übergang in Richtung komplett Ereignis-gesteuerter Simulation ohne einheitliche Taktung des zellularen Automaten. Um herauszufinden, ob sich das lohnt, müsste zunächst analysiert werden, welcher Teil der Berechnung überhaupt am meisten Rechenzeit kostet.

3.4 Erweiterungen

An dieser Stelle nur ein paar Ideen, die jedoch spekulativ oder unverhältnismäßig erscheinen. Denn für alle Erweiterungen gilt: Sie machen die Sache eigentlich nur unnötig kompliziert und sind bestenfalls Spielerei, solange es keine dazu passende interessante Fragestellung gibt.

Arbeitsteilung? Mindestens zwei Spezialisierungen von Ameisen: Typ A verteilt nur das Heimat-anzeigende Pheromon und orientiert sich *nicht* an anderen Pheromonen, sondern bewegt sich mit starker Zufallskomponente, so dass eine gleichmäßige Verteilung entsteht, um Flüchtlingen von den "Inseln" den Weg zu weisen. Typ B sucht nur nach Nahrung und könnte sich dabei (von Zeit zu Zeit) aktiv vom Heimat-anzeigenden Pheromon entfernen, um "gezielt" nach neuen Futterquellen zu suchen.

Hindernisse? Man könnte die Fähigkeit der Ameisen, sich bei Störungen schnell und flexibel anzupassen (was für technische Anwendungen besonders geschätzt wird), auf die Probe stellen, indem plötzlich und zufällig einige Felder durch "Hindernisse" blockiert werden. Beispielsweise wäre es nicht ganz unrealistisch, wenn an einer stark genutzten Straße ein Räuber auftauchen würde, dem die Ameisen mithilfe ihres Geruchssinns aus dem Weg gehen müssen. Oder das Terrain wird generell heterogener, und es gäbe Elemente wie Brücken und Täler.

[1] Eine Vergrößerung der Umwelt von 40 auf 90 oder 160 Felder bewirkt eine Verlängerung der Rechenzeit um 3,3% bzw. 5,2%.

[2] Im Vergleich zur Simulationsdauer bei der Anzeige jedes Einzelbildes kostet die Berechnung bei jedem 2. Bild etwa 88%, bei jedem 5. Bild etwa 78%, bei jedem 10. Bild etwa 74%, bei jedem 50. Bild etwa 73% und bei komplettem Verzicht auf jede grafische Ausgabe gerade mal 72% der Zeit.

Konkurrenz? Es könnte sich eine zweite, konkurrierende Ameisenart im System befinden, die ihre Nahrung bevorzugt stiehlt, was sich durch plötzlich und zufällig auftretenden Verlust von Nahrungspaketen bemerkbar machen würde. Darauf müssten die friedlichen Ameisen durch Strategieanpassung oder Arbeitsteilung reagieren. Es ginge auch ohne Diebstahl, indem einfach zwei Arten mit verschiedenen Eigenschaften (Parametern) um die gleichen Nahrungsquellen im Wettbewerb stehen.

Energie? Die Ameisen sammeln die Nahrung nicht nur ein, sondern verbrauchen sie auch und können verhungern. Sie sollten sich dann aber auch fortpflanzen können... In Verbindung mit der Konkurrenz (Selektion) und der mäßigen Änderung der Parameter von Generation zu Generation (Mutation) erschließt sich sodann die Klasse der *genetischen Algorithmen*.

3.5 Fazit

Ohne konkret bis zu diesem Punkt geplant zu haben, hat sich aus einer eher unspezifischen Aufgabenstellung nicht nur ein funktionierendes Ameisenmodell, sondern dazu eine möglicherweise nützliche Simulationsumgebung entwickelt, die sich vielfältig konfigurieren lässt, und mit der nun weitere Experimente durchführt werden können. Grundlegende Untersuchungen wurden durchgeführt, typische Strukturen beobachtet und Lösungsansätze für aufgetretene Probleme beschrieben.

