Ameisenalgorithmen

Nils Boysen

Universität Hamburg
Institut für Industriebetriebslehre und Organisation
Von-Melle-Park 5
20146 Hamburg
E-Mail: boysen@econ.uni-hamburg.de

Überblick

Ameisenalgorithmen halten seit ihrer Vorstellung im Jahre 1991 zunehmend Einzug ins Schrifttum des Operations Research. Dies liegt nicht allein an der originellen Adaption ihres natürlichen Vorbilds, sondern vor allem an den viel versprechenden Ergebnissen, welche die Meta-Heuristik in unterschiedlichen Problemstellungen der kombinatorischen Optimierung liefert. In diesem Artikel soll die Übertragung der Futtersuche der Ameisen auf die Optimierung dargestellt werden. Beginnend mit dem biologischen Vorbild in Abschnitt 1 wird anschließend in Abschnitt 2 eine formale Beschreibung der Ameisenalgorithmen vorgenommen, um im abschließenden Abschnitt 3 Hinweise für die Anpassung an konkrete Optimierungsprobleme zu liefern.

1. Die Idee

Ameisenalgorithmen beziehen sowohl ihren Namen als auch die sich dahinter verbergende Idee der Lösungssuche aus ihrer Analogie zur Natur. So ließ sich der italienische Mathematiker Marco Dorigo von der Futtersuche der Ameisen inspirieren, als er 1991 den Ablauf in Ameisenkolonien auf die Lösungssuche in kombinatorischen Optimierungsproblemen übertrug, vgl. Schmundt (2000).[1]

Ameisen orientieren sich bei ihrer Futtersuche mittels eines chemischen Sekrets namens Pheromon, welches sie während ihrer Fortbewegung laufend aus einer Drüse am hinteren Teil ihres Körpers auf den Boden absondern, vgl. Bonabeau und Meyer (2001). Nachfolgende Ameisen wählen, vor die Entscheidung gestellt, in welcher Richtung sie ihren Weg fortsetzen sollen, mit einer höheren Wahrscheinlichkeit den Weg, auf dem bereits mehr Pheromon hinterlassen wurde, vgl. Bonabeau et al. (2000). Das Pheromon nimmt damit die Rolle einer Art kollektiven Gedächtnisses der Kolonie ein, welches die vergangenen Wegentscheidungen speichert.

Beobachtungen aus der Natur zeigen, dass Ameisen ihre Straßen meist auf direktem Weg zwischen Nest und Futterquelle errichten. Doch wie ermöglicht nun das Pheromon der Kolonie als Ganzem, einen kürzesten Weg zu finden, wozu eine einzelne Ameise mittels ihrer individuellen Fähigkeiten allein nur zufällig in der Lage wäre. Ameisen machen es sich zu

[1] Unter den ersten Veröffentlichungen finden sich: Colorni et al. (1991), Dorigo et al. (1991) und Dorigo (1992).

Nutze, dass kürzere Wege schneller durchlaufen werden können und dementsprechend ihre Markierung schneller erhalten. Pro Zeiteinheit können somit mehr Ameisen den kürzeren Weg durchlaufen als einen längeren. Ein Mehr an Ameisen bedeutet aber auch, dass Nachfolger vor die Wahl gestellt mit höherer Wahrscheinlichkeit den stärker belaufenen und markierten Weg wählen. Dadurch verstärkt sich die Anziehungskraft des kürzeren Weges im Zeitablauf immer weiter, bis eine Ameisenstraße auf der annähernd direkten Verbindung zwischen Nest und Nahrung entstanden ist.

Diesen Ablauf soll die folgende schematische Wegsuche aus Abbildung 1 verdeutlichen, deren Ablauf von Biologen mit Experimenten ähnlichen Aufbaus nachgewiesen wurde, siehe hierzu Pasteels et al. (1987), Goss et al. (1989), Deneubourg et al. (1990).

