

Ameisenalgorithmen

—

Optimierung und Modellierung

Zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften

(Dr. rer. pol.)

von der Fakultät für Wirtschaftswissenschaften
der Universität Fridericiana zu Karlsruhe
genehmigte Dissertation

von

Dipl.-Inform. Daniel Merkle

Referent:	Prof. Dr. H. Schmeck
Korreferenten:	Prof. Dr. M. Middendorf Prof. Dr. K.-H. Waldmann
Tag der mündlichen Prüfung:	25. Juli 2002

I believe there is no philosophical high-road in science, with epistemological signposts. No, we are in a jungle and find our way by trial and error, building our road behind us as we proceed.

(Max Born)

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter des Lehrstuhls von Prof. Dr. H. Schmeck am Institut für Angewandte Informatik und Formale Beschreibungsverfahren der Universität Fridericiana zu Karlsruhe. Ihm gilt mein Dank für die Betreuung und Begutachtung der Dissertation und für die kreative Freiheit bei der Wahl und Durchführung dieser Arbeit. Besonders bedanke ich mich bei Ihm auch dafür, dass er mir in einem sehr fruchtbaren Arbeitsumfeld die Möglichkeit zur Promotion gegeben hat. Mein ganz besonderer Dank gilt Prof. Dr. M. Middendorf für die hervorragende Zusammenarbeit, die unzähligen fachlichen Diskussionen, für seine nicht versiegende Motivation, für die Übernahme des Korreferats, und vor allem dafür, dass er mich gelehrt hat, wieviel Spaß wissenschaftliches Arbeiten machen kann. Herrn Prof. Dr. K.-H. Waldmann danke ich für die freundliche Übernahme des Korreferats, sein Interesse an der Arbeit und deren rasche Begutachtung.

Frau Dr. Kathrin Paschen und Herrn Dr. Thomas Worsch danke ich für die zahlreichen wissenschaftlich sehr anregenden Gespräche, die einen des Denkens niemals müde werden lassen, und wieder und wieder offenbaren, wie man durch kritisches Hinterfragen zu tieferem Verständnis gelangt.

Für die sehr angenehme Arbeitsatmosphäre am Institut und die ständige Bereitschaft insbesondere fachliche Diskussionen zu führen, danke ich stellvertretend meinen ehemaligen Kollegen Dr. Jürgen Branke, Michael Guntsch und Andreas Hotho, die zudem durch sehr wertvolle Vorschläge während der Korrekturphase die Qualität der Arbeit verbessert haben. Auch bei Maria Wünstel bedanke ich mich herzlich dafür, dass Sie die vorliegende Arbeit komplett Korrektur gelesen hat.

Nicht zuletzt gilt mein ganz besonderer Dank Stefanie Buchert, zum einen dafür, dass Sie ständig neu überarbeitete Fassungen dieser Arbeit mit Ihrem kritischen Auge begutachtet hat, aber noch viel mehr weil Sie mit viel Verständnis immer Halt und Ruhepol für mich war.

Kurzfassung

Ameisenalgorithmen gehören zur Klasse der populationsbasierten zufallsgeleiteten Suchverfahren für die Lösung kombinatorischer Optimierungsprobleme und sind inspiriert von Vorgehensweisen, die Ameisenkolonien bei der Suche nach Nahrung verwenden. Durch indirekte Kommunikation mittels Duftstoffmarkierung gelangen die Ameisen in einem Prozess der Wegoptimierung schnell zu Nahrungsquellen. Im algorithmischen Analogon wählen künstliche Ameisen im Entscheidungsgraphen des zugrunde liegenden Optimierungsproblems iterativ Wege, die zunehmend besseren Lösungen entsprechen.

In dieser Arbeit werden gezielt Verfahrensmodifikationen und Erweiterungen der herkömmlichen Ameisenalgorithmen vorgestellt, die für bestimmte Problemklassen zu einem deutlich besseren Optimierungsverhalten führen. Insbesondere betrifft dies die Auswertung der gespeicherten Duftstoffinformation und die Entscheidungsreihenfolgen, die eine künstliche Ameise beim Aufbau einer Lösung durchläuft. Die erfolgreiche Anwendung der entwickelten Methoden wird jeweils anhand einfacher Probleme gezeigt, die charakteristische Eigenschaften schwieriger Probleme widerspiegeln, und anschließend auf NP-vollständige Probleme übertragen. Die Betrachtung des Problems der Projektplanung bei beschränkten Ressourcen bezieht empirische Tests der Methoden an Probleminstanzen einer bekannten Benchmark-Bibliothek ein. In den Tests wurden die oberen Schranken für viele Instanzen verbessert und im Vergleich zu anderen Verfahren im Durchschnitt die besten Ergebnisse erzielt. Probleme der Maschinenbelegung bei Beachtung mehrerer konkurrierender Zielkriterien werden durch die Entwicklung von Kooperationsmechanismen zwischen mehreren Kolonien von künstlichen Ameisen erfolgreich gelöst.

Es wird ein deterministisches Modell für Ameisenalgorithmen vorgestellt, das auf den erwarteten Aktualisierungswerten der Duftstoffinformation basiert. Anhand dieses Modells können mit Hilfe einer Fixpunktanalyse der in einer Matrix gespeicherten Duftstoffinformation Aussagen über die Dynamik von Ameisenverfahren für Permutationsprobleme getroffen werden. Das Modellverhalten entspricht in vieler Hinsicht sehr gut dem zufallsgeleiteten Ameisenalgorithmus und ermöglicht auch für komplexe Optimierungsprozesse analytische Vorhersagen.