Wiederkehrende Problematik bei den meisten Experimenten ist und bleibt der Grad der Orientierung der Ameisen an den Pheromonen bzw. im Gegenzug die Stärke des Zufallseinflusses für die Bewegung: Es gilt einen Kompromiss zwischen diesen beiden Faktoren zu finden, durch welche die Simulation wesentlich beeinflusst wird. Ist der Einfluss der Zufallskomponente zu hoch, so dass die Wahrscheinlichkeiten bei der Bestimmung des nächsten Schrittes nur geringfügig durch die Pheromone beeinflusst werden, so versagt die Kommunikation der Ameisen untereinander – sie finden die Spuren ihrer Kollegen nicht ausreichend sicher. Ist die Orientierung an den Pheromonen zu groß – im Extremfall bei Deaktivierung der Zufallskomponente – so können die Ameisen leicht einer falschen Spur folgen und sich in Sackgassen verirren.

Aber auch andere Komponenten, insbesondere die Förderung der Geradeausbewegung, der Schwellwert zur Vermeidung hoher Pheromonkonzentrationen und die Kollisionsvermeidung, haben großen Einfluss auf die Effizienz eines Ameisentransportsystems, gemessen in Nahrungspaketen pro Zeit, und müssen sorgfältig angepasst werden.

4 Anwendung

Das Verhaltensmuster der Ameisen – *Markierung von Wegen, Bevorzugung stärker markierter Wege* – kann bei vielen Problemen helfen, die mit dem Problem des Handlungsreisenden verwandt sind. Der Ansatz gilt als Metaheuristik und hat Ähnlichkeiten mit genetischen Algorithmen und neuronalen Netzen. Die erste Übertragung auf einen Computeralgorithmus gelang Anfang der 1990er Jahre dem Italiener Marco Dorigo [Dor].

4.1 Das Problem des Handlungsreisenden

> "Das Problem des Handlungsreisenden (engl. Travelling Salesman Problem, TSP) ist ein kombinatorisches Optimierungsproblem der Mathematik und der theoretischen Informatik. Die Aufgabe besteht darin, eine Reihenfolge für den Besuch mehrerer Orte so zu wählen, dass die gesamte Reisestrecke des Handlungsreisenden nach der Rückkehr zum Ausgangsort möglichst kurz ist. [...] Die Vorgehensweise in der Praxis unterscheidet sich von der mathematischen Theorie dadurch, dass in der Regel nicht nach einer optimalen Lösung gesucht wird, sondern nur nach einer ausreichend guten." [Wiki03]

Kennzeichnende Eigenschaft dieses Problems ist, dass es nur mit hohem Aufwand exakt zu lösen ist, weil sich die Anzahl der möglichen Wege mit jedem hinzu kommenden Ort (bzw. jeder im Graphen hinzu kommenden Kante) nach der Formel $(n-1)!$ vervielfältigt. Aufgrund der einfachen Verständlichkeit ist es in der Lehre ein Standardbeispiel für die Komplexitätsklasse NP – viele Probleme dieser Klasse lassen sich (bisher) nur mit exponentiellen Algorithmen exakt lösen.

Zur einfachen Lösung des Problems kann man alle möglichen Wege aufzählen und den kürzesten heraussuchen; allerdings gibt es bereits bei nur 15 Orten mehr als 87 Milliarden mögliche Rundreisen. Daher wurden im Laufe der Zeit verschiedene Verfahren entwickelt, um den Aufwand zu reduzieren – darunter sowohl exakte als auch heuristische Verfahren.