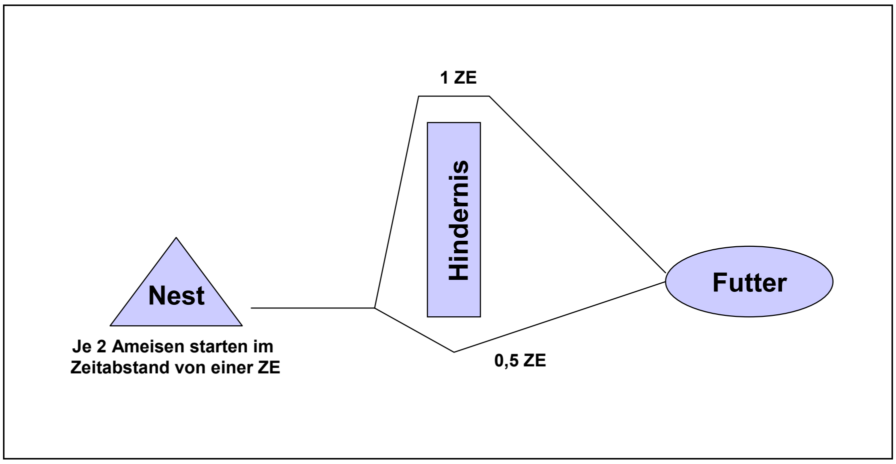

Abb. 1: Schematischer Ablauf der Futtersuche von Ameisen

Angenommen pro Zeiteinheit starten jeweils zwei Ameisen vom Nest aus, um Nahrung zu sammeln. Zwischen Ameisenhaufen und Futterquelle befindet sich ein Hindernis, dessen Umlaufen die Ameisen auf dem längeren Weg eine Zeiteinheit kostet und auf dem kürzeren lediglich eine halbe. Da die Ameisen die Güte der Wege von ihrer Position aus nicht überschauen können und zu Beginn noch keine Ameise die Wege mit Pheromon gekennzeichnet hat, sei unterstellt, dass ihre Zufallsauswahl die eine Ameise den längeren und die andere den kürzeren Weg beschreiten lässt. Ist eine Zeiteinheit vergangen und zwei weitere Ameisen sind vor die Wegentscheidung gestellt, so hat die Ameise, welche den längeren Weg gewählt hat, gerade die Futterquelle erreicht, mithin ihren Weg mit einer Pheromoneinheit markiert. Die andere Ameise hat aber in der einen Zeiteinheit schon den Weg zurück ins Nest zurückgelegt und somit den Weg doppelt markiert. Die nachfolgenden Ameisen werden nun mit einer höheren Wahrscheinlichkeit den kürzeren Weg wählen, da die größere Pheromonmenge ihre Auswahlentscheidung in diese Richtung beeinflusst. So bildet sich schon nach kurzer Zeit eine stärkere Präferenz für den kürzeren Weg heraus, die weiter verstärkt wird, je mehr Ameisen erfolgreich den kürzeren Weg beschritten haben.

Nun lässt sowohl die Natur als auch ein kombinatorisches Optimierungsproblem den einzelnen Ameisen wesentlich mehr Freiheitsgrade bei ihrer Wegentscheidung als in dem Beispiel, so dass die Konvergenz zum kürzesten Weg bzw. der besten Lösung mehr Ameisen und eine längere Zeit beansprucht; der Mechanismus dahinter bleibt aber der beschriebene.

2. Übertragung des natürlichen Vorbilds auf einen Algorithmus

Artifizielle Ameisen bilden das Verhalten ihrer natürlichen Vorbilder nach, indem sie die Variablen einer gesuchten Lösung sukzessive analog einer Wegsuche festlegen. Dabei machen sie die Wahlentscheidung über den Wert der aktuell betrachteten Lösungsvariablen abhängig von der Höhe der Pheromonvariablen, welche die Güte der vorangegangenen Wahlentscheidungen zwischen der Lösungsvariablen und ihrer möglichen Ausprägungen an dieser Stelle des Festlegungsprozesses kennzeichnet.

Die künstlichen Ameisen orientieren sich aber nicht allein am Pheromon. Zusätzlich wird ihnen eine Art Sehfähigkeit attestiert. Bei ihrer Wegentscheidung ziehen sie nicht nur die Pheromonmenge hinzu, sondern orientieren sich zusätzlich an einer heuristischen Information, einer Prioritätsregel. Übertragen auf die Wegsuche betont eine Prioritätsregel etwa die nächstgelegene Weggabelung (Kante in einem Netzwerk), vgl. Maniezzo und Carbonaro (1999).

Nach der Fixierung einer Variablen kennzeichnen auch die artifiziellen Ameisen ihre getroffene Wegentscheidung für die Nachfolger mit Pheromon. Dies geschieht, indem in einer Matrix eine Fließkommazahl, die den Pheromonwert an der Verbindung zwischen der festgelegten Variablen und ihrer gewählten Ausprägung aus dem Wertebereich repräsentiert, proportional zur der Güte der Auswahl manipuliert wird.

Entsprechend dieser Skizzierung der Übertragung des natürlichen Vorbilds auf einen Algorithmus gilt es, folgende drei Hauptelemente näher zu beschreiben:

a) Den Ablauf der sukzessiven Festlegung der Variablen analog zur Wegsuche der Ameisen,
b) die Regel nach der die Fixierung einer Variablen aus ihrem Wertebereich entsprechend einer Wegentscheidung erfolgt, und
c) die Markierung der erzeugten Lösungen analog der Pheromonablage.

Formal lässt sich die **Wegsuche** folgendermaßen darstellen:[2] Gesucht wird eine Lösung x mit x_i ($i = 1,\ldots,n$) Variablen. Da Ameisenalgorithmen für kombinatorische Optimierungsprobleme eingesetzt werden, kann jede dieser Variablen eine endliche Anzahl an Ausprägungen aus ihrem Wertebereich annehmen, dargestellt durch die Menge J^i. Die Wegsuche der artifiziellen Ameisen erfolgt nun durch den Lösungsraum, indem sukzessive die Variablen x_i aus ihren Wertebereich J^i festgelegt werden. Zur Fixierung einer Variablen analog zu einer einzelnen **Wegentscheidung** wird jeder möglichen Ausprägung der Variablen eine Auswahlwahrscheinlichkeit $P(x_{ij})$ zugeordnet, die sich aus dem Pheromonwert und der Prioritätsregel berechnet, siehe Formel (1.1). Anschließend sorgt eine Monte-Carlo-Auswahl für eine zufallsgestützte Wegauswahl proportional zur Auswahlwahrscheinlichkeit.

