

Ant Algorithmen

für kombinatorische Optimierungsprobleme

Heiko Stamer*

Institut für Informatik
UNIVERSITÄT LEIPZIG

9. Oktober 2001[†]

Zusammenfassung

In der Natur treffen wir auf vielfältige emergente Verhaltensweisen. Ein häufig betrachtetes Beispiel ist die Futtersuche eines Ameisenvolkes: Eingesetzte Arbeitsameisen helfen ihren Artgenossen durch den Einsatz von Botenstoffen (Pheromone), die Futterquelle leichter zu finden und damit den Sammelprozess zu beschleunigen.

In dieser Arbeit sollen eingangs solche biologischen Beispiele untersucht werden. Im Hauptteil stellen wir darauf aufbauende kollektive Algorithmen zur Optimierung (Ant System, Ant-Q, ACS) anhand des bekannten *Traveling Salesman Problem (TSP)* vor. Mit dieser Grundlage wenden wir uns dann konkreteren Anwendungen wie z.B. dem *Routing in optischen Netzwerken* zu und diskutieren die ACO Paradigmen im Rahmen des „Artificial Life“. Den Abschluß bildet die Vorstellung eigener Experimente, die an ACO Algorithmen durchgeführt wurden und zum *Recruiting Ant System (RAS)* geführt haben.

Inhaltsverzeichnis

1	Einleitung	2
2	Biologische Beispiele	3
2.1	Termiten	3
2.2	Bienen und Wespen	4
2.2.1	Tanz der Bienen	4
2.2.2	Temperaturregulierung bei der Überwinterung	4
2.3	Ameisen	5
2.3.1	Subtile Beobachtungen	6
3	Ant Algorithmen und deren Anwendung	6
3.1	Algorithmen	6
3.1.1	Ant System (AS)	6
3.1.2	Ant-Q	12
3.1.3	Ant Colony System (ACS)	12
3.1.4	<i>MAX-MIN</i> Ant System (<i>MMAS</i>)	13
3.1.5	Rang-basiertes Ant System (<i>AS_{rank}</i>)	13
3.2	Ant Colony Optimierung (ACO)	13
3.3	Konkrete Anwendungen	14
3.3.1	Telekommunikation	14
3.3.2	Optische Netzwerke (VWP-Routing)	15
4	Recruiting Ant System (RAS)	17
5	Zusammenfassung und Ausblick	18

*stamer@informatik.uni-leipzig.de, <http://stinfwww.informatik.uni-leipzig.de/~mai97ixb>

[†]Revision: 7. Januar 2005. Some images have been vectorized with potrace.

*Die Natur ist das weiseste Buch,
mit schönen Lehren auf jedem Blatt.*

(A. G. OEHLENSCHLÄGER)

1 Einleitung

Natürliche Vorgänge sind häufig Gegenstand wissenschaftlicher Untersuchungen. Zum einem, um Kenntnisse über die Natur als System zu erlangen, und zum anderen, um natürliche Techniken für menschliche Anwendungen zu adaptieren.

Der Begriff *Ant Algorithmen* wurde 1991 von DORIGO, MANIEZZO und COLORNI in ihrer Arbeit „Positive Feedback as a Search Strategy“ erstmals angegeben. Ant System (AS) wurde dann als Anwendung von DORIGO in seiner Dissertation „Optimization, Learning and Natural Algorithms“ vorgestellt. Neben der Adaptierung für das *Traveling Salesman Problem (TSP)* wurde AS auch für das *Quadratic Assignment Problem (QAP)* und andere Probleme portiert. Später arbeiteten DORIGO und GAMBARDILLA an verschiedenen Erweiterungen dieses Basisalgorithmus. Ant-Q ist beispielsweise eine Kombination von AS mit Q-Lernen¹. Ant Colony System (ACS) ist eine Vereinfachung von Ant-Q, die dann 1996 publiziert wurde. Besonders im Bereich der kombinatorischen Optimierungsprobleme haben diese Algorithmen gezeigt, daß sie den Vergleich mit herkömmlichen Heuristiken nicht zu scheuen brauchen. Aber es wurden auch konkrete Anwendungen, z.B. in der Telekommunikation, entwickelt, in denen ACO Verfahren ihre Leistungsstärke unter Beweis stellen können.

Seit 1998 gibt es internationale Workshops, um die Forschung an Ant Algorithmen zu intensivieren. Bisher wurden folgende Veranstaltungen durchgeführt:

- ANTS'98 - From Ant Colonies to Artificial Ants: First International Workshop on Ant Colony Optimization
Brüssel, Belgien, 15./16. Oktober 1998
- ANTS'2000 - From Ant Colonies to Artificial Ants: Second International Workshop on Ant Algorithms
Brüssel, Belgien, 8./9. September 2000

Diese Arbeit gliedert sich nun wie folgt: Im zweiten Abschnitt wollen wir anhand biologischer Beispiele die Grundlage der Ant Algorithmen klären. Dazu soll das Sozialleben der Termiten, der Tanz der Bienen und die Futtersuche einer Ameisenkolonie betrachtet werden. Abschnitt 3 führt dann in die Klasse der Ant Algorithmen als Nutzer solcher kollektiver Verhaltensweisen ein. Wir beschäftigen uns ausführlich mit dem Ant System (AS) und stellen darauf aufbauende Verbesserungen vor. Schließlich diskutieren wir die ACO Meta-Heuristik, die viele Gemeinsamkeiten der Ant Algorithmen charakterisiert. Im letzten Unterabschnitt wenden wir uns konkreten Anwendungen von ACO Verfahren zu, wie z.B. Telekommunikation und Routing in Optischen Netzwerken. Unsere eigenen Experimente und Verbesserungen am Ant System werden im Teil 4 mit dem Recruiting Ant System (RAS) vorgestellt. Letztendlich fassen wir im fünften Abschnitt die gesammelten Erkenntnisse zusammen und geben einen kurzen Ausblick in die Zukunft.

¹eine bekannte *reinforcement learning*-Strategie

2 Biologische Beispiele

Kollektives Verhalten in der Natur läßt sich recht gut am Beispiel von *Insektenstaaten* verdeutlichen. Honigbienen, Ameisen und Termiten bilden besonders hoch organisierte Staaten mit strukturell und funktionell spezialisierten Individuen, die jeweils für sich allein auf Dauer nicht lebensfähig sind.

Das Leben in der Gemeinschaft ist hier eine kollektive Maßnahme zur Arterhaltung. Dazu sind soziale Verhaltensweisen erforderlich, deren Ausprägungsgrad von der Organisationsstufe des zugrundeliegenden Verbandes abhängt. Jedoch verkörpert eine bloße Ansammlung von Tieren, wie z.B. der Schlafverband von Fledermäusen, der Überwinterungsverband von Feuersalamandern oder der Fraßverband von Schmetterlingsraupen noch kein kollektives Verhalten. Echte *Sozietät* setzt intensivere gegenseitige Bezugnahme der teilnehmenden Individuen voraus, die meist durch Gemeinschaftshandlungen, zumindest aber durch gemeinsame Verständigungsmöglichkeiten (Kommunikationsformen) gegeben sein muß.

Einige in der Natur vorkommende Verhaltensweisen und Organisationsstrukturen wollen wir nun an drei konkreten Beispielen betrachten.

2.1 Termiten

Ein Termitenstaat besteht mitunter aus mehreren Millionen Tieren. Die *Königin* produziert ununterbrochen vom *König* befruchtete Eier. *Soldaten* übernehmen die Verteidigung und *Arbeiter*, die, wie die Soldaten auch, unfruchtbar sind, verrichten alle übrigen Tätigkeiten: Abtransport der Eier, Fütterung und Pflege des Königs-paares und der Larven, Errichtung und Reparatur der oft riesigen Nestburgen, Bau von Lüftungsschächten, Tiefengrabungen zum Grundwasser, Nahrungssuche, Zerkleinern und Herantragen der Nahrung, Anlegen und Betreuung von Pilzgärten und anderes mehr.

Das Verhalten der Termiten ist im wesentlichen angeboren, zugleich aber durch Umweltfaktoren erstaunlich formbar und auf die Tätigkeit der Kollektivmitglieder abgestimmt.



Abbildung 1: Termitenhügel in der afrikanischen Savanne

2.2 Bienen und Wespen

Bienen- und Wespenvölker setzen sich aus zehntausenden von Individuen zusammen. Obwohl die Tiere weitgehend selbständig handeln, kann man auch hier ein komplexes Sozialleben beobachten. Zwei der interessantesten Verhaltensmuster sollen nachfolgend diskutiert werden. Weitere Fakten und Formalisierungen findet der geneigte Leser in [3] (mathematisches Modell, WSCCS), [4] und [5]. Einen allgemeineren Überblick gibt dazu [1].

2.2.1 Tanz der Bienen

Mit dieser Verhaltensform übermitteln die Honigbienen Informationen über Richtung, Entfernung und Art einer Nahrungsquelle an Artgenossen. Dabei repräsentiert die Anzahl der Tanzläufe einen funktionalen Zusammenhang zur Entfernung, während die Flugfigur selbst den Winkel zwischen Sonnenstand und direkter Linie zur Nahrungsquelle wiedergibt. Abbildung 2 zeigt dieses Verhalten schematisch. Die Funktion zwischen Anzahl der Tanzläufe und Entfernung ist von der geographischen Lage abhängig (unterschiedliche Bienendialekte). Da aber lokal die gleichen Werte verwendet werden, stellt diese Besonderheit kein Problem dar.