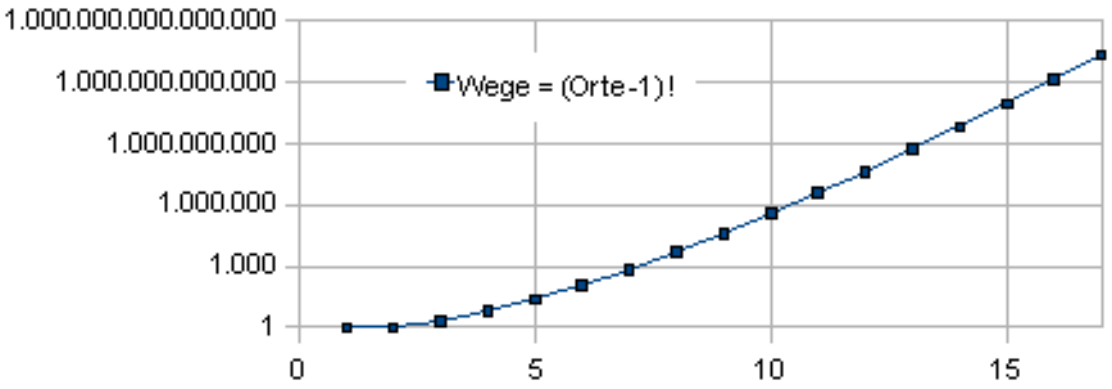

Abbildung 15: Problem des Handlungsreisenden – Anzahl der Wege

Kennzeichnend für eine Heuristik ist, dass innerhalb relativ kurzer Zeit zwar Näherungen an ein mögliches Optimum gefunden werden, allerdings nicht zwingend ein optimaler Wert; eine Heuristik kann grundsätzlich auch beliebig schlechte Ergebnisse liefern.

Zur Übertragung des in Kapitel 2 entwickelten Ameisen-Wegmarkierungs-Algorithmus auf das Problem des Handlungsreisenden muss zunächst ein völlig anderer Ansatz in der Repräsentation der Umwelt gewählt werden – ein Zellularer Automat kommt hier nicht infrage. Die Umwelt besteht aus einer Menge von Orten mit beliebigen Koordinaten, die miteinander durch Wege verbunden sind, von denen nur die Entfernung zählt. Dies lässt sich als Graph in Form eines voll vermaschten Netzes darstellen, für dessen Speicherung sich eine $n \times n$ -Matrix anbietet, welche die Distanzen zwischen den n Orten enthält. Eine weitere Matrix der gleichen Ausdehnung wird für die Ablage der Pheromon-Markierungen verwendet. Pheromone und Distanzen werden gemeinsam zur Bewertung eines Weges vor Betreten durch eine Ameise herangezogen.

Im Unterschied zum Matlab-Ameisensimulator werden die Pheromone nicht den Knoten zugeordnet, sondern den Wegverbindungen zwischen ihnen. Ihre Verteilung findet nicht bei jedem Schritt einer Ameise statt, sondern erst nach einer kompletten Rundreise.

Die Ameise als Individuum spielt im neuen Modell keine Rolle; sie wird nicht als Objekt im System gespeichert, sondern als Teil einer Gruppe für die Zeit ihrer Aktion erzeugt. Die Gesamtstrecke jeder Ameise wird nicht indirekt über den schrumpfenden Pheromonvorrat betrachtet, sondern fließt ein in die Zielwertfunktion – die Berechnung der "Kosten" eines Weges.

Aus Sicht der Ameise bleibt ihre Suche ein Random Walk, bei dem sie für jeden Schritt eine neue Berechnung durchführt und sich dabei neben der nun potenziell unterschiedlichen Distanz zu den jeweiligen Bewegungszielen wiederum an der Höhe der Pheromonkonzentration orientiert: je kürzer ein Weg, doch je stärker markiert, desto niedriger sind seine Kosten, desto häufiger wird er ausgewählt.

4.2 Entwurf eines Ameisenalgorithmus

Der folgende Entwurf für *Matlab* orientiert sich an der *Java*-Implementierung in [Boy05], die wiederum die "Ant Colony Optimization"-Metaheuristik [ACO] von Marco Dorigo zum Vorbild hat. Enthalten ist bereits die Optimierung, dass nur die jeweils beste Ameise einer Gruppe den Weg markieren darf. Der vollständige Quellcode inklusive grafischer Darstellung befindet sich in der Datei `tsp.m`.