$$P(x_{ij}) = \frac{\tau_{ij}^{\ \alpha} \cdot \nu_{ij}^{\ \beta}}{\sum_{k \in J^i} \tau_{ik}^{\ \alpha} \cdot \nu_{ik}^{\ \beta}} \qquad \forall i = 1,\ldots,n;\ \ j \in J^i \tag{1.1}$$

Für die Festlegung einer Variablen x_i aus dem zulässigen Wertebereich $j \in J^i$ werden das Pheromon τ_{ij} und der Wert einer Prioritätsregel ν_{ij} der betrachten Wegentscheidung mit den Ausprägungen aller möglichen Alternativen ins Verhältnis gesetzt. Dabei sind α und β Gewichtungsfaktoren, welche den Einfluss von Pheromon bzw. heuristischer Information auf die Wegentscheidung steuern können. Anhand der ermittelten Auswahlwahrscheinlichkeiten kann anschließend eine Monte-Carlo-Auswahl über die Festlegung der betrachten Variablen entscheiden; siehe Abb. 2.

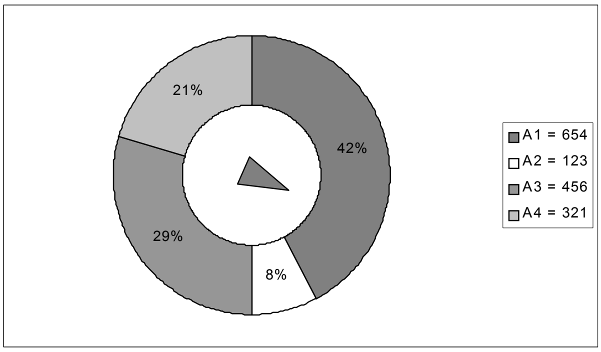

Abb. 2: Schematische Darstellung der Monte-Carlo Auswahl durch den Ameisenalgorithmus

Zahlreiche Erweiterungen hat die Grundform der Auswahlentscheidung seit ihrem Bestehen erfahren. Eine der wichtigsten Erweiterungen, die auch ACO[3] genannt wird, vgl. Dorigo et al. (1999), wendet im Wechsel neben der Monte-Carlo-Auswahl auch eine Regel an, welche

---

[2] Eine weiterführende formale Betrachtung losgelöst von einem konkreten Optimierungsproblem findet sich bei Birattari et al. (2002).

[3] Die Abkürzung ACO steht für Ant Colony Optimization.

immer diejenige Auswahlentscheidung mit dem maximalen Produkt aus Pheromon und heuristischer Information trifft. Die Steuerung, welche der beiden Auswahlregeln angewendet wird erfolgt über den Vergleich einer gezogenen Zufallszahl $0 \leq rnd \leq 1$ mit einem vorzugebendem Schwellenwert $0 \leq Q \leq 1$, wie es Formel (1.2) nahe legt.

$$P(x_{ij}) = \begin{cases} \textit{wenn } rnd > Q: & \dfrac{\tau_{ij}^{\ \alpha} \cdot \nu_{ij}^{\ \beta}}{\sum_{k \in J^i} \tau_{ik}^{\ \alpha} \cdot \nu_{ik}^{\ \beta}} \quad \forall\, i=1,\ldots,n;\ j \in J^i \\ \textit{wenn } rnd \leq Q: & \begin{cases} 1 & \forall\, i=1,\ldots,n;\ j = \arg\max_{k \in J^i} [\, \tau_{ik}^{\ \alpha} \cdot \nu_{ik}^{\ \beta}\,] \\ 0 & \textit{sonst} \end{cases} \end{cases} \tag{1.2}$$

Durch diese modifizierte Auswahlregel kann die Suche in der Nachbarschaft von der bis dato besten gefundenen Lösung verstärkt werden.

Neben der Wegsuche ist die Rückmeldung des Erfolgs via Pheromon das wichtigste Element eines Ameisenalgorithmus. Die **Pheromonablage** erfolgt allerdings etwas anders als bei ihren lebendigen Vorbildern. Während dort kürzere Wege in geringerer Zeit zurückgelegt werden und dementsprechend ihre Kennzeichnung schneller erhalten als längere Wege, ist dies am Computer durch dessen diskrete Natur schwierig abzubilden. Daher erfolgt die Ablage des künstlichen Pheromons dergestalt, dass zunächst alle Ameisen einer Iteration ihren Weg bestimmen. Anschließend wird für jede Ameise die Güte der gefundenen Lösung bestimmt und erst jetzt ex post jede in der Gesamtlösung enthaltene Teilentscheidung proportional zur Lösungsgüte mit Pheromon gekennzeichnet. Sind alle n Variablen festgelegt, so liegt eine Lösung x für das Ausgangsproblem vor, für die wiederum der Zielfunktionswert $F(x)$ errechnet werden kann. Alle Elemente (i,j) des Lösungsraumes, die Teil der Lösung x sind, werden nun in der Pheromonmatrix t entsprechend des Zielfunktionswertes der Lösung gekennzeichnet, siehe Formel (1.3) für ein Minimierungsproblem.

$$\tau_{ij} = \underbrace{\tau_{ij}(1-\rho)}_{\textbf{Verwitterung}} + \underbrace{\rho \cdot \begin{cases} \dfrac{1}{F(x)} & \forall (i,j) \in x \\ 0 & \forall (i,j) \notin x \end{cases}}_{\textbf{Pheromonablage}} \tag{1.3}$$