Inhaltsverzeichnis

Kurzfassung	v
1 Einleitung	1
1.1 Zusammenhang mit der Biologie	1
1.2 Ameisenalgorithmus	4
1.3 Vergleich künstlicher und realer Ameisen	7
1.4 Erweiterungen	8
1.5 Anwendungen	10
1.6 Theorie	11
1.7 Gliederung der Arbeit	13
2 Ameisenalgorithmen und einfache Permutationsprobleme	15
2.1 Motivation	15
2.2 Definition einfacher Probleme	16
2.3 Die Problem Instanz P_A	16
2.4 Verzerrung der Auswahlwahrscheinlichkeiten	19
2.5 Zusammenfassung	22
3 Globale Pheromonauswertung	25
3.1 Summen-Pheromonauswertung	26
3.1.1 Motivation und Vorgehensweise	26
3.1.2 Einfache Problem Instanzen	30
3.1.3 Ein NP-vollständiges Problem	33
3.2 Relative Pheromonauswertung	38
3.2.1 Motivation und Vorgehensweise	39
3.2.2 Einfache Problem Instanzen	43
3.2.3 Ein NP-vollständiges Problem	47
3.3 Zusammenfassung	51

4	Entscheidungsreihenfolgen	53
4.1	Motivation und Vorgehensweise	53
4.2	Einfache Probleminstanzen	59
4.2.1	Die Probleminstanzen P_A und P_{MF}	59
4.2.2	Die Probleminstanz P_{MA}	61
4.2.3	Die Probleminstanzen P_{P_k}	64
4.3	Ein NP-vollständiges Problem	67
4.3.1	Heuristik für SMTWDP	69
4.3.2	Probleminstanzen und Parameter	71
4.3.3	Ergebnisse	72
4.4	Zusammenfassung	74
5	Projektplanung unter beschränkten Ressourcen	77
5.1	Definitionen	78
5.2	Prioritätsregelverfahren	79
5.3	Ameisenalgorithmus für RCPSP	82
5.3.1	Prioritätsregeln	84
5.3.2	Zusätzliche Erweiterungen	86
5.4	Benchmark-Probleme und Parameter	92
5.5	Ergebnisse	94
5.5.1	s-ACO-RCPSP	94
5.5.2	ACO-RCPSP	98
5.6	Zusammenfassung	109
6	Bikriterielle Optimierung	111
6.1	Motivation und Definitionen	112
6.2	SMTTP mit Umrüstkosten	115
6.3	Ameisenalgorithmen für bikriterielle Optimierungsprobleme . .	116
6.3.1	Eine Kolonie	116
6.3.2	Mehrere Kolonien	118
6.4	Probleminstanzen und Parameter	121
6.5	Ergebnisse	122
6.6	Zusammenfassung	128
7	Modellierung	129
7.1	Einfache Probleme	131
7.2	ACO-Modell	131
7.3	Eingeschränkte Permutationsprobleme	137
7.4	Homogene Permutationsprobleme	143

7.4.1	ACO-Modell	143
7.4.2	Beispiel	146
7.5	Fixpunktanalyse	154
7.6	Dynamisches Verhalten	160
7.7	Simulationen	170
7.8	Heterogene Permutationsprobleme	176
7.8.1	ACO-Modell	176
7.8.2	Dynamisches Verhalten und Simulationen	179
7.8.3	Skalierung und Maskierung	183
7.9	Zusammenfassung	187
8	Zusammenfassung und Ausblick	189
	Literaturverzeichnis	193
	Index	209

Kapitel 1

Einleitung

1.1 Zusammenhang mit der Biologie

Ameisenstaaten, Bienen- und Vogelschwärme sind Beispiele für biologische Systeme, denen wir aufgrund des kollektiven Verhaltens der Individuen eine Schwarmintelligenz zuordnen können. Der Begriff Schwarmintelligenz (swarm intelligence) wurde erstmals in (Beni und Wang, 1989; Beni, 1988) im Bereich der Robotikforschung eingeführt und definiert. Die Definition wurde in (Bonabeau et al., 1999) derart erweitert, dass allgemein jedem Algorithmus oder jedem verteilten problemlösungsorientierten System, welches sich am Verhalten sozialer Insekten oder anderer Tiergesellschaften orientiert, die Eigenschaft der Schwarmintelligenz zugeordnet wird. Ameisenalgorithmen sind eines der Beispiele für erfolgreiche Systeme, in denen das kollektive Verhalten vieler, auf einfachen Regeln beruhender Agenten zu einer funktional zusammenhängenden intelligenten Methodik für die Optimierung von kombinatorischen Problemen führt. Unter sozialen Insekten (oder auch Staaten bildenden Insekten) verstehen wir solche, die in selbst organisierten Gruppen agieren. Sehr starkes Forschungsinteresse wurde seit der Entstehung der Forschungsrichtung Soziobiologie auf die Unterklasse der eusozialen Insekten gerichtet. Insekten mit sich überlappenden Generationen, kooperativer Brutpflege und dem Merkmal der Arbeitsaufteilung werden als eusozial definiert. In die Klasse der eusozialen Insekten fallen neben Spezies von Bienen und Wespen auch alle Spezies von Termiten und Ameisen. Die Kommunikation der sich selbst organisierenden eusozialen Insekten kann direkt oder indirekt sein. Wir sprechen von indirekter Kommunikation, wenn ein Individuum Einfluss auf die Umgebung nimmt und die dadurch veränderte Umgebung das Verhalten anderer Individuen beeinflusst. Das von Ameisenalgorithmen genutzte Prinzip der indirekten

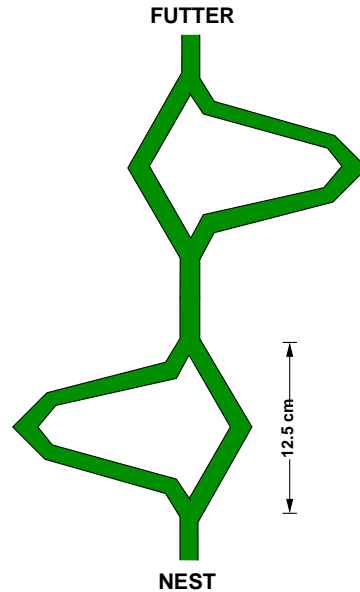


Abbildung 1.1: Darstellung des Doppelbrückenexperiments nach (Goss et al., 1989)

Kommunikation bei sozialen Insekten (stigmergy) wurde schon in den fünfziger Jahren unter anderem in (Grassé, 1959) untersucht. Von besonderem Interesse ist das Verhalten von Ameisen bei der Wegfindung zur Futtersuche, das experimentell unter anderem in (Goss et al., 1989) und (Deneubourg et al., 1990) analysiert wurde. Bei dem in (Goss et al., 1989) durchgeführten so genannten Doppelbrückenexperiment mit argentinischen Ameisen (*Linepithema humile*) finden diese, mit Hilfe der indirekten Kommunikation durch Duftstoffe (Pheromone¹), einen kürzesten Weg vom Nest zur Futterquelle.