Als Testdaten während der Entwicklung dienen zunächst einerseits Zufallswerte für die Distanzen zwischen 20 Orten[3] und andererseits realistische Daten, berechnet aus Koordinaten für 52 Orte[4] in Berlin, für die von [TSPLIB] eine optimale Lösung als Referenzwert zur Verfügung steht.

[3] $(20-1)! = 1,21 * 10^{17}$ Routen
[4] $(52-1)! = 1,55 * 10^{66}$ Routen

Objekte	**Datentypen**	**Wertebereiche**
Anzahl der Orte	Ganzzahl	$N \in [1\ \infty]$
Distanzen	Feld: Fließkommazahlen	$[1\ N]\ [1\ N]$ $[0.0\ \infty]$
Pheromone	Feld: Fließkommazahlen	$[1\ N]\ [1\ N]$ $[0.0\ \infty]$
Zähler für Ameisen, Gruppen, Distanzen	Ganzzahlen	$[0\ \infty]$
Listen für Zielwerte	Vektoren: Fließkommazahlen	$[1\ N]$ $[0.0\ \infty]$
Listen für Routen	Vektoren: Ganzzahlen	$[1\ N]$ $[1\ \infty]$
Farbschema für die Falschfarben-Kodierung	Tabelle: Fließkommazahlen	$\{rot, gruen, blau\}$ $[0.0\ 1.0]$

Tabelle 6: Datenstrukturen für das Handlungsreisendenproblem

```
1   für jede Ameisengruppe:
2
3       für jede Ameise in der Gruppe:
4
5           Startpunkt zufällig auswählen
6
7           solange es noch unbesuchte Ziele gibt:
8
9               für jedes mögliche Ziel
10                  Wert der Zielfunktion bestimmen
11                      aus Distanz und Pheromonkonzentration
12              aus allen Zielwerten per Zufall einen auswählen
13                  (vergleichbar der Unterfunktion Wegsuche in Kap. 2)
14              aktuelles Ziel aus der Liste der möglichen Ziele entfernen
15              Gesamtdistanz aufaddieren
16              nächstes Ziel vormerken
17
18          Rückweg vom letzten Ziel zum Startpunkt addieren
19          Strecke der Rundreise mit der bisher kürzesten der Gruppe
20              vergleichen und diese ggf. aktualisieren
21
22      für jeden Weg der besten Route dieser Gruppe:
23          Pheromon ablegen abhängig von der Länge der Route
24
25      Verwitterung des Pheromons gleichmäßig auf allen Wegen
```

Listing 2: Pseudocode für das Handlungsreisendenproblem

Bei der grafischen Darstellung zeigen sich Unterschiede, sowohl bei den Distanzen als auch im Verlauf der Wegsuche. In der Falschfarbenansicht der Distanzmatrix für die realen Daten (Abbildung 16) sind deutliche Strukturen erkennbar, die auf relativ lange lokale Entfernungen zwischen einzelnen Orten hindeuten. Weiterhin braucht der virtuelle Ameisenstaat für diese Daten eine Art "Erkundungsphase", die für Zufallsdaten (Abbildung 17) fehlt – dort werden im direkten Vergleich viel schneller gute Routen gefunden, auf den ersten Blick unabhängig von der Anzahl der Orte.
Aufgrund dieser Unterschiede werden für die folgenden Tests weitere Matrizen mit 50 zufällig verteilten Orten vorberechnet.

4.3 Beobachtungen

Für die realen Daten der Orte aus Berlin ist die Entwicklung einer S-Kurve für den Verlauf der Wegsuche typisch, vergleichbar einer gespiegelten logistischen Funktion, unterteilt in drei aufeinander folgende Phasen:

1. *Erkundung*: Da noch wenig Pheromone im System verteilt sind, bewegen sich die Ameisen relativ zufällig. Der Graph verläuft annähernd horizontal.