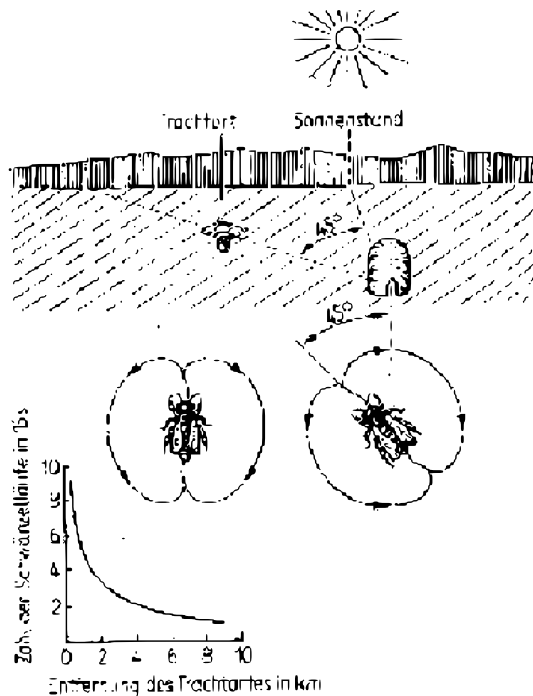


Abbildung 2 (aus [1]): Schwänzeltanz der Honigbiene

2.2.2 Temperaturregulierung bei der Überwinterung

Zum Zwecke der Temperaturregulierung bilden die Individuen sogenannte *Cluster*. Die Bienen agieren hierbei im Wesentlichen als ein *Superorganismus*, welcher die Form und Größe des Clusters je nach umgebenden Temperaturbedingungen anpassen kann. Biologische Untersuchungen [6] haben gezeigt, daß Bienen im Zentrum dabei nicht mit den Tieren am Rand kommunizieren. Dies scheint die Hypothese zu untermauern, daß Formierung und Verhalten des Schwarms Resultat unabhängiger Aktionen einzelner Bienen ist. In [5] wird diese These mit einem agentenbasierten Modell simuliert und untersucht.

2.3 Ameisen

In diesem Unterabschnitt wollen wir das Verhalten von Ameisen näher betrachten. Neben einigen biologischen Grundlagen soll hier aber speziell auf Methoden der Arbeitsteilung und Kommunikation eingegangen werden. Diese natürlichen Strategien sind Basis der im Abschnitt 3.1 (auf Seite 6) vorgestellten Ant Algorithmen.

Der lateinischen Name FORMICIDAE bezeichnet die Familie der Ameisen, die sich in 11 Unterfamilien mit über 12 000 Arten gliedert. Ein typisches Merkmal aller Ameisen ist das stark ausgeprägte Sozialleben, welches in seiner Organisation nur noch von der Blindmulle und dem Menschen übertroffen wird. Die einzelnen Ameisenarten unterscheiden sich zum Teil erheblich in Größe, Körperbau, Verhalten, Kommunikation, sozialer Organisation und Spezialisierung. Trotzdem lassen sich grundsätzliche Verhaltensmuster beschreiben:

Koloniebildung: Fast alle Arten bilden Kolonien, die meist aus einer Königin und etlichen „sterilen“ Arbeitsameisen bestehen. Während die Königin lediglich für die Fortpflanzung verantwortlich ist, erledigen die Arbeiter eine Vielzahl von Tätigkeiten. Auf einige davon, wie z.B. die Verteidigung und die Futtersuche, wollen wir gleich noch genauer eingehen. Fruchtbare Tiere besitzen zudem Flügel und können somit (in entsprechender Entfernung liegende) neue Kolonien gründen.

Orientierung: Hierbei unterscheidet man zwischen optischer und chemischer Orientierung. Bei Ersterem kommen Lichtquellen oder markante Geländemarken als Orientierungspunkte in Frage. Weitaus wichtiger ist aber das Auslegen und Folgen von Duftspuren. Die Substanzen, die dabei zum Einsatz kommen, werden allgemein als *Pheromone* bezeichnet und lösen bei anderen Tieren der selben Art ein bestimmtes Verhalten aus. Insbesondere basiert auch die Kommunikation auf Pheromonausschüttung. Zur Orientierung werden diese Botenstoffe von Arbeitsameisen beim Markieren von Wegen zwischen Futterstelle und Nest benutzt. Die **Futtersuche** gestaltet sich im Allgemeinen wie folgt: Eine Ameise findet durch zufällige Suche einen Futterplatz. Auf dem Weg zurück ins Nest hinterlässt sie eine Pheromonspur. Andere Arbeitsameisen orientieren sich an dieser Spur, finden den Futterplatz und verstärken/erneuern beim Rückweg die Botenstoffkonzentration. Wenn das Futter aufgebraucht ist, werden keine Pheromone mehr ausgeschüttet und die Spur verschwindet nach kurzer Zeit von selbst. Da eine solche Spur wahrscheinlich nicht gerichtet ist, muß die Ameise auch noch auf andere Orientierungsmethoden zurückgreifen, um festzustellen, ob der Weg zum Nest oder zur Futterstelle führt. Am Rande sei hier noch die geomenotaktische Orientierung (Schwerkraft) erwähnt, die aber nur eine begrenzte Genauigkeit besitzt und vorrangig im Nest oder an Bäumen zur Anwendung kommt.

Kommunikation: Eine notwendige Voraussetzung für soziales Zusammenleben ist die Kommunikation der Individuen untereinander. Ameisen besitzen vielfältige Verständigungsmöglichkeiten, bei der meist optische, chemische, akustische und mechanische Signale zusammenwirken. Hauptsächlich findet die Kommunikation über etliche chemische Botenstoffe (Pheromone) statt. Diese sind z.B. bei der **Verteidigung** wichtig: Eine Gefahrenquelle (Gegner) wird mit dem „Alarmstoff“ benetzt. Die erregten Ameisen laufen mit hoher Geschwindigkeit umher und wenden sich mit gespreizten Kiefern der Alarmstoffquelle zu. Sie berühren den Gegner mit ihren Fühlern, beißen zu oder zerren ihn möglichst weit fort. Auch optische und taktile Wahrnehmungen sind von Bedeutung. Dies zeigt sich beispielsweise beim Prozess der **Rekrutierung**: Von einer Kundschafterin werden eine oder mehrere Ameisen durch Aufforderungsgebärden zu einem bestimmten Verhalten animiert. Bei einigen Arten spielt diese Kommunikationsform bei der Ausbeutung von Futterquellen oder beim Umzug in ein neues Nest eine Rolle.² Akustische Signale werden nur selten eingesetzt und haben lediglich unterstützende Wirkung.

²Auffallend ist hierbei der sogenannte *Tandemlauf*, bei dem die Kundschafterin eine andere Ameise zur Futterquelle führt. Weitere Informationen dazu finden sich z.B. in [7].

2.3.1 Subtile Beobachtungen

Die Größe einer Kolonie ist ein wichtiges Maß für das „Funktionieren“ der schon beschriebenen Verhaltensmuster. Bei kleineren Kolonien herrscht bei der Futtersuche Rekrutierung vor, da die chemische Orientierung nur mit einer Vielzahl an Individuen aufrechterhalten werden kann. SUMPSTER und BEEKMAN beschreiben in „*How many agents make a working whole ?*“ [8] einige Experimente dazu. Insbesondere stellen sie fest, daß bei Agentensystemen mit kleineren und mittleren Populationsgrößen eine Kombination von Rekrutierungsstrategien erfolgreicher sein könnte.

In anderen Untersuchungen wurde das **synchrone** Verhalten innerhalb einer Ameisenkolonie betrachtet. Dabei fiel auf, daß periodische Zusammenhänge von Ruhe- bzw. Aktivitätsphasen bei bestimmten Arten auftreten. Solche Periodizitäten gelten als wichtiges Indiz emergenter Interaktion und sind in viele Simulationssysteme für *Artificial Life* übertragen worden.

3 Ant Algorithmen und deren Anwendung

Einige der im vorherigen Abschnitt beschriebenen Verhaltensmuster wurden, wie andere biologische Beispiele (Genetische Algorithmen, Simulated Annealing, Neuronale Netze, ...) auch, von der Informatik zur Modellbildung eingeführt und für den Algorithmenentwurf adaptiert.

Systeme, die sich kollektiver Techniken bedienen, werden häufig als *Multi-Agenten-Systeme* bezeichnet. Unabhängige Agenten - in unserem Fall virtuelle Ameisen - erledigen eine bestimmte Aufgabe, zu der einzelne Individuen alleine nicht fähig wären bzw. eine deutlich geringere Performance erreichen würden. Wir wollen nun hauptsächlich auf das besprochene Verhalten *Futtersuche mittels Pheromonspuren* der Ameisen eingehen und darauf aufbauende Ant Algorithmen näher betrachten. Es soll uns dabei vorrangig um kombinatorische Optimierungsprobleme gehen.

3.1 Algorithmen

Ausgehend von dem durch DORIGO et al. in [11], [10], [9], und [12] eingeführten *Ant System (AS)* beschreiben wir diverse Verbesserungen und Abwandlungen dieses Basisalgorithmus, wie z.B. *Ant-Q*, *ACS*, etc., die im Laufe der Zeit hauptsächlich von Forschergruppen aus Italien entwickelt wurden. Die hier verwendete Reihenfolge entspricht in etwa der zeitlichen Entwicklungsgeschichte.

3.1.1 Ant System (AS)

Dieser Algorithmus wurde 1991 im Rahmen der Forschung zu intelligenten Systemen für kombinatorische Optimierung von MARCO DORIGO in Zusammenarbeit mit ALBERTO COLORNI und VITTORIO MANIEZZO an der POLITECNICO DI MILANO (Italien) entwickelt. Weil die zugrundeliegenden Methoden an biologische Verhaltensweisen von Ameisen bei der Suche und der Ausbeutung einer Futterquelle angelehnt sind, wurde der Algorithmus als *Ant System (AS)* bezeichnet.