Aufgrund der Bedeutung für Ameisenalgorithmen werden wir im Folgenden das Doppelbrückenexperiment näher erläutern. Abbildung 1.1 zeigt den Versuchsaufbau, bei dem das Nest einer Ameisenkolonie über zwei Doppelbrücken mit einer Futterquelle verbunden wurde. Die Ameisen müssen sich auf dem Weg zur Futterquelle zweimal zwischen einem kurzen und einem langen Teilpfad entscheiden, wobei die langen Teilpfade jeweils r mal länger als die kurzen Teilpfade sind. Ziel des Doppelbrückenexperiments war die Feststellung, ob die Ameisen in der Lage sind, mit hoher Wahrscheinlichkeit den kürzeren Weg zu wählen. Für $r = 2$ wurde statistisch signifikant gezeigt, dass

¹Der aus dem Griechischen stammende Begriff Pheromon wurde durch die Arbeit von (Karlsön und Butenandt, 1959) geprägt (pherein=übertragen, horman=erregen).

die Ameisen den kurzen Weg bevorzugen. In einem modifizierten Versuch hatten die Ameisen in den ersten 30 Minuten des Experiments nur die Möglichkeit, den jeweils langen Weg zu wählen. Nach Ablauf dieser Zeit konnten die Ameisen zusätzlich den kurzen Pfad nutzen. Es stellt sich die Frage, ob die Ameisen in der Lage sind, noch immer den kürzeren Pfad zu finden. Das beobachtete Verhalten ist nun abhängig von der Spezies der Ameisen. Während die Argentinischen Ameisen (*Linepithema humile*) auf dem langen Pfad verharren, ist die Spezies *Lasius niger* in der Lage, mit hoher Wahrscheinlichkeit nach einiger Zeit den kurzen Pfad zu wählen. In (Beckers et al., 1992) wurde das unterschiedliche Verhalten der einzelnen Spezies hinsichtlich mehrerer Eigenschaften untersucht (beispielsweise Frequenz der Pheromonabsonderung, Größe der Kolonie und Fähigkeit der Ameisen, unterschiedlich starke Pheromonspuren vom Nest weg und zum Nest hin abzusondern). Die für das Doppelbrückenexperiment verwendeten Ameisen sind nahezu blind. Dies ist jedoch nicht notwendigerweise der Fall. Die Wüstenameise *Cataglyphis fortis* kann nach einer Futtersuche, die mehrere Hundert Meter durch hügeliges Dünengelände vom Nest weg führt, durch eine Verrechnung des Weges geradlinig zum Ausgangspunkt zurückkehren. Diese erstaunliche Fähigkeit basiert auf der visuellen Orientierung anhand von Polarisationsmustern des Himmels (Wohlgemuth et al., 2001).

Das prinzipielle Verhalten der Ameisen bei Untersuchungen, für die Nest und Futterquelle durch ein oder zwei Doppelbrücken verbunden sind, lässt sich durch folgende Eigenschaften der Pheromone und der Ameisen erklären. Erstens werden Pheromone entlang des Weges, auf dem die Ameisen laufen, abgesondert, zweitens verdunsten die Duftstoffe mit der Zeit, und drittens bevorzugen Ameisen die Wege, auf denen sie eine höhere Pheromonkonzentration mit ihren Sinnesorganen wahrnehmen können. Beim Doppelbrückenexperiment sind Ameisen, die jeweils den kürzeren Teilpfad bei der Futtersuche wählen, die ersten, die bei der Futterquelle ankommen. Laufen diese Ameisen wieder zurück zum Nest, dann werden sie an der ersten Gabelung beim kürzeren Teilpfad eine höhere Pheromonintensität erkennen, da diese Weggabelung von den Ameisen, welche die langen Teilpfade gewählt haben, noch nicht erreicht wurde. Auf dem Rückweg wird daher mit höherer Wahrscheinlichkeit der kürzere Teilpfad gewählt. Dieser sich selbst verstärkende oder autokatalytische Prozess führt dazu, dass nach einer gewissen Zeit die Pheromonintensität auf den langen Teilpfaden deutlich geringer ist, und nahezu alle Ameisen den kürzeren Teilpfad wählen.

Die Experimente inspirierten Dorigo in seiner Dissertation zur Verwen-

derung der Prinzipien der indirekten Kommunikation für ein Suchverfahren zur Lösung schwieriger kombinatorischer Optimierungsprobleme. Das Ziel der algorithmischen Umsetzung ist nicht die Modellierung eines realen Ameisenstaats oder die Simulation des Verhaltens der Ameisen — vielmehr soll für eine konkrete Problemstellung eine möglichst gute Lösung gefunden werden. In (Dorigo et al., 1991a,b) und Dorigo (1992) wurden erstmals verschiedene Varianten von Ameisenalgorithmen vorgeschlagen. Als übergeordneter Begriff für Ameisenalgorithmen wurde durch die Arbeiten (Dorigo und Di Caro, 1999; Dorigo et al., 1999) die Bezeichnung „Ant Colony Optimization (ACO) Metaheuristik“ geprägt. Eine Metaheuristik ist allgemein ein iteratives Verfahren, bei dem zugrunde liegende, meist problemspezifische Heuristiken zur Lösung eines Optimierungsproblems verwendet werden. Dorigo et al. verstehen unter einem ACO-Algorithmus eine „Instanziierung der ACO-Metaheuristik“, wobei die Definition der ACO-Metaheuristik sehr allgemein gehalten ist. Im Wesentlichen wird jedes auf einer Kolonie von künstlichen Ameisen basierende Verfahren, bei dem die Individuen konstruktiv eine Lösung erzeugen, die stochastischen Einzelentscheidungen der Lösungskonstruktion auf Pheromonwerten basieren, und durch Pheromonausschüttung und Pheromonverdunstung viel versprechende Entscheidungen verstärkt werden, als Instanziierung der ACO-Metaheuristik gesehen. Ein Ameisenalgorithmus folgt nicht notwendigerweise allen Aspekten der ACO-Metaheuristik. Daher ist jeder ACO-Algorithmus ein Ameisenalgorithmus, das Umgekehrte gilt jedoch nicht. Ein Beispiel für einen Ameisenalgorithmus, der nicht allen Aspekten der ACO-Metaheuristik folgt, ist in (Gambardella et al., 1999b) gegeben. Hier werden Pheromonwerte nicht zur Lösungskonstruktion, sondern vielmehr zur Lösungsmodifikation im Sinne einer lokalen Suche verwendet. Ameisenalgorithmen, die keine ACO-Algorithmen sind, stellen jedoch die Ausnahme dar. Daher verwenden wir in dieser Arbeit beide Begriffe, ohne Ameisenalgorithmen als Verallgemeinerung von ACO-Algorithmen zu sehen.