2. *Orientierung*: Die Markierungen gewinnen an Kontrast, sind aber noch schwach genug ausgeprägt, dass häufig Abweichungen ausprobiert und fast regelmäßig kürzere Routen gefunden werden. Der Graph fällt.

3. *Stabilisierung*: Die Mehrzahl der Ameisen bewegt sich auf den gut markierten Strecken, nur selten wird durch mutige Einzelgänger eine kürzere Route gefunden. Der Graph der Bestwerte verläuft wieder horizontal, die Mittelwerte jeder Ameisengruppe nähern sich den Bestwerten an.

Für die zufälligen Daten hingegen scheint keine Erkundungsphase nötig zu sein, so dass die Ameisen vom Start weg relativ schnell gute Routen finden können und der Verlauf einem exponentiellen Rückgang ähnelt.

In jedem Fall entsteht ein hoher Kontrast in der Pheromon-Matrix, der die Markierung der kurzen Wege widerspiegelt. Die müssen jedoch nicht immer eindeutig sein, sondern es sind mehrere "parallele" Wege (mit vermutlich ähnlicher Länge) möglich, über die sich die Ameisen auch über längere Zeit relativ gleichmäßig verteilen, obwohl man ein instabiles Gleichgewicht vermuten könnte.

Welchen Einfluss die Parameter jeweils exakt haben, ist schwer zu erkennen, da der Zufall im Modell eine große Rolle spielt. Nach Definition einer Bewertungsfunktion für die Effizienz des Gesamtsystems wären wiederum umfangreiche statistische Tests nötig. Die Effizienz könnte beispielsweise durch die Anzahl der Ameisengruppen repräsentiert werden, die im Durchschnitt durch das System laufen müssen, bis eine vorgegebene Grenze (z.B. 10% oberhalb einer bekannten optimalen Lösung) erreicht wird, oder aber die Länge der besten Route, die innerhalb einer vorgegebenen Zeit gefunden wird.