Die Betonung guter Lösungen kann weiter erhöht werden, indem nicht alle Ameisen für die Pheromonablage berechtigt werden. Abweichend von der Natur kann etwa nur diejenige Ameise mit dem besten Zielfunktionswert (x^{max}) für die Pheromonablage ausgewählt werden, vgl. Dorigo et al. (1999). Zusätzlich ist in Formel (1.3) ein ebenfalls an die Natur angelehnter Verwitterungsfaktor 0<r<1 integriert. Wie die Witterungseinflüsse auf eine reale

Pheromonspur einwirken, schwächt dieser Faktor auch die virtuelle Spur vergangener Iterationen. Damit wird der Einfluss zurückliegender Lösungen gegenüber jüngst gefundenen abgeschwächt. Da jüngst erzeugte Lösungen auf mehr Wegerfahrung ihrer Vorgänger zurückgreifen, wird so das Suchen einer guten Lösung aus der kollektiven Erfahrung heraus gegenüber eher zufällig gefundenen guten Lösungen weiter verstärkt.

Alle die von der Natur übernommenen Mechanismen können nicht darüber hinwegtäuschen, dass es sich bei den Ameisenalgorithmen lediglich um eine Heuristik handelt. Genauso wie Ameisen in der Natur zum Teil ihre Straßen nicht immer auf der kürzesten Verbindung zwischen Nest und Futterquelle errichten, kann es auch den virtuellen Ameisen ergehen.

Die größte „Gefahr" droht dadurch, dass im Laufe des Verfahrensablaufs frühzeitig ein einzelner Weg durch das auf ihm abgelegte Pheromon eine so starke Anziehungskraft ausübt, dass auch die proportionale Zufallsauswahl keine Variationen in der Wegsuche mehr gewährleisten kann. Genauso wie ein Genetischer Algorithmus durch eine Mutation von Zeit zu Zeit eine Erneuerung der Varietät im Genpool benötigt,[4] muss eine Konvergenz des Ameisenalgorithmus hin zu einer suboptimalen Lösung verhindert werden. Aus diesem Grund wird häufig entgegen dem Vorbild der Natur ein Ausgleichsmechanismus, der sowohl ein Zuviel als auch ein Zuwenig an Pheromon an einzelnen Stellen im Lösungsraum verhindern will, verwendet, siehe Dorigo et al. (1999). Je nach Ausprägung des Steuerungsparameters $0<j<1$ wird das Pheromon, immer wenn sie Teil einer Lösung wird, wieder in Richtung der Startbelegung t_0 angepasst (1.4).

$$\tau_{ij} = \tau_{ij}(1-\varphi) + \varphi \cdot \tau_0 \qquad \forall (i,j) \in x \tag{1.4}$$

Diese Startbelegung mit Pheromon lässt sich anhand des Zielfunktionswertes einer heuristischen Lösung $F(x^{heu})$ mittels Formel (1.5) wiederum für ein Minimierungsproblem bestimmen; vgl. etwa Bullnheimer et al. (1999).

$$\tau_0 = \frac{1}{n \cdot F(x^{heu})} \qquad \forall i = 1,\ldots,n; \;\; j \in J^i \tag{1.5}$$

Ein praktischer Nebeneffekt der Ameisenalgorithmen ist es, dass keinesfalls eigens für die Bestimmung von $F(x^{heu})$ eine weitere Heuristik programmiert werden muss. Vernachlässigt man das Pheromon und die proportionale Zufallsauswahl, so erfolgt die Wegsuche im Lösungsraum myopisch, indem die Variable x_i jeweils auf das durch die Prioritätsregel nahe gelegte beste j ($j = \arg\max_{k \in J^i}[v_{ik}]$) fixiert wird. Dieses Vorgehen wird von sog. Nearest-

[4] Zur Konvergenz Genetischer Algorithmen hin zu einem Genotyp siehe Mühlenbein (1997).

Neighbor-Heuristiken[5], vgl. Domschke und Drexl (1998, S. 120f.), verfolgt. Somit stellt die Nearest-Neighbor-Heuristik eine Sonderform des Ameisenalgorithmus mit einer Ameise, Q=1 in (1.2) und r_0œÑ$^+$. Nach dem Durchlaufen einer Iteration des Ameisenalgorithmus mit dieser Parameterkonstellation kann dieser Zielfunktionswert in der Formel (1.5) zur Berechnung der Startbelegung mit Pheromon verwendet werden.

3. Anpassung von Ameisenalgorithmen an konkrete Optimierungsprobleme

Um die Idee der Ameisenalgorithmen zum Lösen von konkreten Optimierungsproblemen nutzbar zu machen, muss ein Computerprogramm erstellt werden. Dieses besteht primär aus zwei Steuerungsroutinen. Die erste veranlasst auf der Ebene der Ameisenkolonie, dass pro Iteration jede Ameise einen Weg durch den Lösungsraum konstruiert. Anschließend wird die jeweils beste Lösung bestimmt und die Pheromonablage ausgelöst. Dieser Ablauf ist im Pseudocode der Abbildung 3 dargestellt.

```
F(x^heu):= Ergebnis von Ameisenalgorithmus mit Q:=1,
|A|:=1, r0=1, maximale Iterationszahl:=1
Pheromon initialisieren nach (1.5)
while i ≤ maximale Iterationszahl
        for each a œ A
                a.Wegsuche; (Abb. 4)
        Pheromon abgeben nach (1.3)
        i:=i+1
beste Lösung aller Iterationen und Ameisen zurückgeben
```

Abb. 3: Pseudocode für die Koloniesteuerung

Als zweite elementare Prozedur fungiert die Wegsuche einer jeden Ameise *a* aus der Menge aller Ameisen der Kolonie *A*. Die Konstruktion einer Lösung durch eine einzelne Ameise verdeutlicht der Pseudocode aus Abbildung 4.