1.2 Ameisenalgorithmus

Ein Hauptaugenmerk dieser Arbeit wird in der Analyse wesentlicher Elemente von ACO-Algorithmen und der darauf aufbauenden Verbesserungen herkömmlicher Verfahren liegen. Ein zweiter Schwerpunkt liegt in der Modellierung von ACO-Algorithmen, durch die wir ein tieferes Verständnis des Optimierungsverfahrens erhalten. Wir führen einen sehr einfachen ACO-Algorithmus für Permutationsprobleme ein, der die wesentlichen Elemente der in der Literatur

üblichen ACO-Algorithmen enthält, und als Grundlage für die in dieser Arbeit vorgestellten Untersuchungen dient. Der Pseudocode ist in Algorithmus 1.1 dargestellt.

Algorithmus 1.1 einfacher ACO-Algorithmus für Permutationsprobleme

```

1: Initialisiere alle Pheromonwerte  $\tau_{ij}$ 
2: repeat
3:   for Ameise  $k \in \{1, \dots, m\}$  do
4:      $S := \{1, 2, \dots, n\}$            //Menge der verfügbaren Elemente
5:     while  $S \neq \emptyset$  do
6:       berechne Index  $i$  der nächsten Entscheidung
7:       wähle  $j \in S$  mit Wahrscheinlichkeit  $p_{ij}$  entsprechend
          Gleichung (1.1)
8:        $S := S - \{j\}$ 
9:        $\pi(i) := j$ 
10:    end while
11:  end for
12:  for all  $(i, j)$  do
13:     $\tau_{ij} := (1 - \rho) \cdot \tau_{ij}$            //Verdunstung
14:  end for
15:  for all  $(i, j) \in$  beste Lösung do
16:     $\tau_{ij} := \tau_{ij} + \Delta(\rho, \text{Kosten der besten Lösung})$  //Aktualisierung
17:  end for
18: until Abbruchkriterium erreicht

```

Sowohl bei den empirischen Untersuchungen als auch bei der Einführung eines deterministischen Modells für Ameisenalgorithmen, werden Permutationsprobleme als wesentliches Beispiel dienen, sodass sie hier kurz eingeführt werden sollen. Es sei \mathcal{P}_n die Menge aller möglichen Permutationen der Länge n . Weiterhin sei eine Kostenfunktion c gegeben, mit der wir einer Permutation Kosten zuordnen können. Gesucht ist eine Permutation π aus der Menge der Permutationen mit minimalen Kosten, d.h. $c(\pi) = \min\{c(\pi') \mid \pi' \in \mathcal{P}_n\}$. Innerhalb einer Generation (oder auch Iteration) erzeugt jede der m Ameisen eine Lösung. Dabei gehen die Ameisen konstruktiv vor: für jede der n Stellen einer Permutation wird festgelegt, welches Element an die entsprechende Stelle der Permutation platziert wird. Die zu wählende Reihenfolge der Entscheidungen kann dabei im voraus bekannt sein oder sich durch die jeweils zuletzt gefällte Entscheidung ergeben. Die Auswahlwahrscheinlichkeit eines

der noch verfügbaren Elemente ergibt sich wie folgt.

$$p_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{h \in \mathcal{S}} (\tau_{ih})^\alpha (\eta_{ih})^\beta} \quad (1.1)$$

Dabei ist \mathcal{S} die Menge aller noch verfügbaren Elemente. Für die Berechnung der Wahrscheinlichkeiten der möglichen Entscheidungen werden dabei Pheromonwerte (τ_{ij}) und Heuristikwerte (η_{ij}) verwendet. Ein hoher Pheromonwert bedeutet ähnlich wie im biologischen System, dass vorherige Ameisen gute Lösungen gefunden haben, wenn die entsprechende Entscheidung getroffen wurde (a posteriori Maß). Bei Permutationsproblemen lassen sich die Pheromonwerte in einer *Pheromonmatrix* speichern. Geeignete Heuristikwerte (η_{ij}) geben die Güte einer Entscheidung aufgrund im Voraus bekannter Informationen der Problem Instanz wieder (a priori Maß). Der relative Einfluss der Pheromonwerte und der heuristischen Informationen wird durch die Parameter α und β gesteuert. Nachdem alle m Ameisen eine Lösung erzeugt haben, werden alle Pheromonwerte um einen prozentual fixen Anteil verkleinert. Diese *Verdunstungsrate* wird üblicherweise mit ρ bezeichnet. Danach folgt die *Pheromonaktualisierung*. Wenn wir von der Aktualisierung der Pheromonmatrix sprechen, dann meinen wir, dass Ameisen die Pheromonwerte von Entscheidungen entsprechend der jeweiligen Lösungspermutation erhöhen. In Algorithmus 1.1 wird die Pheromonmatrix anhand der Lösung der besten Ameise einer Iteration aktualisiert. Bei mehreren Ameisen mit gleicher bester Lösungsqualität aktualisiert üblicherweise eine zufällig gewählte der besten Ameisen die Matrix. Für die Aktualisierung wird eine evtl. von der Verdunstungsrate und meist von der Lösungsqualität abhängige Menge an Pheromon auf die entsprechenden Stellen der Pheromonmatrix addiert.

Beispiel 1: Ein sehr intensiv untersuchtes Problem der kombinatorischen Optimierung ist das Problem des Handlungsreisenden (TSP, Lawler et al. 1985). Um Algorithmus 1.1 auf das asymmetrische TSP anzupassen, müssen wir die Matrixkodierung der Pheromonmatrix, die Vorgehensweise bezüglich der Entscheidungsreihenfolge und die Berechnung der Heuristikwerte festlegen. Die Matrixkodierung wird so gewählt, dass sowohl die Zeilen als auch die Spalten den Städten entsprechen. Der Wert τ_{ij} ist somit ein Maß dafür, wie die Lösungsgüte vorheriger Ameisen war, die ausgehend von Stadt i als nächstes die Stadt mit dem Index j gewählt haben. Die Anfangsstadt der Lösungspermutation wird zufällig gewählt (nicht in Algorithmus 1.1 dargestellt), die Entscheidungsreihenfolge ergibt sich durch die jeweils letzte Entscheidung.