Grundlage: Dieses Optimierungsverfahren greift auf einen *autokatalytischen* Prozeß³ zurück. Betrachten wir nochmal das Beispiel *Futtersuche eines Ameisenvolkes*: Anfangs verläuft die Suche zufällig. Wenn aber ein Futterplatz gefunden wurde, wird der Weg dorthin durch Pheromone markiert. Andere Ameisen erkennen die Spur und folgen ihr, wobei sie mittels Botenstoffausschüttung die Markierung weiter verstärken. Dadurch werden weitere Ameisen aufmerksam, die die Intensität nochmals steigern. Noch interessanter wird

³Ein *autokatalytischer Prozeß* ist ein Vorgang, der sich durch häufige Selbstaussführung extrem verstärkt und dadurch schnell konvergiert, aber ggf. auch „außer Kontrolle“ geraten kann.

dieser Prozeß, wenn sich plötzlich ein Hindernis auf dem markierten Pfad befindet. Die Ameisen suchen sich vorerst irgendeinen Weg um diesen Gegenstand herum. Da aber im gleichen Zeitraum mehr Ameisen den kürzeren Weg zurücklegen können, verstärkt sich dort die Pheromonkonzentration, was wiederum mehr Tiere diesen Weg zurücklegen läßt. Eine möglichst kurze Umgehung ist gefunden. In Abbildung 3 aus [12] wird dieses Verhalten anschaulich dargestellt. Ant System nutzt einen solchen Ansatz in Kombination mit der Bewertung der Futterquelle.

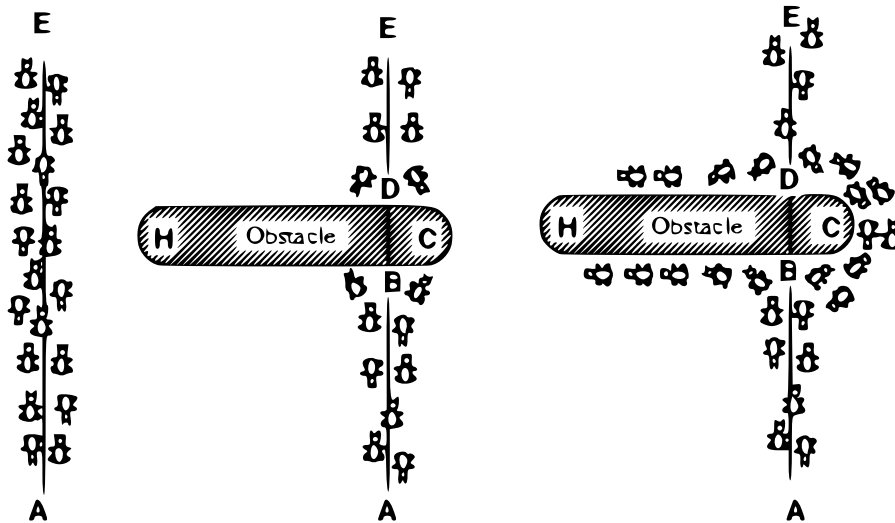


Abbildung 3: Umgehung von Hindernissen (aus [12])

Ameisen als Agenten: Da hier keine Simulation von natürlichen Systemen betrieben werden soll, unterscheiden sich die Eigenschaften der eingesetzten Agenten von ihren natürlichen Vorbildern:

- Die virtuellen Ameisen verfügen über einen Speicher,
- sind nicht komplett blind und
- leben in einer Umgebung, wo die Zeit diskret ist.

Algorithmus am Beispiel des TSP: Der Ant System Algorithmus soll am Beispiel des *Traveling Salesman Problem (TSP)* vorgestellt werden. Das TSP ist die Fragestellung nach der kürzesten Route, die ein Handelsreisender, der alle Städte einer gegebenen Menge besuchen will, zurücklegen muß. Dieses Problem ist \mathcal{NP} -hart, d.h. es sind nur Algorithmen mit exponentieller Laufzeit bekannt, die zu einer optimalen Lösung führen. Deshalb wurden hierfür etliche Heuristiken und Algorithmen entwickelt, welche Näherungs- und/oder suboptimale Lösungen liefern. AS gehört neben Greedy-Algorithmen, Hill Climbing, Sampling & Clustering, GA, Neuronalen Netzen, 2-opt, Lin-Kerningham und anderen Meta-Heuristiken auch mit zu dieser Klasse von approximativen Lösungsverfahren.

Eine TSP-Instanz ist durch einen Graphen $\mathcal{G} = (N, E)$ gegeben, wobei $N = \{1, 2, \dots, n\}$ die Menge der Städte und E die Verbindungen zwischen ihnen repräsentiert. Auf der euklidischen Ebene können wir die Entfernung zwischen den Städten i und j mit

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

berechnen. Wir suchen nun eine Route $R = \{(c_1, c_2), (c_2, c_3), \dots, (c_{n-1}, c_n), (c_n, c_1)\}$ mit $\forall \ell = 1, 2, \dots, n : c_\ell \in N$, bei der die Summe $\sum_{e \in R} d_e$ relativ klein wird.

Die Anzahl der virtuellen Ameisen, die sich zum Zeitpunkt t in der Stadt i befinden, wollen wir mit $b_i(t)$ bezeichnen, so daß sich die Gesamtanzahl durch $m = \sum_{i=1}^n b_i(t)$ ergibt. Da während des Algorithmus keine Ameisen hinzukommen bzw. verschwinden, ist

die Gesamtanzahl hinsichtlich des Zeitpunkts t konstant. τ_{ij} gibt die Pheromonintensität auf der Verbindung zwischen i und j an.

Jede Ameise ist ein einfacher Agent mit folgendem Verhalten:

- Wenn die Ameise von i nach j „wandert“, markiert sie die Verbindung mit Pheromonen.
- Die Frage, zu welcher Stadt die Ameise als nächstes geht, wird durch Wahl der Wahrscheinlichkeit in Abhängigkeit von der Entfernung und der Pheromonintensität des Weges dorthin bestimmt.⁴
- Damit Städte nicht wiederholt angelaufen werden, führt die Ameise in einer sogenannten *Tabu-Liste*⁵ Buch darüber, wo sie schon gewesen ist.

Die Änderung der Pheromonintensität auf den Verbindungen von \mathcal{G} kann durch

$$\tau_{ij}^{\text{neu}} = \rho \cdot \tau_{ij}^{\text{alt}} + \Delta\tau_{ij}$$

beschrieben werden, wobei ρ ein Flüchtigkeitskoeffizient und

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k$$

ist. Dabei ist $\Delta\tau_{ij}^k$ die Pheromonmenge, mit der die k -te Ameise den Weg zwischen i und j markiert hat. Dieser Wert ist insbesondere 0, wenn die Ameise sich gar nicht von i nach j bewegt hat. Der Flüchtigkeitskoeffizient ρ muß kleiner 1 gewählt werden, da sich sonst unbegrenzt Pheromone auf den Verbindungen ansammeln. Die Anfangsintensität τ_{ij} zum Zeitpunkt $t = 0$ sollte mit relativ kleinen Werten vorinitialisiert werden. Es kann dabei der gleiche Wert für alle Verbindungen genutzt werden. Die Tabu-Liste speichert die Städte, die bis zum Zeitpunkt t schon besucht wurden. Dies sichert, daß die Ameise stets eine komplette Route nach Maßgabe des TSP abläuft. Danach wird die Liste geleert, und die Ameise kann nach einer neuen Route suchen. Unter *Sichtbarkeit* verstehen wir die einfache Ausgangsheuristik $\eta_{ij} = 1/d_{ij}$. Die Auswahlwahrscheinlichkeit p_{ij}^k wird nun wie folgt definiert:

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in A_k} (\tau_{il})^\alpha \cdot (\eta_{il})^\beta} & \text{falls } j \in A_k \\ 0 & \text{sonst} \end{cases} \quad (1)$$

Hier ist A_k eine Menge von Städten, die von der k -ten Ameise noch nicht besucht wurden. Die Parameter α und β wichten den Einfluß der Ausgangsheuristik und der Pheromonsteuerung, was nachher noch genauer erläutert wird.

Ursprünglich wurden drei verschiedene Algorithmen unterschieden, die *Ant-density*, *Ant-quantity* und *Ant-cycle* genannt werden. Später hat sich durch Untersuchungen gezeigt, daß *Ant-cycle* bessere Lösungen als die beiden vorgenannten liefert. In neueren Veröffentlichungen wird deshalb häufig nur noch *Ant-cycle* als *Ant System Algorithmus* bezeichnet. Wir wollen zuerst *Ant-density* und *Ant-quantity* betrachten, welche noch recht nahe am biologischen Vorbild orientiert sind.

Während beim *Ant-density* Verfahren die Pheromonänderung auf der Verbindung zwischen i und j von der Entfernung d_{ij} unabhängig ist, werden bei *Ant-quantity* bei kürzeren Strecken mehr Botenstoffe abgelegt als bei längeren Strecken. Dies führt uns zu folgenden Formeln (Q : Pheromongrundmenge einer Ameise):

Ant-density Modell:

$$\Delta\tau_{ij}^k = \begin{cases} Q & \text{wenn die } k\text{-te Ameise von } i \text{ nach } j \\ & \text{unterwegs war} \\ 0 & \text{sonst} \end{cases} \quad (2)$$

⁴siehe Formel (1)

⁵Dies ist nur eine technische Notwendigkeit und lehnt sich nicht an die bekannten Tabu-Suchstrategien an. Lediglich der Name ist ähnlich gewählt.