[5] Teilweise werden solche Heuristiken auch mit Best-Fit bezeichnet, vgl. Domschke und Drexl (1998, S. 121).

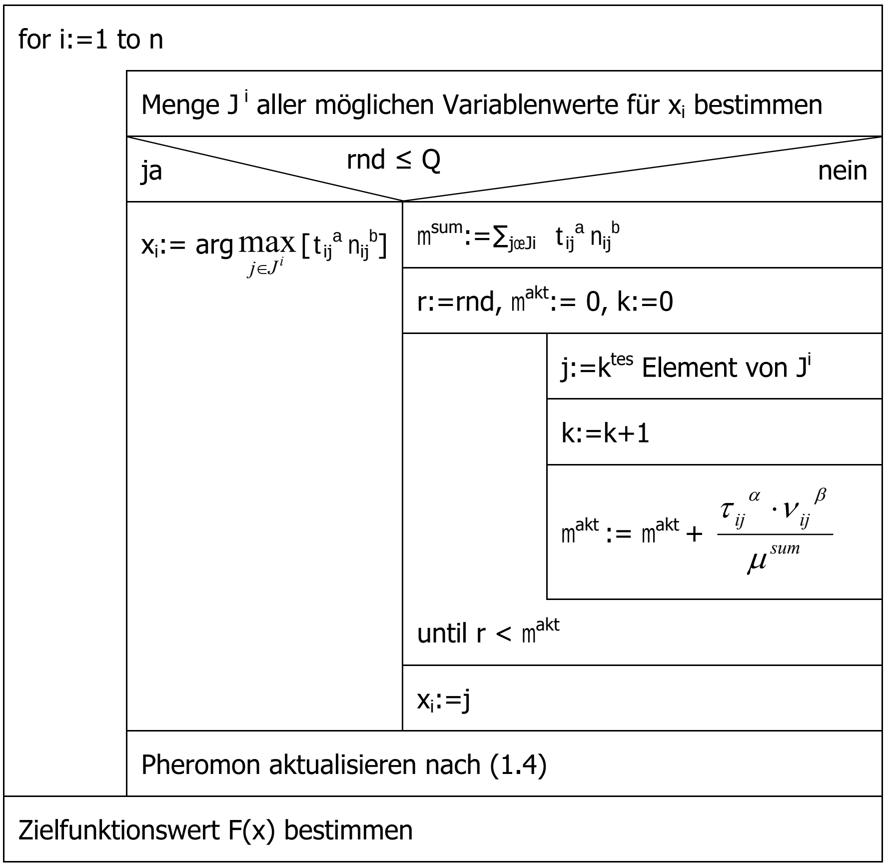

Abb. 4: Pseudocode für die Wegsuche einer einzelnen Ameise

Mit dem so beschriebenen Gerüst kann die Metaheuristik auf konkrete Probleminstanzen übertragen werden. Dabei müssen vor allem drei Teile eines Ameisenalgorithmus problemspezifisch konkretisiert werden.

Zunächst gilt es eine geeignete **Lösungsrepräsentation** für die Menge der Variablen abzuleiten, welche der Problemstellung zugrunde liegen. Gilt es etwa beim sog. Travelling Salesman Problem (TSP) die kürzeste Route eines Handlungsreisenden durch eine vorgegeben Anzahl von *n* Städten zu bestimmen, so erfolgt die Lösungsrepräsentation durch einen Reihenfolgevektor p_i mit $i = 1,\ldots,n$.[6] Die Wegsuche einer Ameise kann dann wie bei einer realen Wegsuche von Ort zu Ort fortschreiten, bis der Reihenfolgevektor gefüllt ist. Zur Auswahl der nächsten Stadt an der Reihenfolgeposition *i* sind nur noch diejenigen Städte erlaubt, die noch nicht besucht wurden. Dementsprechend ist bei der **Bestimmung der Menge der erlaubten Variablenausprägungen** J^i, dieser Sachverhalt zu beachten. Zur Auswahl des nächsten Ortes aus der Menge J^i wird dann der Pheromonwert aus (n,n) Matrix bestimmt. Diese Matrix hält fest, wie erfolgreich es in der Vergangenheit war die Kante (i,j) in eine Tour aufzunehmen. Zusätzliche zum Pheromon wird ein heuristischer Wert

[6] Zur Anwendung von Ameisenalgorithmen auf das TSP siehe Dorigo et al. (1991), Dorigo et al. (1996), Dorigo und Gambardella (1997), Stützle und Dorigo (1999b), Bullnheimer et al. (1997).

herangezogen, der nur problemspezifisch festgelegt werden kann. Beim TSP ist die **myopische Information** zur Wegsuche der Kehrwert der Entfernung vom aktuellen Standort zu dem betrachteten nachfolgenden Ort.