Als Heuristikwert einer Entscheidung wird für das Problem des Handlungsreisenden üblicherweise $\eta_{ij} = \frac{1}{d_{ij}}$ gewählt, wobei d_{ij} der Abstand von Stadt i zu Stadt j ist.

Beispiel 2: Als zweites Beispiel betrachten wir ein einfaches Ein-Maschinen-Planungsproblem, bei dem wir eine Permutation von n Jobs suchen, sodass der dadurch beschriebene unterbrechungsfreie Belegungsplan minimale Kosten erzeugt. Jeder Job habe eine Bearbeitungsdauer und einen Fälligkeitstermin. Die Kosten eines einzelnen Jobs entsprechen der Verspätung bezüglich seines Fälligkeitstermins. Die Kosten eines Belegungsplans berechnen sich aus der Summe der Kosten aller Jobs. Die Kodierung der Pheromonmatrix wird so gewählt, dass die Zeilen den Plätzen der Jobs in der Lösungspermutation und die Spalten den einzelnen Jobs entsprechen. Die Entscheidungsreihenfolge wird entsprechend der „intuitiven“ Einplanungsreihenfolge für das Ein-Maschinen-Planungsproblem gewählt, d.h. der i -te Lösungskonstruktionsschritt entspricht der Entscheidung, welcher der noch nicht eingeplanten Jobs auf Platz i der Lösungspermutation gewählt wird. Der Wert τ_{ij} ist somit ein Maß dafür, wie die Lösungsgüte vorheriger Ameisen war, die Job j auf Platz i der Lösungspermutation gewählt haben. Die Heuristikwerte lassen sich beispielsweise wie folgt berechnen. Wenn eine Ameise eine partielle Lösung erzeugt hat, dann ergibt sich der Heuristikwert der i -ten Entscheidung aus den Kosten c_{ij} , die durch die Einplanung des Jobs j auf Platz i entstehen würden. Wir wählen z.B. $\eta_{ij} = \frac{1}{c_{ij}+1}$. Man beachte, dass im Gegensatz zum TSP, bei dem die Entfernungen zwischen den Städten und somit die in Beispiel 1 vorgeschlagenen Heuristikwerte $\eta_{ij} = \frac{1}{d_{ij}}$ statisch sind, die Kosten c_{ij} bei einem Ein-Maschinen-Planungsproblem evtl. abhängig von der schon erzeugten partiellen Lösung sein können.

1.3 Vergleich künstlicher und realer Ameisen

In diesem Abschnitt vergleichen wir das Verhalten realer Ameisen mit dem Verhalten künstlicher Ameisen in ACO-Algorithmen. Reale Ameisen führen natürlich keine Zufallsexperimente durch, ebenso wenig beruht das Verhalten realer Ameisen direkt darauf, dass beispielsweise das Ziel verfolgt wird, einen kürzesten Pfad zu finden. Vielmehr ergibt sich durch die in den vorherigen Abschnitten erwähnten Mechanismen ein Optimierungsverhalten. Ameisen werden in der Fachliteratur zur Biologie häufig durch Individuen modelliert, die zufallsbasiert Entscheidungen treffen.

Bezüglich der Vorgehensweise bei der Lösung eines Optimierungsproblems bestehen sowohl Ähnlichkeiten als auch Unterschiede zwischen realen und künstlichen Ameisen (Dorigo et al. 1999). Bei beiden Systemen agieren Ameisen als Teil einer Kolonie von Ameisen. Beide Systeme verändern ihre „Umgebung“ mit Hilfe von Pheromonen und verwenden das Prinzip der indirekten Kommunikation. In beiden Systemen wird, ausgehend von einem definierten Anfangszustand, eine partielle Lösung sukzessiv erweitert. Des Weiteren sind in beiden Systemen bei der sukzessiven Lösungskonstruktion die Einzelentscheidungen zufallsbasiert (bzw. können so beschrieben werden), und es werden, so weit bekannt, vornehmlich nur lokale Informationen des Systems verwendet. Im Gegensatz zu realen Ameisenstaaten agieren die künstlichen Ameisen eines Algorithmus in einem diskreten Suchraum, die Absonderung der Pheromone wird beim Algorithmus meist erst nach der Erzeugung einer Lösung durchgeführt und ist evtl. auch abhängig von der Lösungsgüte. Künstliche Ameisen haben ein Gedächtnis und können bei der konstruktiven Lösungserzeugung ihr Wissen über schon gefällte Entscheidungen einbringen. Des Weiteren können künstliche Ameisen im Gegensatz zu realen Ameisen mit zusätzlichen Fähigkeiten bei der Lösungserzeugung versehen werden (z.B. lokale Optimierung oder Vorausschau).

1.4 Erweiterungen

Im Folgenden wollen wir ausgehend von den erstmals publizierten Ameisenalgorithmen erfolgreiche Erweiterungen vorstellen, die das Optimierungsverhalten deutlich verbessern können. Bei den drei mit Ant System (AS) bezeichneten Varianten, welche in (Dorigo, 1992) auf das TSP angewendet wurden, wird noch, im Gegensatz zu dem Beispiel aus Abschnitt 1.2, eine Pheromonaktualisierung verwendet, bei der alle Ameisen einer Generation Pheromon auf alle zur Lösungspermutation gehörenden Stellen der Pheromonmatrix addieren. Die einzelnen Varianten unterscheiden sich wie folgt. Bei der mit Ant-density bezeichneten Variante wird auf den Pheromonwert τ_{ij} (dieser entspricht dem Pheromonwert auf der Kante von Stadt i zu Stadt j) eine konstante Pheromonmenge addiert. Bei der Variante Ant-quantity ist die Pheromonmenge umgekehrt proportional zur Entfernung der beiden Städte, die zur entsprechenden Entscheidung gehören. Bei der mit Ant-cylce bezeichneten Variante ist die addierte Pheromonmenge umgekehrt proportional zur Qualität der von einer Ameise gefundenen Lösung. Bei diesen ersten Untersuchungen war Ant-cylce die beste der drei Varianten. In (Dorigo, 1992) wurde bereits ei-