Ant-quantity Modell:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{d_{ij}} & \text{wenn die } k\text{-te Ameise von } i \text{ nach } j \\ & \text{unterwegs war} \\ 0 & \text{sonst} \end{cases} \quad (3)$$

Der eigentliche Algorithmus kann jetzt in Pseudocode einfach beschrieben werden:

```
Initialisierung
FOR  $t = 1$  TO Anzahl der Iterationen des Algorithmus
    Wiederhole, solange Ameisen noch Städte besuchen können:
        FOR  $k = 1$  TO  $m$ 
            Wähle mit Wahrscheinlichkeit  $p_{ij}$  nach Formel (1) eine Stadt  $j$  aus,
            die als nächste besucht werden soll.
            Setze die neue Pheromonintensität  $\tau_{ij}$  für alle Verbindungen aus  $\mathcal{G}$ .
        Leere alle Tabu-Listen.
```

Bei der *Initialisierung* wird die Ausgangsstadt i jeder Ameise und die Anfangspheromonintensität τ_{ij} auf allen Verbindungen von \mathcal{G} festgelegt.

Wir wollen uns nun dem Ant-cycle Modell zuwenden. Der wesentliche Unterschied besteht im Zeitpunkt der Pheromonänderung. Während bei Ant-density und Ant-quantity das Anpassen nach jedem Teilschritt erfolgte, werden bei Ant-cycle erst nach der kompletten Route aller Ameisen die Pheromonlevel in Abhängigkeit von der Gesamtlänge der zurückgelegten Strecke neu gesetzt.

Ant-cycle Modell:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{wenn die } k\text{-te Ameise auf der Ver-} \\ & \text{bindung } (i, j) \text{ unterwegs war} \\ 0 & \text{sonst} \end{cases} \quad (4)$$

L_k ist hierbei die Gesamtlänge der Route, auf der die k -te Ameise unterwegs war. Der Algorithmus gestaltet sich nun wie folgt:

```
Initialisierung
FOR  $t = 1$  TO Anzahl der Iterationen des Algorithmus
    FOR  $k = 1$  TO  $m$ 
        Wiederhole, solange  $k$ -te Ameise noch Städte besuchen kann:
            Wähle mit Wahrscheinlichkeit  $p_{ij}$  nach Formel (1) eine Stadt  $j$  aus,
            die als nächste besucht werden soll.
            Berechne die Gesamtlänge  $L_k$ , die von der  $k$ -ten Ameise zurückgelegt
            wurde.
            Setze die neue Pheromonintensität  $\tau_{ij}$  für alle Verbindungen aus  $\mathcal{G}$ .
        Leere alle Tabu-Listen.
```

Ant-cycle ist nicht so nah am biologischen Vorbild angelehnt wie Ant-density bzw. Ant-quantity, da „richtige“ Ameisen bereits während des Weges Botenstoffe ausschütten und die Pheromonmengen nicht im Nachhinein noch anpassen können. Außerdem ist wissenschaftlich bisher unklar, ob sie, Strecken hinsichtlich der Länge zu messen, in der Lage sind. Für Optimierungszwecke hat sich aber Ant-cycle als der eindeutig bessere Algorithmus herausgestellt, gerade weil hier globale Informationen (Gesamtlänge der Route) in die Pheromonänderung einfließen. Dies klärt zumindest teilweise auch, warum Ant System überhaupt funktioniert und, wie im folgenden Absatz erläutert, sehr gute Lösungen für das TSP liefert.

Experimentelle Untersuchungen: DORIGO et al. haben in [10], [9], [12] den Algorithmus experimentell untersucht. Dabei wurde im Wesentlichen der Einfluß der vielen Parameter näher betrachtet. Einige Ergebnisse daraus möchten wir hier wiedergeben.

Vorher wollen wir noch einen Begriff einführen: Das Verfahren *stagniert*, wenn alle Ameisen sich nur noch auf einer Route bewegen. Das System „erkundet“ somit keine neuen Verbindungen mehr, weshalb der Algorithmus auch abgebrochen werden kann.

Die beiden Parameter α und β aus der Formel (1) steuern maßgeblich das Verhalten des Algorithmus. Beide Werte sollten größer Null gewählt werden. Wenn $\alpha = 0, \beta > 0$ degeneriert Ant System zu einem Greedy-Algorithmus, bei dem die nächste Stadt in Abhängigkeit von ihrer Entfernung (Sichtbarkeitskriterium) bestimmt wird. Dies führt lokal zu akzeptablen Ergebnissen, insgesamt betrachtet ist die Lösung aber eher unzureichend. Umgekehrt bringt $\alpha > 0, \beta = 0$ aber auch keinen Gewinn, da somit keine lokale Ausgangsheuristik vorhanden ist (vgl. Absatz *Allgemeine Charakteristik*). Die Auswahl von α, β muß also sehr sensibel erfolgen. Die Übergewichtung von α führt schnell zur Stagnation des Verfahrens; wird β zu groß gewählt, erhält man keine guten Lösungen. In den Experimenten haben sich $\alpha = 1$ und $\beta = 5$ als optimale Kombination herausgestellt.

Der Flüchtigkeitskoeffizient ρ beeinflusst ebenfalls die Qualität der gefundenen Routen. Da bei Ant-cycle zu Beginn stark auf die Ausgangsheuristik zurückgegriffen wird, muß das System auch in der Lage sein, diese Starttrouten zu vergessen, sobald die Pheromonsteuerung deutlich bessere globale Ergebnisse liefert. Deshalb wird hier $\rho = 0.5$ gesetzt, während durch die lokale Bewertung von Ant-density und Ant-quantity dort eher $\rho = 0.99$ eine gute Wahl ist.

Die Pheromongrundmenge Q jeder Ameise hat sich, wenn sie nicht zu klein gewählt wird, als unwesentlich erwiesen und kann deshalb beliebig gewählt werden. Meist wird jedoch $Q = 100$ angegeben, da so eine gewisse Vergleichbarkeit gewährleistet ist.

Weiterhin von Bedeutung ist die Anzahl der eingesetzten Agenten m und ihre Anfangsverteilung auf die Städte. DORIGO et al. stellen in [12] fest, daß $m \approx n$ ein optimales Laufzeitverhalten für dieses Optimierungsproblem (TSP) ergibt. Bei der Verteilungsmethode im Rahmen der Initialisierung ergaben sich deutlich bessere Ergebnisse, wenn die Ameisen in gleicher Anzahl (uniform) in alle Städte aufgeteilt wurden. Die zufällige Verteilung brachte keine signifikante Verbesserung.

Die Experimente aus [10], [9], [12] zeigen, daß der Algorithmus für solche Optimierungsaufgaben wie das Traveling Salesman Problem sehr gut geeignet ist und im Vergleich zu anderen natürlichen Heuristiken (GA, Simulated Annealing, Space Filling Curve, Tabu Search) hervorragend abschneidet. In manchen Fällen lieferte er für Testinstanzen sogar kürzere Routen, als die mit solchen Meta-Heuristiken bisher erreichten. Abbildung 4 zeigt solch ein Beispiel. Der Algorithmus verfügt zudem über eine relativ geringe Laufzeitkomplexität von $O(T \cdot n^3)$, wenn man $m \approx n$ wählt und T die Anzahl der Iterationen angibt. Entsprechende Statistiken und Diagramme können den zugehörigen Veröffentlichungen entnommen werden. In [29] werden diverse natürliche Heuristiken (GA, SA, S & C, TS, NN, AS) verglichen.

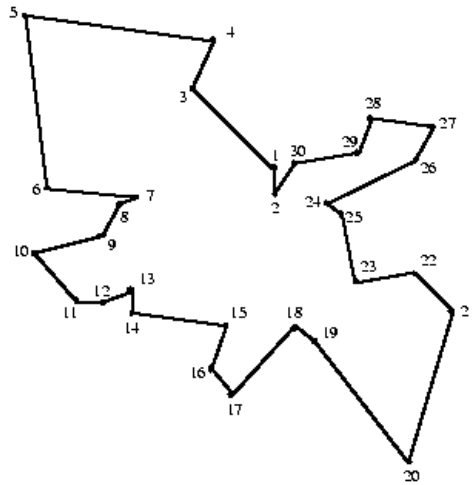


Abbildung 4 (aus [12]): Beste Route ($L = 423.741$) für das Oliver30 TSP
Ant-cycle mit 342 Iterationen ($\alpha = 1$, $\beta = 5$, $\rho = 0.5$, $Q = 100$)

Übertragbarkeit auf andere Probleme: Das Ant System (AS) ist nicht nur auf das TSP anwendbar. Vielmehr handelt es sich um einen Algorithmus, der für etliche kombinatorische Probleme eingesetzt werden kann. Dazu sind i.A. folgende Anpassungen nötig:

- Repräsentation des Problems als Graph, in dem viele einfache Agenten nach einer Lösung suchen können.
- Implementierung des autokatalytischen Prozesses, z.B. durch positive Bewertung von Verbindungen (Pheromonsteuerung).
- Eine entsprechende Ausgangsheuristik, die zur Konstruktion von lokalen Lösungen führt (Greedy-Algorithmus).
- Einhaltung der Problembedingungen durch Mechanismen, wie z.B. die Tabu-Liste.

In [12] wird diese Adaption für das *Quadratic Assignment Problem (QAP)* und das *Job Scheduling Problem (JSP)* durchgeführt. Andere Arbeiten verwenden AS für *Vehicle Routing Problem (VRP)* [13], *Graph Coloring* [14] und *Shortest Common Supersequence* [15].

Parallelisierbarkeit: BULLNHEIMER, KOTSIS und STRAUSS untersuchten in [17] synchrone und asynchrone Parallelisierungsstrategien für das Ant System am Beispiel des TSP. Sie kamen zu dem Schluß, daß, bedingt durch gute Parallelisierbarkeit von Ant-cycle, in verteilten Umgebungen auch größere Probleminstanzen effizient bearbeitet werden können. Ein anderes performantes Beispiel ist ANTabu [30] (C/PVM Implementierung) für das QAP.