Neben Reihenfolgeproblemen wie dem TSP stellen Zuordnungsprobleme eine weitere grundlegende Problemklasse kombinatorischer Optimierungsprobleme dar, vgl. Ellinger (1990, S. 8), Zimmermann (1995, S. 147).[7] Beim klassischen Zuordnungsproblem, siehe etwa Hansmann (2001, S. 219), geht es darum m Arbeitern m Arbeitsplätzen zuzuordnen. Dabei hat jeder Arbeiter eine unterschiedliche Eignung e_{ij} für jeden Arbeitsplatz. Insgesamt gilt es, die Summe der durch eine Zuordnung realisierten Eignungen zu maximieren. Die Wegsuche erfolgt bei dieser Problemstellung, indem sukzessive einem Arbeiter einer der Arbeitsplätze zugeordnet wird. Dementsprechend erfolgt die **Lösungsrepräsentation** in einem Vektor π_i mit $i = 1,\ldots,m$, der für jeden Arbeiter i die Nummer des ihm zugeordneten Arbeitsplatzes $j \in \{1,\ldots,m\}$ aufnimmt. Dabei ist leicht ersichtlich das im aktuellen Zuordnungsschritt nur diejenigen Arbeitsplätze bei der **Bestimmung der Menge der erlaubten Variablenausprägungen** in der Menge J^i verbleiben dürfen, die noch nicht in einem der vorigen Zuordnungsentscheidungen vergeben wurden. Die Pheromonmenge zwischen der Variablen i, die einen Arbeiter repräsentiert, und den erlaubten Ausprägungen der Variablen, der Nummern der Arbeitsplätze, wird aus einer (m,m) Matrix gelesen. Zusammen mit der **myopischen Information**, die einfach die Eignung e_{ij} für den untersuchten Arbeitsplatz widerspiegelt, kann durch die Wegsuche jedem Arbeiter ein Arbeitsplatz zugeordnet werden.

Somit kann durch die problemspezifische Ausgestaltung der drei Elemente:

- Lösungsrepräsentation
- Bestimmung der Menge der erlaubten Variablenausprägungen
- myopischen Information

zusammen mit den problemunabhängigen Steuerungsroutinen eine leistungsstarke Metaheuristik erzeugt werden, wie zahlreiche Implementierungen von Ameisenalgorithmen für unterschiedlichste Optimierungsprobleme nahe legen, vgl. Tabelle 1.

[7] Aus der Familie der Zuordnungsprobleme wurde z.B. das sog. Quadratic Assignment Problem (QAP) mit Ameisenalgorithmen gelöst, vgl. Stützle und Hoss (1998), Stützle und Dorigo (1999a), Gambardella et al. (1999b).

Problemstellung	Fundort
Travelling Salesman Problem	Dorigo et al. (1991), Dorigo et al. (1996), Dorigo und Gambardella (1997), Stützle und Dorigo (1999b), Bullnheimer et al.(1997)
Vehicle Routing Problem	Bullnheimer et al. (1999), Gambardella et al. (1999a)
Quadratic Assignment Problem	Stützle und Hoss (1998), Stützle und Dorigo (1999a), Gambardella et al. (1999b)
JIT Sequencing Problem	McMullen (2001)
Graph Coloring	Costa und Hertz (1997)
Shortest Common Supersequence Problem	Michel und Middendorf (1999)
Constraint Satisfaction Problem	Roli et al. (2001)
Sequential Ordering Problem	Gambardella und Dorigo (2000)
Routing in Telekommunikationsnetzwerken	Di Caro und Dorigo (1997), Schoonderwoerd et al. (1997a), Schoonderwoerd et al. (1997b), Di Caro und Dorigo (1998)
Project Scheduling Problem	Merkle et al. (2000), Boysen et al. (2002)
Physikalische Speicherung von Daten im Data Warehouse	Maniezzo et al. (2001)
Graph Partioning	Kuntz und Snyers (1994)
Scheduling	Colorni et al. (1994), Stützle (1998), Merkle und Middendorf (2000), Stützle et al. (2000), Gagne et al. (2001), Gagne et al. (2002), T'kindt et al. (2002)
Assembly Line Balancing	Bautista und Pereira (2002)
Zuweisung von Radiofrequenzen	Maniezzo und Carbonaro (2000)
Portfolio-Selection	Maringer (2002)
Anordnung der Tasten auf einer Tastatur	Eggers et al. (2003)