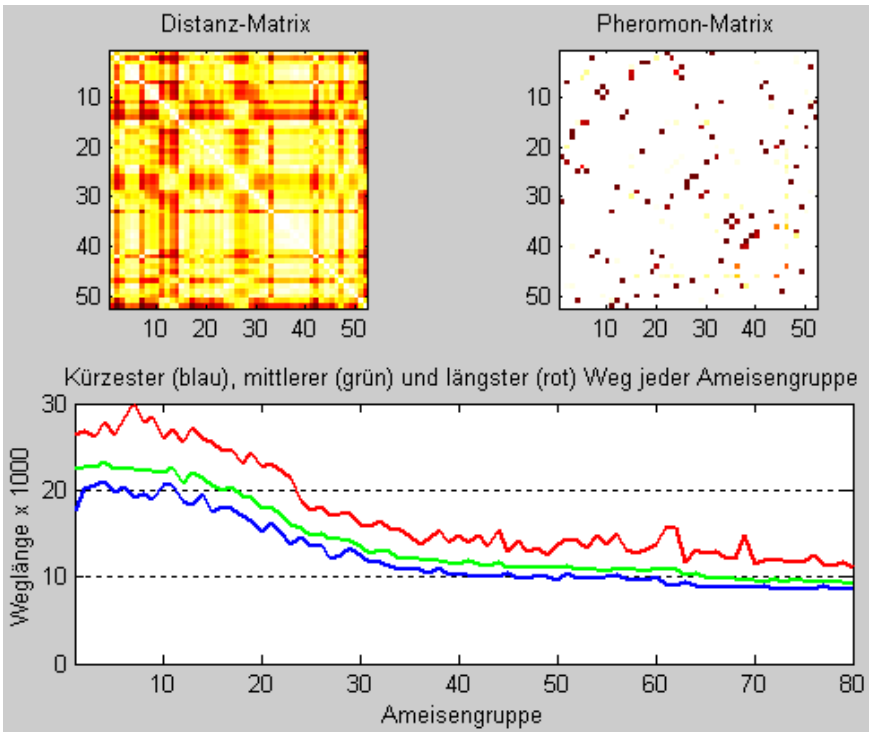

Abbildung 16: TSP – Entwicklung für reale Daten – Optimum: 7542

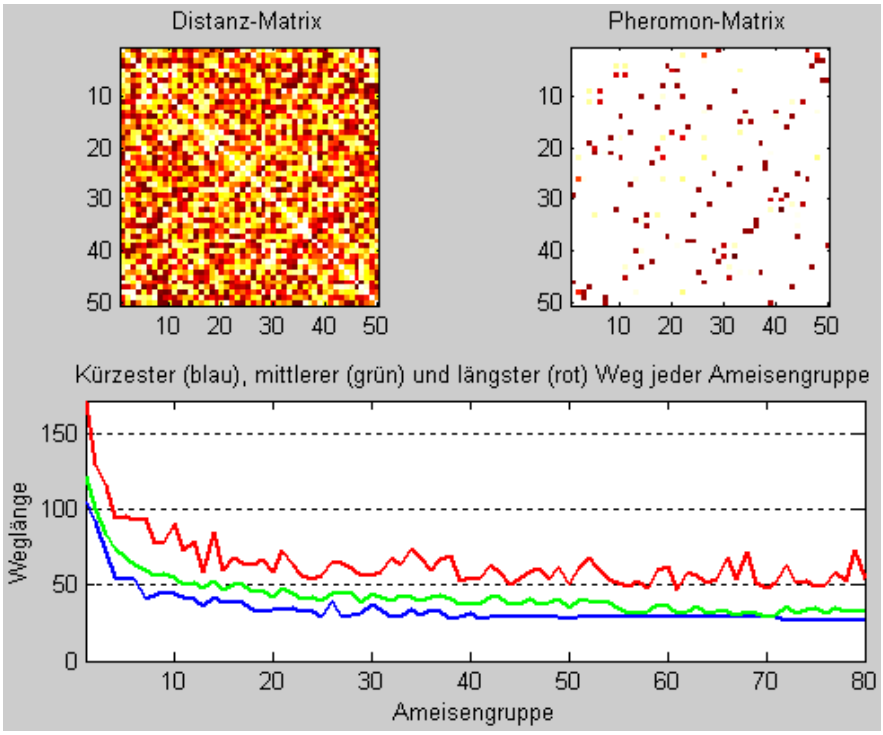

Abbildung 17: TSP – Entwicklung für zufällige Daten

Die wichtigsten Parameter sind:

- Aufteilung der Ameisen in Gruppen: Größe und Anzahl der Gruppen
- Verhältnis des Einflusses der Weglängen im Vergleich zu den Pheromonen
- Verstärkungsexponent für die Zielwertfunktion zur Verminderung des Zufalls
- Verwitterungsfaktor

4.4 Störungen

Der am stärksten mit Pheromonen markierte Weg sei ein instabiler Engpass, dem die Ameisen bei dessen Blockierung (vielleicht durch eine Pfütze) ausweichen müssen, so dass sich die "Kosten" für ihren Rundweg, also die benötigte Zeit, erhöhen – oder auch nicht, falls sie hierdurch einen noch besseren Weg finden. Wird eine solche Störung eingebracht, so ist gut zu beobachten, dass die Markierung an dieser Stelle innerhalb kurzer Zeit verblasst und ein neuer Weg gefunden wird – sofern dieser Weg nicht bereits sehr billig (kurz) war und sich auch nach Einbringen der Störung in Verbindung mit der Gesamtroute weiterhin lohnt.
Dass die Ameisen durch eine Störung motiviert werden, eine gänzlich neue Route zu erkunden, die bis dahin weitab lag, kann nur selten beobachtet werden.

Im Ergebnis bestätigt sich das Prinzip des Ameisenalgorithmus als robuste und effiziente Methode zur Optimierung von Problemen ähnlich dem Handlungsreisendenproblem.