Allgemeine Charakteristik: Abschließend kann man feststellen, daß es sich beim Ant System (AS) um einen verteilten autokatalytischen Prozeß handelt, der im Wesentlichen für klassische Optimierungsaufgaben eingesetzt werden kann. Anfangs arbeiten die Agenten mit einer Ausgangsheuristik und erreichen gute lokale Lösungen. Im weiteren Verlauf des Algorithmus erlangt die Kombination solcher Teilstücke eine größere Bedeutung. Das passiert, weil die Ameisen über eine globale Datenstruktur (Pheromonfeld τ_{ij}) kommunizieren und damit Präferenzen konstruieren können. Sie arbeiten also gemeinsam an einer Gesamtlösung, wobei die vorgegebene Ausgangsheuristik immer mehr an Einfluß verliert.

Es konnte zwar bisher nicht bewiesen werden, daß dieser Meta-Prozeß konvergiert, in den besprochenen Experimenten haben sich jedoch Ergebnisse gezeigt, welche diesen Ansatz rechtfertigen. Außerdem erkennt man hieran hervorragend, daß biologische Verhaltensmuster - entsprechend adaptiert - in der Informatik erfolgreich genutzt werden können.

3.1.2 Ant-Q

Im Jahr 1995 kommt es zu einer ersten Verbesserung des Ant System's. MARCO DORIGO und LUCA MARIA GAMBARDELLA führen in [18] eine Familie von Algorithmen ein, welche starke Ähnlichkeiten mit dem von WATKINS 1989 in seiner Dissertation beschriebenen Q-Lernen haben. Es zeigte sich, daß Ant-Q bei symmetrischen und asymmetrischen TSP schneller konvergierte und auch noch bessere Lösungen lieferte. Drei wesentliche Merkmale unterscheiden Ant-Q vom Ant System:

1. **Stadtauswahlregel jeder Ameise:** Die Auswahl der nächsten Besuchsstadt des Agenten wird nun durch das *pseudo-random-proportional action choice rule* bestimmt:

$$j = \begin{cases} \arg \max_{j \in A_k} \{(\tau_{ij})^\delta \cdot (\eta_{ij})^\beta\} & \text{falls } q \leq q_0 \\ S & \text{sonst} \end{cases} \quad (5)$$

Die Ameise wählt mit Wahrscheinlichkeit q_0 den Weg, der dem Pheromonfeld entsprechend z.Zt. der Beste ist oder startet mit $(1 - q_0)$ eine Erkundung der Umgebung. Hierbei ist S eine gemäß Formel (1) zufällig gewählte Stadt. Bei Experimenten haben sich $\delta = 1$ und $q_0 = 0.9$ als geeignete Parameter herausgestellt. Die anderen Variablen sind analog zu Ant System.

2. **Änderung der Pheromonintensität:** Die Änderung der Pheromonwerte erfolgt wie gehabt nach dem Ant-cycle Modell, wobei aber nur die lokal bzw. global beste Route beachtet wird. Während ursprünglich die Route Verwendung fand, die in jeder Iteration (lokal) die kürzeste war, hat sich später gezeigt [19], daß nur geändert werden sollte, wenn die Route über alle bisherigen Iterationen (global) die Beste ist. Insbesondere fließt also nicht jede Route in die Pheromonberechnung ein, wie es noch im Ant System der Fall war.
3. **Zusätzliche Änderung der Pheromonintensität:** Um bereits „durchsuchte“ Routen für folgende Ameisen weniger interessant zu machen, kommt eine lokale Pheromonänderung hinzu, die nach jeder Bewegung des Agenten sofort zum Tragen kommt:

$$\tau_{ij}^{\text{neu}} = (1 - \alpha) \cdot \tau_{ij}^{\text{alt}} + \alpha \cdot (\Delta\tau_{ij} + \gamma \cdot \max_{\ell \in A_k} \tau_{i\ell}) \quad (6)$$

In dieser ans Q-Lernen angelehnter Formel ist $\Delta\tau_{ij}$ der aus Ant System bekannte *reinforcement*-Term und $\gamma \cdot \max_{\ell \in A_k} \tau_{i\ell}$ eine Bewertung des Folgezustands. Mit $\alpha = 0.1$ wurden gute Resultate erzielt. In [19] wird beschrieben, wie wichtig die korrekte Auswahl von γ ist. Man sollte $0.2 \leq \gamma \leq 0.6$ wählen.

In [19] werden einige Experimente mit dem Algorithmus beschrieben, die dem interessierten Leser weitere Einblicke geben können. Eine schöne Zusammenfassung zum Q-Lernen ist beispielsweise [28].

3.1.3 Ant Colony System (ACS)

Ant Colony System (ACS), das 1996 von DORIGO und GAMBARDELLA in [20] vorgestellt wurde, ist im Wesentlichen nur eine Vereinfachung von Ant-Q. Neben dem Festsetzen des Parameters δ aus (5) auf die Konstante 1 wurde vorallem die umständliche Berechnung $\gamma \cdot \max_{\ell \in A_k} \tau_{i\ell}$ aus (6) durch die Konstante τ_0 ersetzt, welche der Anfangspheromonintensität auf allen Verbindungen entspricht. Experimente in [20] hatten gezeigt, daß beide Ansätze ein ähnliches Konvergenzverhalten aufweisen. Deshalb wurde die aus dem Q-Lernen stammende Formel zu Gunsten der Vereinfachung in ACS entfernt. Weitere Resultate sind in [25] zusammengefasst.

3.1.4 *MAX-MIN* Ant System (*MMAS*)

Das *MAX-MIN* Ant System (*MMAS*) ist eine direkte Verbesserung des AS. Es verwendet dieselbe Strategie zur Konstruktion von Lösungen wie das Ant System und ist erstmals in [21] untersucht wurden. Auch [22] gibt einen guten Überblick zu den gemachten Veränderungen:

1. **Änderung des Pheromonintensität:** Ähnlich wie bei Ant-Q wird auch hier nur verändert, wenn die gefundene Route besser ist als die bisher in allen Iterationen erreichten Lösungen (global beste Route). Alle anderen Wege werden bei der Pheromonänderung nicht beachtet (vgl. Ant-Q).
2. **Einführung von Pheromongrenzwerten:** Es werden obere und untere Schranken ($\forall \tau_{ij} : \tau_{\min} \leq \tau_{ij} \leq \tau_{\max}$) für die Pheromonintensität festgelegt. Experimente aus [23] haben gezeigt, daß die unteren Grenzwerte wesentlich wichtiger sind, da nach oben mittels Flüchtigkeitskoeffizient eine natürliche Schranke gegeben ist.
3. **Initialisierung:** Die Anfangspheromonwerte werden auf die oberen Pheromongrenzen gesetzt. Damit wird die Erkundung neuer Routen zu Beginn des Verfahrens erhöht.

3.1.5 Rang-basiertes Ant System (*AS_{rank}*)

Eine andere Erweiterung des Ant Systems ist *AS_{rank}*, welches 1997 in [24] vorgestellt wurde. Auch hier wird nur die global beste Route zur Pheromonänderung genutzt. Zusätzlich wird eine Reihe von guten Agenten jeder Iteration zum Pheromonupdate herangezogen. Dazu werden die Ameisen, der Länge ihrer Route entsprechend, geordnet. Die global beste Lösung erhält das Gewicht w . Den $(w - 1)$ besten lokalen Routen wird, je nach ihrem Rang r in der Ordnung, ein Gewicht durch $\max\{0, w - r\}$ zugewiesen. Das ganze führt zur Formel

$$\tau_{ij}^{\text{neu}} = (1 - \rho) \cdot \tau_{ij}^{\text{alt}} + \left(\sum_{r=1}^{w-1} (w - r) \cdot \Delta\tau_{ij}^r \right) + w \cdot \Delta\tau_{ij}^{gb}$$

mit $\Delta\tau_{ij}^r = 1/L_r$ und $\Delta\tau_{ij}^{gb} = 1/L_{gb}$. Dabei ist L_r die Länge der Route des r -ten Agenten und L_{gb} die Länge der global besten Route. Mit $w = m$ wurden gute Ergebnisse erzielt.

3.2 Ant Colony Optimierung (ACO)

Die hier vorgestellten Ant Algorithmen folgen einer speziellen Meta-Heuristik (Ant Colony Optimierung), die u.a. 1998 in [16] vorgestellt wurde und sich durch folgende Punkte auszeichnet:

- **Kolonie von kooperierenden Individuen:** Wie in richtigen Ameisenkolonien suchen eine endliche Anzahl an einfachen Agenten nach akzeptablen Lösungen. Durch die im nächsten Punkt besprochenen Mechanismen wird aus diesen Einzellösungen eine „gute“ Gesamtlösung konstruiert.
- **Pheromonausschüttung und Stimulation:** Die Agenten kommunizieren mit ihrer Umgebung durch eine numerische Repräsentation der Pheromonkonzentration. Sie können „Pheromone“ ausschütten bzw. werden durch diese beeinflusst. Dies wird i.A. durch eine Lese- und Schreibmöglichkeit auf die Umgebung (Pheromonfeld) realisiert. Diese Änderungen können bei jedem lokalen Schritt oder zeitversetzt global vorgenommen werden, wenn der Agent eine bestimmte Position erreicht hat. Entsprechend dem chemischen Vorbild spielt in ACO Algorithmen meist auch ein

Flüchtigkeitskoeffizient eine Rolle, um vergangene Ausschüttungen nur für einen gewissen Zeitraum eingehen zu lassen. Die Agenten werden in ihrem Verhalten vom Pheromonfeld gesteuert, was jedoch der einzige Kommunikationskanal zwischen den Individuen ist.