Tab. 1: Überblick über die mit Ameisenalgorithmen gelösten Problemstellungen

Literaturverzeichnis

Bautista und Pereira (2002)	Bautista, J.; Pereira, J.: Ant Algorithms for Assembly Line Balancing, in: Dorigo, M. et al. (Hrsg.): ANTS 2002, LNCS 2463, Berlin 2002, S. 65-75
Birattari et al. (2002)	Birattari, M.; Di Caro, G.; Dorigo, M.: Toward the Formal Foundation on Ant Programming, in: Dorigo, M. et al. (Hrsg.): ANTS 2002 LNCS 2463, Berlin 2002, S. 188-201
Bonabeau und Meyer (2001)	Bonabeau, E.; Meyer, C.: Schwarm-Intelligenz: Unternehmen lernen von Bienen und Ameisen, in: Harvard Business manager 2001 Heft 6, S. 38-48
Bonabeau et al. (2000)	Bonabeau, E.; Dorigo, M.; Theraulaz, G.: Inspiration for optimization from social insect behaviour, in: Nature 406 2000, S. 39-42
Boysen et al. (2002)	Boysen, O.; Juretzka, J.; Kimms, A.: Ameisen-Systeme zur kapitalwertmaximierenden Projektplanung, in:, Zeitschrift für Planung, 11 2002, S. 289-305
Bullnheimer et al (1997)	Bullnheimer, B.; Hartl, R. F.; Strauß, C.: A new rank based version of th Ant System – a computational study, working paper No. 1 April 1997, Universität Wien
Bullnheimer et al (1999)	Bullnheimer, B.; Hartl, R. F.; Strauss, C.: Applying the Ant System to the Vehicle Routing Problem. in: Voss S., Martello S., Osman I.H., Roucairol C. (Hrsg..): Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Boston 1999
Colorni et al. (1991)	Colorni, A.; Dorigo, M.; Maniezzo, V.: Distributed optimization by ant colonies, in: Proceedimgs of ECAL'91, European Conference on Artifical Life, Berlin 1991
Colorni et al. (1994)	Colorni, A.; Dorigo, M.; Maniezzo, V.; Turbin, M.: Ant system for Job-shop Scheduling, in: Belgian Journal of Operations Research, Statistics and Computer Science 34 1994, S. 39-53
Costa und Hertz (1997).	Costa D.; Hertz, A.: Ants Can Colour Graphs, in: Journal of the Operational Research Society 48 1997, S. 295-305
Deneubourg et al. (1990)	Deneubourg, J.-L.; Aron, S.; Goss, S.; Pasteels, J.-M.: The self-organizing exploratory pattern of the argentine ant, in: Journal of Insect Behaviour 3 1990, S. 159-168
Di Caro und Dorigo (1997)	Di Caro, G.; Dorigo, M.: AntNet: A mobile agents approach to adaptive routing, working paper IRIDIA/97-12, Université Libre de Bruxelles, Belgium.
Di Caro und Dorigo (1998)	Di Caro, G.; Dorigo, M.: AntNet: Distributed Stigmergetic Control for Communications Networks, in: Journal of Artificial Intelligence Research 9 1998, S. 317-365
Domschke und Drexl (1998)	Domschke, W.; Drexl, A. :Einführung in Operations Research, 4. Aufl. Berlin 1998
Dorigo (1992)	Dorigo, M.: Optimization, learning and natural algorithms, (Diss.) Mailand 1992
Dorigo und Gambardella (1997)	Dorigo, M; Gambardella, L. M.: Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem, in: IEEE Transactions on Evolutionary Computation, 1 1997, S. 53-66
Dorigo et al. (1991)	Dorigo, M.; Maniezzo, V.; Colorni, A.: Ant System: An autocatalytic optimizing process, working paper No. 91-016 Revised, Politecnico di Milano, Italy
Dorigo et al. (1996)	Dorigo, M.; Maniezzo, V.; Colorni, A.: The Ant System: Optimization by a colony of cooperating agents, in: IEEE Transactions on Systems, Man and Cybernetics – Part B 26 1996, S. 29-41

Dorigo et al. (1999)	Dorigo, M.; Di Caro, G.; Gambardella, L. M.: Ant algorithms for discrete optimization, in: Artifical Life 5(2) 1999, S. 137-172
Eggers et al. (2003)	Eggers, E.; Feillet, D.; Kehl, S.; Wagner, M. O.; Yannou, B.: Optimizing of the keyboard arrengement problem using Ant Colony algorithm, in: European Journal of Operational Research 148 2003, S. 672-686
Ellinger (1990)	Ellinger, T.: Operations Research, 3. Aufl. Berlin 1990
Gagne et al. (2001)	Gagne, C.; Price, W. L.; Gravel, M.: Scheduling a single machine with sequence dependent setup time using Ant Colony Optimization, working paper 2001-003, Faculte des science de l'administration Universite Laval Quebec 2001
Gagne et al. (2002)	Gagne, C.; Price, W. L.; Gravel, M.: Comparing an ACO algorithm with other heuristics for the single machine scheduling problem with sequence-dependent setup times, in: Journal of the Operational Research Society 53 2002, S. 895-906
Gambardella und Dorigo (2000)	Gambardella, L. M.; Dorigo, M.: An Ant Colony System Hybridized with a New Local Search for the Sequential Ordering Problem, in; INFORMS Journal on Computing 12 2000, S. 237-255
Gambardella et al. (1999a)	Gambardella, L. M.; Taillard, E.; Agazzi, G.: MACS-VRPTW: A multiple Ant Colony System for vehicle routing problems with time windows, in: Corne, D.; Dorigo, M.; Glover, F. (Hrsg.): New Ideas in Optimization, McGraw-Hill 1999
Gambardella et al. (1999b)	Gambardella, L. M.; Taillard, E.; Dorigo, M.: Ant colonies for the quadratic assignment problem, in: Journal of the Operational Research Society 50 1999, S. 167-176
Goss et al. (1989)	Goss, S.; Aron, S.; Deneubourg, J.-L.; Pasteels, J.-M.: Self-organized shortcuts in the Argentine Ant, in: Naturwissenschaften 76 1989, S. 579-581
Hansmann (2001)	Hansmann, K.-W.: Industrielles Management, 7. Aufl. 2001
Kuntz und Snyers (1994)	Kuntz, P.; Snyers, D.: Emergent Colonization and Graph Partitioning, in: Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3, Cambridge Mass. 1994
Maniezzo und Carbonaro (2001)	Maniezzo, V.; Carbonaro, A.: Ant Colony Optimization: an overview, in: Ribeiro, C. (Hrsg.): Essays and Surveys in Metaheuristics, Kluwer 2001, S. 21-44
Maniezzo et al. (2001)	Maniezzo, V.; Carbonaro, A.; Golfarelli, M.; Rizzi, S.: An ANTS Algorithm for Optimizing the Materialization of Fragmented Views in Data Warehouses: Preliminary Results, in: Lecture Notes in Computer Science, Volume 2037, Berlin 2001, S. 80-89
Maniezzo und Carbonaro (2000)	Maniezzo, V.; Carbonaro,A.: An ANTS Heuristic for the Frequency Assignment Problem, in: Future Generation Computer Systems 16, Amsterdam 2000, Seite 927 – 935
Maringer (2002)	Maringer, D. G.: Wertpapierselektion mittels Ant Systems, in: Zeitschrift für Betriebswirtschaft 72 2002, S. 1221-1240
McMullen (2001)	McMullen, P. R.: An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives, in: Artificial Intelligence in Engineering 15 2001, S. 309-317
Merkle und Middendorf (2000)	Merkle, M.; Middendorf, M.: An Ant Algorithm with new Pheromone Evaluation Rule for Total Tardiness Problems, in: Cagnoni, S. (Hrsg.): Real-World Applications of Evolutionary Computing. Proceedings of EvoWorkshops 2000, Berlin 2000, S. 287-296
Merkle et al. (2000)	Merkle, M.; Middendorf, M.; Schmeck, H.: Ant Colony Optimization for Ressource-Constraint Project Scheduling, in: Proceedings of the Genetic and