ne *Elitestrategie* eingeführt: zusätzlich zur eben erläuterten Vorgehensweise aktualisiert die Ameise mit der besten aller bisher gefundenen Lösungen in jeder Iteration die Pheromonmatrix. Diese Lösung wird als *global beste Lösung* bezeichnet. In (Gambardella und Dorigo, 1995; Dorigo und Gambardella, 1996) werden bei der mit Ant-Q bezeichneten Erweiterung von AS bei der Pheromonaktualisierung Ideen des Q-Learning (Watkins, 1989) verwendet, die zu einem verbessertem Optimierungsverhalten auf dem Problem des Handlungsreisenden führten. Durch eine Verkleinerung von Pheromonwerten während der Lösungskonstruktion können darauf folgende Ameisen der gleichen Iteration davon abgehalten werden, gleiche Entscheidungen immer wieder zu fällen (also den gleichen Weg zwischen zwei Städten immer wieder zu wählen). Durch diese Vorgehensweise wird die Diversität der Lösungen einer Iteration erhöht. In (Gambardella und Dorigo, 1996; Dorigo und Gambardella, 1997a,b) wurde, wiederum für das TSP, ausgehend von Ant-Q die mit Ant Colony System (ACS) bezeichnete Variante von Ameisenalgorithmen vorgeschlagen. Die Berechnungen der Pheromonwertveränderungen während der Lösungskonstruktion wurde unter Beibehaltung der erreichten Lösungsqualität deutlich vereinfacht. Üblicherweise wird diese einfachere Variante mit *lokale Pheromonaktualisierung* bezeichnet. Für die *globale Pheromonaktualisierung*, also die Aktualisierung der Pheromonwerte, nachdem alle Ameisen Lösungspermutationen erzeugt haben, wird sowohl bei Ant-Q als auch bei ACS eine Elitestrategie verwendet, d.h. es aktualisieren nicht mehr alle Ameisen einer Generation die Pheromonmatrix, sondern entweder nur die beste Ameise einer Iteration, die global beste oder beide Ameisen. Diese Methode hat sich auch für andere Optimierungsprobleme als sehr erfolgreich erwiesen (vgl. Stützle und Hoos 2000; Merkle et al. 2002). Eine rangbasierte Variante der (globalen) Pheromonaktualisierung wurde in (Bullnheimer et al., 1999) untersucht. Bei der rangbasierten Aktualisierung dürfen die k besten Ameisen einer Generation die Pheromonmatrix verändern, wobei Ameisen mit einem besseren Rang einen stärkeren Einfluss auf die Aktualisierung haben.

In (Gambardella und Dorigo, 1995; Dorigo und Gambardella, 1997b) wurde die so genannte Pseudo-random Proportional Rule vorgeschlagen, mit der die Suche in viel versprechenden Regionen des Suchraumes verstärkt wird. Dabei werden die Wahrscheinlichkeiten für Einzelentscheidungen nicht immer entsprechend Gleichung (1.1) berechnet. Mit einer gewissen Wahrscheinlichkeit, die üblicherweise mit q_0 bezeichnet wird, wählt eine Ameise das Element mit dem maximalen Wert des Produktes aus Pheromon- und Heuristikwert. Hierfür wird für jede Entscheidung zunächst eine Zufallszahl $q \in [0, 1]$ er-

zeugt. Falls $q > q_0$ gilt, dann wird das Element j entsprechend Gleichung (1.1) gewählt, ansonsten wählen wir das Element j mit

$$j = \operatorname{argmax}_{h \in S} (\tau_{ih})^\alpha (\eta_{ih})^\beta \quad (1.2)$$

In (Stützle, 1999; Stützle und Hoos, 1997, 2000) wurde ein auf mehreren Optimierungsproblemen sehr erfolgreich getesteter ACO-Algorithmus mit der Bezeichnung MAX-MIN Ant System vorgestellt. Bei dem von Stützle und Hoos vorgeschlagenen Algorithmus dürfen die Pheromonwerte τ nicht unter einen minimalen Pheromonwert τ_{\min} fallen und nicht über einen maximalen Pheromonwert τ_{\max} steigen. Dadurch wird eine vorzeitige Konvergenz in einem lokalen Optimum vermieden, da die Wahrscheinlichkeit, eine beliebige Lösung zu finden, nicht gegen 0 strebt. In Kombination mit einer der Lösungskonstruktion nachgeschalteten lokalen Suche und einer Elitestrategie konnten für das TSP und insbesondere für das quadratische Zuordnungsproblem (QAP, Koopmans und Beckmann 1957) hervorragende Ergebnisse erzielt werden.

Obwohl die Parallelisierung von Ameisenalgorithmen nahe liegt, wurden bisher verhältnismäßig wenige Methoden hierfür vorgeschlagen. Mit der ersten in (Bolondi und Bondanza, 1993) vorgestellten sehr feinkörnig parallelen Variante, bei der jeder Ameise ein eigenes Prozessor-Element eines hochparallelen Rechners zugeordnet wird, führt ein ungünstiges Skalierungsverhalten zu schlechten Ergebnissen. Bessere Resultate wurden in (Stützle, 1995) erreicht. Stützle vergleicht die Lösungsqualität vieler kurzer unabhängiger und damit einfach parallel ausführbarer Läufe mit der Lösungsqualität eines langen Laufs. Gute Ergebnisse für das QAP wurden in (Talbi et al., 1999) mit einem auf dem Master-Worker-Prinzip basierenden Ameisenalgorithmus erzielt. In (Middendorf et al., 2002) wurden für einen Ansatz mit mehreren Kolonien unterschiedliche Methoden des Informationsaustauschs zwischen den Kolonien untersucht. In (Merkle und Middendorf, 2001b), (Merkle und Middendorf, 2002c) und (Janson et al., 2002) wurden Methoden zur schnellen Implementierung von Ameisenalgorithmen auf rekonfigurierbaren Architekturen vorgeschlagen. Die Verfahrensänderungen führten dabei unter Beibehaltung der Lösungsqualität zu einer in der Anzahl der Ameisen und der Problemgröße quasi-linearen Laufzeit.