- **Suche nach kurzen Wegen durch lokale Bewegungen:** „Artificial Ants“ und ihre natürlichen Artgenossen haben eine wichtige Gemeinsamkeit. Sie versuchen, kürzeste Wege in der vorgegebenen Problemwelt zu finden. Sie bewegen sich dazu lokal Schritt-für-Schritt durch ihre Umgebung, die natürlich bei den Agenten durch das zu bearbeitende Optimierungsproblem bestimmt wird.
- **Stochastische Übergangscharakteristik:** Die Entscheidung für den folgenden Schritt (Zustandswechsel) wird auf Grundlage einer Wahrscheinlichkeitsverteilung getroffen. Die Individuen benutzen hierzu nur lokale Informationen (Pheromonkonzentration) und ggf. eine Ausgangsheuristik (Entfernung). Bei den virtuellen Ameisen spielen zudem Einschränkungen des Optimierungsproblems eine Rolle, denen mit agenteninternen Datenstrukturen, wie z.B. der Tabu-Liste, Rechnung getragen wird.

ACO Algorithmen sind nun solche Verfahren, welche der ACO Meta-Heuristik genügen und sich insbesondere durch den in [16] gegebenen Pseudoalgorithmus ausdrücken lassen. Insgesamt gehören alle ACO Algorithmen zur Klasse der Ant Algorithmen, was aber umgekehrt nicht immer der Fall sein muß. So wird z.B. in HAS-QAP [26, 27] die Meta-Heuristik insofern verletzt, weil hier die Agenten das Pheromonfeld nicht zur Konstruktion sondern zur Modifikation vorgegebener Lösungen benutzen.

3.3 Konkrete Anwendungen

In diesem Unterabschnitt wollen wir einige konkrete Anwendungen für ACO Algorithmen vorstellen, die einen Bezug zur praktischen Informatik aufzeigen. Insbesondere möchten wir auf ACO-basierte Routing-Verfahren für Kommunikationsnetzwerke eingehen.

3.3.1 Telekommunikation

Erstmals wird in [31, 32] ein ACO Algorithmus für Routing-Aufgaben eingesetzt. Das als *ant-based control* (ABC) bezeichnete Verfahren wurde u.a. auf ein Modell des Telefonnetzes der British Telecom (BT) angewendet. Der Vergleich mit einer herkömmlichen Routing-Methode der BT zeigte, daß ABC in fast allen Situationen eine bessere Performance liefert.

Ein weiterer Ansatz stammt von WHITE, PAGUREK, OPPACHER [33] und hat starke Ähnlichkeit zum Ant System (siehe 3.1.1 auf Seite 6). Die Ausgangsheuristik η ist durch eine lokale Kostenfunktion gegeben. Für jede angenommene Verbindung werden vom Quellknoten aus Ameisen gestartet, die nach einem „billigen“ Pfad zum Ziel suchen. Auf ihrem Rückweg ändern sie, in Abhängigkeit der auf dem Hinweg berechneten Kosten, die Pheromonkonzentration ab. Die entgültige Route wird dann durch den Weg bestimmt, den die meisten Agenten benutzt haben. Im Fall von Netzwerkfehlern wird noch während der aufgebauten Verbindung eine erneute Suche vorgenommen. Der optimale Wert der Parameter α und β wird mit Hilfe eines genetischen Algorithmus [34] ermittelt, weshalb das gesamte Verfahren ASGA genannt wird.

Im Gegensatz zu den oben diskutierten *verbindungsorientierten* Lösungen wurden von DI CARO und DORIGO [35, 36, 37] mehrere Versionen des AntNet Algorithmus für *verbindungslose* Kommunikationsnetze vorgestellt. In [38] wird das Verfahren mit einigen *state-of-the-art* Technologien verglichen und zeigt insgesamt ein sehr robustes Verhalten mit guter und vor allem stabiler Durchsatzrate. AntNet-FA [39, 40]

und AntNet-FS sind Erweiterungen des Basisalgorithmus für spezielle Anwendungsbereiche.

Eine Verbesserung von ABC für verbindungslose Kommunikation wurde u.a. von VAN DER PUT, ROTHKRANTZ 1998 vorgenommen [41, 42] und an einem Fax-Verteilungs-Problem der niederländischen KPN Telecom getestet.

3.3.2 Optische Netzwerke (VWP-Routing)

Etwas genauer möchten wir auf eine Anwendung im Bereich Optische Netzwerke eingehen. Grundlage unserer Betrachtungen ist das in [43] besprochene Verfahren:

Optische Informationsübertragung: Hochgeschwindigkeitsnetze basieren i.A. auf optischen Verbindungen, welche in der Lage sind, Datenströme mit Geschwindigkeiten von 10 Gbit/s oder mehr zu übertragen. Jede dieser Verbindungen besteht aus einer Anzahl von Glasfasern, die wiederum Signale auf unterschiedlichen Wellenlängen transportieren können. In solchen Netzwerken gibt es zwei Routing-Ansätze:

1. WP (wavelength path): Hier erhält eine End-zu-End-Verbindung auf allen Teilstrecken dieselbe Wellenlänge. Damit kann jedoch nicht die vollständige Bandbreite des Netzwerkes genutzt werden, weil aus technischen/Kostengründen nur endlich viele Wellenlängen zur Verfügung stehen.
2. VWP (virtual wavelength path): Die End-zu-End-Verbindung kann auf Teilstrecken unterschiedliche Wellenlängen besitzen, so daß ggf. eine Konvertierung an den Knoten notwendig ist. Dadurch ergibt sich aber auch eine bessere Auslastung des Gesamtnetzes.

Für ein vorgegebenes Netzwerk besteht nun die Aufgabe, die Anzahl der verwendeten Wellenlängen (NWR, network wavelength requirement) beim VWP-Routing zu minimieren und gleichzeitig die Gesamtlänge der Route klein zu halten. Zur Vereinfachung wollen wir hier nur Verbindungen mit einer Glasfaser betrachten. Abbildung 5 zeigt ein kleines Beispielnetzwerk aus [43] mit 4 End-zu-End-Verbindungen.

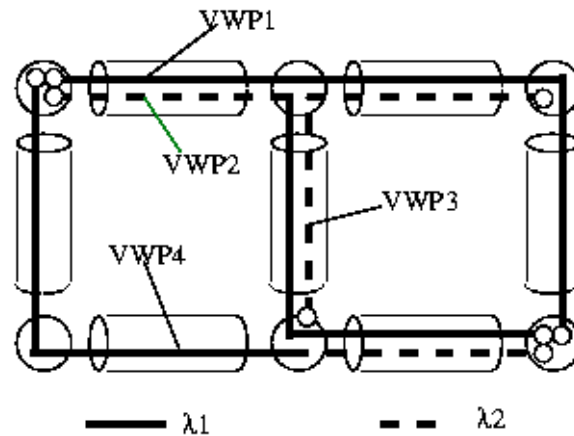


Abbildung 5 (aus [43]): VWP-Routing von vier End-zu-End-Verbindungen

Der NWR-Wert des gesamten Netzwerkes ist das Maximum der NWR's aller Einzelstrecken.

Anwendung eines ACO Algorithmus: VARELA et al. benutzen in [43] einen Ant Algorithmus für das VWP-Routing. Die Problemstruktur weicht aber in einigen Punkten vom schon betrachteten Traveling Salesman Problem ab:

- Nicht alle Knoten sind miteinander verbunden. Die Gestalt des Graphen hängt von der Topologie des Optischen Netzwerkes ab.

- Bei den Lösungen handelt es sich um End-zu-End-Verbindungen, die keinen Kreis im Graphen bilden. Es gibt also eine Quelle und ein Ziel der Verbindung.
- Durch die beiden vorherigen Punkte kann die Ameise in „Sackgassen“ geraten. Deshalb muß sie auch in der Lage sein, einen lokalen Schritt rückgängig zu machen (Backtracking).
- Je End-zu-End-Verbindung kommt ein anderer Agenttyp mit spezieller Pheromon-signatur zum Einsatz. Um die Mehr-Kanal-Situation auf den Teilstrecken zu modellieren, werden Ameisen nur von ihren eigenen Pheromonen angezogen. Fremde Pheromontypen wirken abstoßend.

Es wurden drei Varianten des Algorithmus entwickelt: LU (lokale Pheromonänderung), GU/D (globale Pheromonänderung, Bewertung nach Distanz) und GU/O (globale Pheromonänderung, Bewertung nach Belegung). In jedem Fall werden pro Iteration Ameisen von der Quelle zum Ziel „gesendet“. Die Auswahlwahrscheinlichkeit des nächsten Knotens wird von der jeweiligen Variante mittels Formel bestimmt. Am Ziel, das die Ameisen durch die Problemgestalt nicht zeitgleich erreichen müssen, werden sie zerstört und eine neue Iteration beginnt. Die Pheromonänderung kann lokal (nach jedem Schritt) oder global (nach jeder Iteration) erfolgen. Falls es zum Backtracking kommt, müssen ggf. vorher durchgeführte Änderungen rückgängig gemacht werden. Nach jeder Iteration wird durch den Flüchtigkeitkoeffizient eine gewisse Menge aller Pheromontypen von den Verbindungsstrecken entfernt. Außerdem werden alle *stochastisch* bestimmten Routen, d.h. solche auf denen sich Agenten bewegt haben, und alle *deterministisch*⁶ errechenbaren Lösungen zur Wellenlängenbelegung herangezogen. Alle Routen durchs Netzwerk werden zur späteren Ausgabe gespeichert, wenn ihr NWR-Wert besser ist als der des vorherigen Durchlaufs. Die Auswahlwahrscheinlichkeit wird durch folgende Formeln bestimmt. Beim Anziehungsgewicht der Ameise vom Typ j auf dem k -ten Teilstück

$$\alpha_{kj} = \frac{\psi_{kj}}{\sum_{i \in A_j} \psi_{ij}} \quad (7)$$

ist A_j die Menge der Verbindungen, die diese Ameise noch besuchen darf und $\psi_{?j}$ die Pheromonkonzentration mit Signatur j auf den fraglichen Abschnitten. Das Abstoßungsgewicht ergibt sich dann analog durch

$$\beta_{kj} = \frac{\sum_{h \neq j} \psi_{kh}}{\sum_{i \in A_j} \sum_{h \neq j} \psi_{ih}} \quad , \quad (8)$$

was uns insgesamt zur Auswahlwahrscheinlichkeit

$$p_{kj} = \frac{\alpha_{kj}/(\beta_{kj})^\varepsilon}{\sum_{i \in A_j} \alpha_{ij}/(\beta_{ij})^\varepsilon} \quad (9)$$

führt. Der Parameter ε steuert die relative Abhängigkeit beider Gewichte. In der LU Variante des Algorithmus wird die lokale Pheromonänderung (vgl. Ant System, Ant-density) durch $\psi_{kj}^{\text{neu}} = \psi_{kj}^{\text{alt}} + Q$ und die globale Änderung mit $\forall j \forall k : \psi_{kj}^{\text{neu}} = \rho \cdot \psi_{kj}^{\text{alt}}$ beschrieben, wobei ρ der Flüchtigkeitkoeffizient und Q die Pheromongrundmenge jeder Ameise ist.