4.5 Routing

Neben der Übertragung auf mit dem Handlungsreisendenproblem verwandte Probleme wie die Tourenplanung für Fahrdienste, Personalplanung, Maschinenbelegung in Fertigungsstraßen oder das Design von Mikrochips, wo zum Teil schon Praxisreife erlangt ist [Wiki03], stellt das Routing in Netzwerken eine vielversprechende Einsatzmöglichkeit dar. Es geht darum, einen möglichst effizienten Weg für einen Nachrichtenstrom durch ein teilvermaschtes Datennetz zu finden, wobei in der Regel über die Weiterleitung (Forwarding) an jedem Knoten neu entschieden wird.

Während die aufgezählten Umgebungen meist stabil sind, kann die Familie der Ameisenalgorithmen ihre Stärken insbesondere bei hoch dynamischen Problemstellungen ausspielen, denn Ameisen erlauben im Gegensatz zu anderen Heuristiken flexible Reaktionen auf Änderungen ihrer Umweltfaktoren. Dabei kann es sich nicht nur um Störungen, sondern durchaus um erwünschte Änderungen handeln, etwa die Bewegung der Stationen in drahtlosen Netzwerken.

Obwohl in den meisten großen Netzwerken noch immer die traditionellen, statischen Routing-Tabellen verwendet werden, gibt es zu Forschungszwecken bereits zahlreiche erfolgreiche Implementierungen von Ameisenalgorithmen, beispielsweise an der Universität Aachen [GS06].

Literatur

[ACO] Ant Colony Optimization (ACO)
`http://www.aco-metaheuristic.org/`

[Boy05] Nils Boysen: Immer der Nase nach – Tourenplanung nach dem Vorbild der Ameisen. c't 2005-05: 204–207.
`http://www.heise.de/ct/05/05/links/204.shtml`

[Dor] Marco Dorigo, Artificial Intelligence research laboratory of the Université Libre de Bruxelles.
`http://iridia.ulb.ac.be/~mdorigo/HomePageDorigo/`

[Gle03] Clemens Gleich: Scheinintelligenz – Der Quick-and-Dirty-Ansatz zu rationalem Verhalten. c't 2003-08: 168–173.

[GS06] Mesut Günes, Otto Spaniol: Was können Informatiker von Ameisen lernen? Universität Aachen, 2006.
`http://nets.rwth-aachen.de/ameisen/`

[Lov02] Jörn Loviscach: Die Würfel Gottes – Stephen Wolframs 'neue Art von Wissenschaft'. c't 2002-13: 234–236.

[Sti03] Dr. Wolfgang Stieler: Körper und Geist – Visionen und Realität der künstlichen Intelligenz. c't 2003-02: 64–68.

[SV01] Michael Sonnenschein, Ute Vogel: Asymmetric Cellular Automata for the Modelling of Ecological Systems; Sustainability in the Information Society - 15th International Symposium Informatics for Environmental Protection, pages 631–636, 2001.

[TSPLIB] TSPLIB – a library of sample instances for the TSP (and related problems) from various sources and of various types.
`http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/`

[WWW06] Matthias Wittlinger, Rüdiger Wehner, Harald Wolf: The Ant Odometer: Stepping on Stilts and Stumps; Science Vol. 312. no. 5782, pp. 1965–1967, 2006.
`http://www.sciencemag.org/cgi/content/full/312/5782/1965/DC1`

[Wiki01] Wikipedia: Ameisenstraße
`http://de.wikipedia.org/wiki/Ameisenstraße`

[Wiki02] Wikipedia: Pheromon
`http://de.wikipedia.org/wiki/Pheromon`

[Wiki03] Wikipedia: Problem des Handlungsreisenden
`http://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden`

[Wiki04] Wikipedia: Software-Agent
`http://de.wikipedia.org/wiki/Software-Agent`

(Alle Internet-Adressen wurden am 28.09.2007 auf Gültigkeit geprüft.)