1.5 Anwendungen

Inzwischen werden Ameisenalgorithmen auf eine Vielzahl unterschiedlicher Optimierungsprobleme erfolgreich angewendet. Eine umfangreiche Übersicht

ist z.B. in (Stützle und Dorigo, 2002a) gegeben. An dieser Stelle werden nur einige der publizierten Arbeiten erwähnt. Wie wir auch schon im vorherigen Abschnitt gesehen haben, wurden viele Erweiterungen auf dem Problem des Handlungsreisenden (Dorigo, 1992; Stützle und Hoos, 2000; Cordón et al., 2000; Gambardella und Dorigo, 2000) oder verwandten Problemen (Bauer et al., 1999; Gambardella et al., 1999a) und auf dem quadratischen Zuordnungsproblem (Maniezzo, 1999; Stützle und Hoos, 2000) untersucht. Sehr gute Ergebnisse konnten bei Planungsproblemen erzielt werden. Darunter fallen z.B. Jobshop-Planungsprobleme (Colorni et al., 1994), Flowshop-Planungsprobleme (Stützle, 1998) und Planungsprobleme mit Fälligkeitsterminen (Merkle und Middendorf, 2002a; den Besten et al., 2000). In (Michels und Middendorf, 1999) wurden Ameisenalgorithmen für das „Shortest Common Supersequence“ Problem untersucht, bei dem für eine endliche Menge an Zeichenketten ein möglichst kurzes Wort w gefunden werden soll, sodass für jede Zeichenkette gilt, dass w Supersequenz der Zeichenkette ist (d.h. durch das Löschen einzelner Buchstaben von w kann die Zeichenkette erzeugt werden). Dabei erzielte der Ameisenalgorithmus im Vergleich mit anderen Verfahren sehr viel versprechende Resultate.

Ameisenalgorithmen werden auch für dynamische Problemstellungen erfolgreich eingesetzt. Unter anderem in (Schoonderwoerd et al., 1996) und (Di Caro und Dorigo, 1998) werden für die Berechnung von Routingtabellen für den Transport von Datenpaketen in Netzwerken Ameisenalgorithmen verwendet. Im Vergleich mit klassischen Verfahren konnten deutlich bessere Ergebnisse erzielt werden. In (Guntsch et al., 2001) wurde für das dynamische Problem des Handlungsreisenden mit wegfallenden und hinzukommenden Städten untersucht, in welchen Fällen sich eine explizite Veränderung der Pheromoninformation auf das Optimierungsverhalten positiv auswirkt.

1.6 Theorie

Bisher gibt es nur relativ wenige theoretische Untersuchungen zu Ameisenalgorithmen. Ziehen wir den Vergleich zum verwandten und länger existierenden Forschungsbereich der evolutionären Algorithmen (siehe z.B. Michalewicz 1996), dann erkennen wir, dass dort die theoretischen Untersuchungen schon sehr viel ausgereifter sind. Eine grobe Klassifizierung der Arbeiten findet sich in (Beyer et al., 2002). In die erste Klasse fallen Arbeiten, in denen exakte Aussagen bezüglich des Verhaltens von evolutionären Algorithmen über nur eine Generation hinweg gemacht werden. Hierzu gehören beispielsweise alle Arbei-

ten über das in diesem Forschungsbereich sehr bekannte und viel diskutierte Schema-Theorem (Holland, 1975). Die zweite Klasse enthält Arbeiten, die Aussagen bezüglich des Konvergenzverhaltens von evolutionären Algorithmen machen. In der dritten Klasse von Arbeiten werden für die Untersuchung der Algorithmen Modelle verwendet, die eine exakte Analyse vereinfachen (z.B. unendlich große Populationen, vgl. Kapitel 7). In die vierte Klasse fallen die Arbeiten, die das Optimierungsverhalten anhand der ebenfalls bekannten und in ihrer Bedeutung umstrittenen Building-Block-Hypothese (Goldberg, 1989) analysieren. Erst in den letzten Jahren wurden genetische Algorithmen mit klassischen Methoden der Algorithmen- und Komplexitätstheorie untersucht (siehe z.B. Jansen und Wegener 2002). Das immer stärker werdende Interesse aus dem Bereich der Algorithmentheorie liegt nicht zuletzt an den hervorragenden Ergebnissen, die empirisch mit evolutionären Verfahren erzielt werden konnten.

Ameisenalgorithmen sind eine vergleichsweise junge Metaheuristik, die inzwischen bei zahlreichen Applikationen auch beste Resultate liefert. Wie schon erwähnt, existieren theoretische Resultate jedoch kaum. Dies liegt unter anderem an den Abhängigkeiten der Einzelentscheidungen während der Lösungskonstruktion, die ACO-Algorithmen für eine theoretische Analyse schwer zugänglich machen. In (Gutjahr, 2000) wurde ein Konvergenzbeweis für GBAS (Graph-based Ant System) vorgestellt. GBAS ist ein Ameisenalgorithmus, der als wesentlichen Unterschied zu AS (siehe Abschnitt 1.4) eine veränderte Pheromonaktualisierung verwendet. Pheromonwerte werden nur von Ameisen erhöht, deren Lösungsqualität mindestens genau so gut wie die bisher global beste Lösung ist. Für Probleme mit genau einer optimalen Lösung wird gezeigt, dass für ein beliebig kleines ϵ die optimale Lösung mit Wahrscheinlichkeit $1 - \epsilon$ gefunden wird. Das Konvergenzresultat wird in (Gutjahr, 2002) verschärft. In (Stützle und Dorigo, 2002b) wird für eine Variante des MAX-MIN Ant Systems gezeigt, dass die optimale Lösung mit Wahrscheinlichkeit 1 gefunden wird. Auch dieser Beweis beruht darauf, dass immer nur eine Ameise die Pheromonmatrix aktualisieren darf, welche die global beste Lösung verbessert.