	Evolutionary Computation Conference (GECCO-2000) Las-Vegas 2000
Michel und Middendorf (1999)	Michel, R.; Middendorf, M.: An ACO algorithm for the shortest common supersequence problem, in: Corne, D.; Dorigo, M.; Glover, F. (Hrsg.): New Ideas in Optimization, McGraw-Hill 1999
Mühlenbein (1997)	Mühlenbein, H.: Genetic Algorithms, in: Aarts, E.; Lenstra, J. K. (Hrsg.): Local Search in Combinatorial Optimization, Chichester 1997, S. 137-172
Pasteels et al. (1987)	Pasteels, J.-M.; Deneubourg, J.-L.; Goss, S: Self-organisation mechanisms in ant societies: Trail recrutment to newly discovered food sources, in: Experimentica Supplmentum 54 1987, S. 155-175
Roli et al. (2001)	Roli, A.; Blum, C.; Dorigo, M.: ACO for Maximal Constraint Satisfaction Problems, in: Proceedings of the Metaheuristics International Conference, MIC 2001, Porto 2001, S. 187-191
Schmundt (2000)	Schmundt, H.: Duft der Daten, in: Spiegel Jg. 2000 Heft 46, S. 264
Schoonderwoerd et al. (1997a)	Schoonderwoerd, R.; Holland, O.; Bruten, J.; Rothkrantz, L.: Ant-based Load Balancing in Telecommunications Networks, in: Adaptive Behavior 5 1997, S. 169-207
Schoonderwoerd et al. (1997b)	Schoonderwoerd, R.; Holland, O.; Bruten, J.: Ant-like Agents for Load Balancing in Telecommunications Networks. in: Proceedings of Agents'97, Marina del Rey 1997, S. 209-216.
Stützle und Dorigo (1999a)	Stützle, T.; Dorigo, M.: ACO Algorithms for the Quadratic Assignment Problem, in: Corne, D.; Dorigo, M.; Glover, F. (Hrsg.): New Ideas in Optimization, McGraw-Hill 1999
Stützle und Dorigo (1999b)	Stützle, T.; Dorigo, M.: ACO Algorithms for the Travelling Salesman Problem, in: Miettinen,K.; Mäkelä, M. M.; Neittaanmaki, P.; Periaux J. (Hrsg.): Evolutionary Algorithms in Engineering and Computer Science, Wiley, 1999
Stützle et al (2000)	Stützle, T.; den Besten, M.; Dorigo, M.: Ant Colony Optimization for the Total Weighted Tardiness Problem, in: Deb (Hrsg.): Proceedings of PPSN-VI, Sixth International Conference on Parallel Problem Solving from Nature, Berlin 2000, S. 611-620
Stützle (1998)	Stützle, T.: An Ant Approach to the Flow Shop Problem, in: Proceedings of EUFIT'98, Aachen 1998, S. 1560-1564
Stützle und Hoss (1999)	Stützle, T.; Hoss, H.: Improvement on the ant system: Introducing Max-MIN ant system, in: Voss S., Martello S., Osman I.H., Roucairol C. (Hrsg.): Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Boston 1999, S. 137-154
T'kindt et al. (2002)	T'kindt, V.; Monmarche, N.; Tercinet, F.; Laügt, D.: An ant colony optimization algorithm to solve a 2-maschine bicriteria flowshop scheduling problem, in: European Journal of Operational Research 142 2002, S. 250-257
Zimmermann (1995)	Zimmermann, W.: Operations Research – Quantitative Methoden zur Entscheidungsvorbereitung, 7. Aufl. Münschen 1995