Beim GU/D gibt es kein lokales Update; das Pheromonfeld ψ wird nach jeder Iteration in Abhängigkeit von der Länge der Route L_j verändert (vgl. Ant System, Ant-cycle):

$$\forall j \forall k : \psi_{kj}^{\text{neu}} = \psi_{kj}^{\text{alt}} + \frac{Q}{L_j}$$

In der bevorzugten Variante GU/O wird das Verfahren um eine zweite Steuergröße u_k erweitert. Diese gibt an, von wievielen Agenten das k -te Teilstück in der aktuellen Iteration (und allen bisherigen, respektive Flüchtigkeitkoeffizient) benutzt wurde. u_k wird also von jedem Agenten, der diese Teilstrecke benutzt, inkrementiert und nach jeder Iteration durch $\forall k : u_k^{\text{neu}} = \rho \cdot u_k^{\text{alt}}$ geändert. Außerdem wurde das Abstoßungsgewicht, vgl. (8), ersetzt:

$$\beta_{kj} = \frac{u_k}{\sum_{i \in A_j} u_i} \quad (10)$$

⁶Bei deterministischer Berechnung wird von der Ameise immer der Knoten gewählt, welcher die höchste Wahrscheinlichkeit besitzt. Der Zufall spielt somit keine Rolle mehr.

Dadurch wird das Verhalten des Agenten nicht nur von anderen Ameisen beeinflusst, sondern in gewisser Weise auch durch eigene Aktionen.

Experimente und Ergebnisse: Alle drei Varianten wurden in [43] anhand von Testnetzen mit der Nagatsu-Heuristik [44] verglichen. Die verwendeten Parameter waren:

- ▶ LU: $\rho = 0.5, Q = 1, \varepsilon = 2, T = 30$
- ▶ GU/D: $\rho = 0.5, Q = 5, \varepsilon = 5, T = 30$
- ▶ GU/O: $\rho = 0.9, Q = 0.5, \varepsilon = 5, T = 250$

Das erste Testnetz hatte vier Knoten, die über eine $100 \text{ km} \times 100 \text{ km}$ große Fläche verstreut waren. Der Gesamtdurchsatz von 150 Gbit/s verteilte sich auf 20 End-zu-End-Verbindungen. Im dritten Szenario, das aus 15 Knoten auf einer $1000 \text{ km} \times 1000 \text{ km}$ Fläche bestand, waren 1500 Gbit/s Durchsatz in 268 End-zu-End-Verbindungen zu bewältigen. Folgende NWR-Werte wurden erreicht:

Netzwerk	Knoten	Nagatsu	LU	GU/D	GU/O
1	4	5	7	5	5
2	9	11	27	17	12
3	15	13	50	32	15

Tabelle 1: NWR-Werte aus [43] für die Nagatsu-Heuristik und die drei Varianten des ACO Algorithmus

Nur die GU/O Variante reicht in der Qualität ihrer Lösungen an die Nagatsu-Heuristik heran. Leider hat der ACO Algorithmus auch eine recht lange Laufzeit von 2 Stunden und 20 Minuten, was aber auch am Einsatz von Sun's JIT Java 1.1.3 Compiler liegen könnte. Die Nagatsu-Heuristik war auf der selben Plattform dagegen schon in 20 Sekunden fertig.

4 Recruiting Ant System (RAS)

In diesem Abschnitt soll noch kurz das *Recruiting Ant Systems (RAS)* vorgestellt werden, was - grob gesprochen - das Ant System (AS) um eine andere Rekrutierungsstrategie erweitert. Alle im folgenden beschriebenen Experimente und Ergebnisse sind, mit Blick auf die gerade erst begonnene Forschung, vorsichtig zu betrachten.

Idee: SUMPTER und BEEKMAN äußern in [8] die Vermutung, daß für mittlere Koloniegroßen eine Kombination von Massenrekrutierung und Gruppenrekrutierung nützlich sein könnte. Daß vorrangige Massenrekrutierung durch Pheromonausschüttung nicht immer zum Erfolg führt, zeigen Experimente mit kleinen bis mittleren Pharao-Ameisenkolonien. Grundlage von RAS ist also neben der Pheromonsteuerung eine zweite Rekrutierungsstrategie, die Ähnlichkeiten zum Tandemlauf (siehe unter **Rekrutierung** auf Seite 5) der Ameisen aufweist. Deshalb existieren im Algorithmus zwei Typen von Agenten:

1. **Suchameisen** haben primär die Aufgabe, die Umgebung nach Futter zu durchstreifen. Für sie ist die Auswahlsteuerung von Ant System geradezu ideal.
2. **Transportameisen** warten im Nest bis eine akzeptable Futterquelle gefunden ist und helfen dann beim Transport der Nahrung. Sie folgen dabei mit einer hohen Wahrscheinlichkeit der Route einer Suchameise. Wenn sie die Spur verlieren sollten, orientieren sie sich am Pheromonfeld. Die Zahl der Transportameisen übersteigt die Anzahl der Suchameisen erheblich.

Algorithmus: Hier nun der Pseudo-Quelltext für das Traveling Salesman Problem:

```

Initialisierung
FOR  $t = 1$  TO Anzahl der Iterationen des Algorithmus
    FOR  $k = 1$  TO  $m$ 
        Wiederhole, solange  $k$ -te Suchameise noch Städte besuchen kann:
            Wähle mit Wahrscheinlichkeit  $p_{ij}$  nach Formel (1) eine Stadt  $j$  aus, die als
            nächste besucht werden soll.
            Berechne die Gesamtlänge  $L_k$ , die von der  $k$ -ten Suchameise zurückgelegt wurde.
        Setze die neue Pheromonintensität  $\tau_{ij}$  für alle Verbindungen aus  $\mathcal{G}$ .
        Rekrutiere jeweils  $r1$  Transportameisen für die  $r0$  „besten“ Suchameisen.
    FOR  $k = 1$  TO  $r0 \cdot r1$ 
        Wähle einen beliebigen Startpunkt.
        Wiederhole, solange  $k$ -te Transportameise noch Städte besuchen kann:
            Wähle mit Wahrscheinlichkeit  $1 - \xi$  die vorgegebene Route, sonst wähle nach
            Formel (1) eine Stadt  $j$  aus, die als nächste besucht werden soll und fahre im
            weiteren Verlauf immer gemäß (1) fort.
            Berechne die Gesamtlänge  $L_k$ , die von der  $k$ -ten Transportameise zurückgelegt
            wurde.
        Setze die neue Pheromonintensität  $\tau_{ij}$  für alle Verbindungen aus  $\mathcal{G}$ .
        Lasse alle  $r0 \cdot r1$  Transportameisen zum Nest zurückkehren.

```

Experimente und Resultate: Die Ergebnisse von RAS sind vielversprechend. Tabelle 2 zeigt die durchschnittliche Lösungsqualität von RAS und AS im Vergleich (jeweils 20 Läufe). Dabei wurde $r0 = 5$ und $r1 = m$ gewählt. Die Implementierung erfolgte in C++.

Algorithmus	$T = 10$	$T = 50$	$T = 100$
AS	727.76555995	711.8184969	706.46014185
RAS mit $\xi = 0.00$	730.24816355	703.7579053	701.59558885
RAS mit $\xi = 0.05$	718.532855762	693.7587304	689.5299028
RAS mit $\xi = 0.15$	720.77685185	705.50525925	700.85882025

Tabelle 2: Mittelwert der global besten Lösungen mit $\alpha = 1.0, \beta = 5.0, \rho = 0.5, Q = 100.0, m = 70$ (RAS), $m = 6 \cdot 70$ (AS) beim 70-Städteproblem Smith/Thompson (Optimum = 675)

In einer späteren Arbeit werden wir das Verhalten von RAS genauer untersuchen.

5 Zusammenfassung und Ausblick

In dieser Arbeit haben wir uns mit Ant Algorithmen und deren biologischen Vorbildern beschäftigt. Ausgehend von der Diskussion des Ant Systems (AS) mit Fragen der Übertragbarkeit, Komplexität und Parallelisierbarkeit wurden mehrere Verbesserungen (Ant-Q, ACS, \mathcal{M} MAS, AS_{rank}) dieses Basisalgorithmus besprochen. Nach kurzer Vorstellung der ACO Meta-Heuristik verließen wir die abstrakte Welt der kombinatorischen Probleme und gingen zu ganz konkreten Anwendungen wie Telekommunikation und Routing in Optischen Netzwerken über. Den Abschluß bildete die Vorstellung eigener Verbesserungen des Ant Systems auf Grundlage einer weiteren Rekrutierungsstrategie, welche uns zum Recruiting Ant System (RAS) geführt hat. Hier sind noch genauere Betrachtungen notwendig, die in einer späteren Arbeit nachgereicht werden.

Abschließend möchten wir einen kleinen Ausblick in die Zukunft geben: Die vorgestellten Ant Algorithmen gehören in den Bereich *Swarm Intelligence* [45], das sich mit dem Design von Algorithmen oder dem Entwurf verteilter Problemlösungen auf Basis kollektiver Verhaltensmuster von Insektenkolonien beschäftigt. Die Kommunikation der Individuen durch gegenseitige Beeinflussung (Stigmergy Effekt) ist dabei

ein wichtiges Mittel mit guter Erfolgsaussicht. Dieser Effekt kann in der Informatik vielseitig angewendet werden; u.a. ist er eine Grundlage für das Simulationssystem *Swarm* [46]. In Zukunft könnte dieser Begriff aber durch Beispiele wie agentenbasierte Administration, intelligente Clustering Verfahren oder Kontrollstrategien für Gruppen von Transportrobotern noch größere Bedeutung erlangen.

Literatur

- [1] R. Gärtner, H. Küstner, D. Linke, G. Wolf: *Kleine Enzyklopädie Natur*
VEB Bibliographisches Institut Leipzig, 1987, ISBN 3-323-00110-9
- [2] David McFarland, Thomad Bösser: *Intelligent Behavior in Animals and Robots*
A Bradford Book, MIT Press 1993, ISBN 0-262-13293-1
- [3] David J. T. Sumpter:
From Bee to Society: An Agent-based Investigation of Honey Bee Colonies
University of Manchester, PhD thesis, März 2000
- [4] David J. T. Sumpter, D. S. Broomhead:
Formalising the Link between Worker and Society in Honey Bee Colonies
Lecture Notes on Artificial Intelligence, **Vol. 1534**, 95-110, 1998
- [5] David J. T. Sumpter, D. S. Broomhead:
Shape and Dynamics of Thermoregulating Honey Bee Clusters
Journal of Theoretical Biology, **204**, 1-14, 2000
- [6] B. Heinrich:
The mechanisms and energetics of honeybee swarm temperature regulation
Journal of Exp. Biology, **91** 25-55, 1981
- [7] Horst Dehmer: *Spezialisierung von Individuen einer Art am Beispiel der Ameisen*
Technische Universität Darmstadt, Seminararbeit, 1998
- [8] D. Sumpter, M. Beekman: *How many agents make a working whole ?*
Oxford University, <http://www.maths.ox.ac.uk/~sumpter>
- [9] M. Dorigo, V. Maniezzo, A. Coloni: *Positive Feedback as a Search Strategy*
Dip. Elettronica, Politecnico di Milano, Technical Report 91-016, 1991
- [10] M. Dorigo, V. Maniezzo, A. Coloni:
Ant System: An Autocatalytic Optimizing Process
Dip. Elettronica, Politecnico di Milano, Technical Report 91-016REV, 1991
- [11] M. Dorigo: *Optimization, Learning and Natural Algorithms*
Dip. Elettronica, Politecnico di Milano, PhD thesis, 1992
- [12] M. Dorigo, V. Maniezzo, A. Coloni:
The Ant System: Optimization by a Colony of Cooperating Agents
IEEE Transaction on Systems, Man and Cybernetics - Part B, **26(1)**, 29-42, 1996
- [13] B. Bullnheimer, R. F. Hartl, C. Strauss:
Applying the ant system to the vehicle routing problem
Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization,
Kluwer Academics, 109-120, 1998
- [14] D. Costa, A. Hertz: *Ants can colour graphs*
Journal of the Operational Research Society, **48**, 295-305, 1997
- [15] R. Michel, M. Middendorf:
An ACO algorithm for the shortest common supersequence problem
New Methods in Optimization, McGraw-Hill, 1999
- [16] G. Di Caro, M. Dorigo, L. M. Gambardella:
Ant Algorithms for Discrete Optimization
Technical Report 98-10, IRIDIA, Université Libre de Bruxelles, 1998
- [17] B. Bullnheimer, G. Kotsis, C. Strauss: *Parallelization Strategies for the Ant System*
Technical Report POM 9-97, Inst. of Management Sci., University of Vienna, 1997

- [18] M. Dorigo, L. M. Gambardella:
Ant-Q: A Reinforcement Learning approach to the traveling salesman problem
Proceedings of MachineLearning-95, Morgan Kaufmann, 252-260, 1995
- [19] M. Dorigo, L. M. Gambardella: *A study of some properties of Ant-Q*
Technical Report 96-4, IRIDIA, Université Libre de Bruxelles, 1994
- [20] M. Dorigo, L. M. Gambardella: *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*
Technical Report 96-5, IRIDIA, Université Libre de Bruxelles, 1996
- [21] T. Stützle, H. H. Hoos:
The MAX-MIN Ant System and Local Search for the Traveling Salesman Problem
Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC'97), 309-314, 1997
- [22] T. Stützle, H. H. Hoos:
Improvements on the Ant System: Introducing the MAX-MIN Ant System
Artificial Neural Networks and Genetic Algorithms, Springer Verlag, 245-249, 1998
- [23] T. Stützle: *Local Search Algorithms for Combinatorial Problems - Analysis, Improvements and New Applications*
Darmstadt University of Technology, PhD thesis, 1998
- [24] B. Bullnheimer, R. F. Hartl, C. Strauss:
A New Rank Based Version of the Ant System - A Computational Study
Technical Report POM 3-97, Inst. of Management Sci., University of Vienna, 1997
- [25] M. Dorigo, L. M. Gambardella: *Ant colonies for the traveling salesman problem*
Technical Report 96-3, IRIDIA, Université Libre de Bruxelles, 1996
- [26] M. Dorigo, L. M. Gambardella, E. D. Taillard: *Ant colonies for the QAP*
Technical Report 4-97, IDSIA Lugano, Switzerland, 1997
- [27] M. Dorigo, L. M. Gambardella, E. D. Taillard: *Ant colonies for the QAP*
Journal of the Operational Research Society (JORS), **50(2)**, 167-176, 1999
- [28] C. J. C. H. Watkins, P. Dayan: *Q-Learning*
Machine Learning, **8**, 279-292, 1992
- [29] A. Coloni, M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini, M. Trubian:
Heuristics from Nature for Hard Combinatorial Optimization Problems
International Transactions in Operational Research, **3(1)**, 1-21, 1996
- [30] E. Talbi, O. Roux, C. Fonlupt, D. Robillard:
Parallel Ant Colonies for Combinatorial Optimization Problems
ANTS'98 First International Workshop on Ant Colony Optimization, 1998
- [31] R. Schoonderwoerd, O. Holland, J. Bruten, L. Rothkrantz:
Ant-based load balancing in telecommunications networks
Adaptive Behavior, **5(2)**, 169-207, 1996
- [32] R. Schoonderwoerd, O. Holland, J. Bruten:
Ant-like agents for load balancing in telecommunications networks
Proc. of the First Int. Conference on Autonomous Agents, ACM Press, 209-216, 1997
- [33] T. White, B. Pagurek, F. Oppacher:
Connection management using adaptive mobile agents
Proc. of the Int. Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'98), CSREA Press, 802-809, 1998
- [34] D. E. Goldberg: *Genetic Algorithms in Search, Optimization and Machine Learning*
Addison-Wesley, 1989
- [35] G. Di Caro, M. Dorigo: *AntNet: A mobile agents approach to adaptive routing*
Technical Report 97-12, IRIDIA, Université Libre de Bruxelles, 1997
- [36] G. Di Caro, M. Dorigo:
Ant colonies for adaptive routing in packet-switched communications networks
Proc. of the Fifth Int. Conference on Parallel Problem Solving from Nature, Springer-Verlag, 673-682, 1998

- [37] G. Di Caro, M. Dorigo:
An adaptive multi-agent routing algorithm inspired by ants behavior
Proc. of the Fifth Annual Australasian Conference on Parallel and Real-Time Systems, Springer-Verlag, 261-272, 1998
- [38] G. Di Caro, M. Dorigo: *Mobile Agents for Adaptive Routing*
Proceedings of 31st Hawaii International Conference on Systems Sciences, 1998
- [39] G. Di Caro, M. Dorigo:
Extending AntNet for best-effort Quality-of-Service routing
ANTS'98 First International Workshop on Ant Colony Optimization, 1998
- [40] G. Di Caro, M. Dorigo:
Two ant colony algorithms for best-effort routing in datagram networks
Proceedings of the Tenth International Conference on Parallel and Distributed Computing and Systems (PDCS'98), IASTED/ACTA Press, 541-546, 1998
- [41] R. van der Put: *Routing in the faxfactory using mobile agents*
Technical Report R&D-SV-98-276, KPN Research, 1998
- [42] R. van der Put, L. Rothkrantz: *Routing in packet switched networks using agents*
Simulation Practice and Theory, in press, 1999
- [43] G. N. Varela, M. C. Sinclair: *Ant Colony Optimisation for Virtual-Wavelength-Path Routing and Wavelength Allocation*
Proceedings of the Congress on Evolutionary Computation (CEC'99), 1999
- [44] N. Nagatsu, Y. Hamazumi, K. I. Sato:
Number of wavelengths required for constructing large-scale optical path networks
Electronics and Communications in Japan, Part I, **78(9)**, 1-11, 1995
- [45] E. Bonabeau, M. Dorigo, G. Théraulaz: *From Nature to Artificial Swarm Intelligence*
Oxford University Press, 1999
- [46] C. Langton, N. Minar, R. Burkhart:
The Swarm Simulation System: A tool for studying complex systems
Swarm Development Group, <http://www.swarm.org>