Eine andere Richtung theoretische Resultate für das Optimierungsverhalten von Ameisenalgorithmen zu erhalten, wird in den Arbeiten (Blum et al., 2001), (Rubinstein, 1999) und (M. Zlochin und Dorigo, 2001) verfolgt. In allen drei Arbeiten werden Verfahren untersucht, die herkömmlichen Ameisenalgorithmen sehr ähnlich sind, aber teilweise mathematisch fundiertere Aussagen bezüglich der Entwurfsentscheidungen der Verfahren zulassen. Blum

et al. schlagen vor, die Pheromonaktualisierung als Bewegung des Vektors aller Pheromonwerte in einem Hyperwürfel zu sehen, bei dem die Eckpunkte Lösungen des Optimierungsproblems entsprechen. Dadurch soll der theoretische Vergleich von Ameisenalgorithmen mit Methoden evolutionärer Algorithmen einfacher werden. Rubinstein hat in seiner Arbeit die so genannte Cross-Entropy-Methode zur Optimierung schwieriger Probleme vorgeschlagen, die starke Ähnlichkeiten zu AS aufweist und beispielsweise auch Aussagen über das Konvergenzverhalten des Verfahrens zulässt. Während ACO-Algorithmen bei zahlreichen Optimierungsproblemen zu den besten Algorithmen überhaupt zählen, fehlt dieser empirische Nachweis für die Cross-Entropy-Methode bisher (vgl. Stützle und Dorigo 2002a). In (M. Zlochin und Dorigo, 2001) sind Ameisenalgorithmen, die Cross-Entropy-Methode und andere verwandte Verfahren Instanzen eines sehr allgemein formulierten Optimierungsverfahrens. In darauf aufbauenden Arbeiten sollen übergreifende mathematisch fundierte Aussagen getroffen werden, mit denen bei den verwendeten Mechanismen zwischen verfahrenstechnischer Notwendigkeit und unnötigem Überbleibsel unterschieden werden kann.

Die erste Formulierung eines deterministischen Modells zur Analyse der Dynamik von ACO-Algorithmen wurde in (Merkle und Middendorf, 2001e, 2002d,e) und (Merkle und Middendorf, 2002g) vorgeschlagen. Diese Methoden werden ausführlich in Kapitel 7 vorgestellt.

1.7 Gliederung der Arbeit

Diese Arbeit verfolgt das Ziel, einen signifikanten Beitrag hinsichtlich der Optimierung *von*, der Optimierung *mit* und der modellbasierten exakten Analyse von Ameisenalgorithmen zu leisten. Diese Dreiteilung spiegelt sich auch in der Gliederung der Arbeit wider. Im ersten Teil werden Methoden zur Verbesserung des Entscheidungsprozesses während der Lösungskonstruktion der Ameisen angegeben und empirisch untersucht. Ausgehend davon werden im zweiten Teil Verfahren für Probleme vorgestellt, mit denen sich viele in der Praxis vorkommende Problemstellungen modellieren lassen. Anhand umfangreicher Tests wird der Erfolg der Methoden gezeigt. Im dritten Teil der Arbeit werden mit Hilfe eines deterministischen Modells numerisch exakte Vorhersagen bezüglich des Optimierungsverhaltens von ACO-Algorithmen gemacht.

Die Kapitelgliederung ergibt sich wie folgt. Die Verbesserung durch Verfahrensmodifikationen und Erweiterungen von herkömmlichen Ameisenalgorithmen wird auf unterschiedlich schwierigen Problemen belegt. In Kapitel 2 defi-

nieren wir einfache Permutationsprobleme, die charakteristische Eigenschaften schwieriger Probleme widerspiegeln und identifizieren einen unerwünschten Effekt (die Verzerrung von Auswahlwahrscheinlichkeiten), der entscheidenden Einfluss auf das Optimierungsverhalten von ACO-Algorithmen hat und daher sowohl bei den Erweiterungen als auch bei der theoretischen Analyse mit Hilfe eines deterministischen Modells von zentraler Bedeutung ist. In Kapitel 3 stellen wir Methoden vor, bei denen durch eine globale Auswertung der in einer Pheromonmatrix gespeicherten Information ein deutlich besseres Optimierungsverhalten erzielt werden kann. Dass eine gezielte Anpassung der Entscheidungsreihenfolge, die eine künstliche Ameise beim Aufbau einer Lösung durchläuft, zu deutlich besseren Resultaten führen kann, wird in Kapitel 4 empirisch nachgewiesen. Ebenso wie in Kapitel 3 werden wir Resultate der neuen Verfahren sowohl für einfache Probleme im Sinne von Kapitel 2 als auch für NP-vollständige Probleme präsentieren. In Kapitel 5 stellen wir Methoden vor, bei denen der gesamte Suchprozess in Phasen eingeteilt wird, und phasenabhängige Anpassungen durchgeführt werden. Der resultierende Ameisenalgorithmus erzielt für die größten und am häufigsten verwendeten Testprobleme für das praxisnahe Problem der Projektplanung unter beschränkten Ressourcen im Vergleich zu den über einem Dutzend weiteren in der Literatur vorgeschlagenen Heuristiken das beste Ergebnis und konnte für viele Instanzen neue beste obere Schranken finden. Für multikriterielle Optimierungsprobleme, die bezüglich mehrerer konkurrierender Kriterien optimiert werden müssen, werden in Kapitel 6 Kooperationsmechanismen zwischen mehreren Kolonien von künstlichen Ameisen entwickelt, die in Kombination mit einer globalen Pheromonauswertung für ein Ein-Maschinen-Planungsproblem bei Beachtung von zwei Optimierungskriterien (Umrüstkosten und Verspätungen) erfolgreich zur Lösung eingesetzt werden.

In Kapitel 7 stellen wir ein deterministisches Modell für ACO-Algorithmen vor, welches auf den erwarteten Aktualisierungswerten der Pheromonwerte basiert. Anhand einer Fixpunktanalyse, die auf der Duftstoffinformation der Pheromonmatrix basiert, werden Aussagen über die Dynamik von Ameisenalgorithmen für Permutationsprobleme getroffen. Das Modellverhalten entspricht in vieler Hinsicht sehr gut dem zufallsgesteuerten Ameisenalgorithmus und ermöglicht auch für komplexe Optimierungsprozesse analytische Vorhersagen.

Im letzten Kapitel werden die Ergebnisse der Arbeit zusammengefasst und ein Ausblick auf weitere Untersuchungen gegeben.

Kapitel 2

Ameisenalgorithmen und einfache Permutationsprobleme

2.1 Motivation

Ameisenalgorithmen wurden inzwischen erfolgreich auf zahlreiche Optimierungsprobleme angewendet (vgl. Abschnitt 1.5). Die in der Literatur übliche Vorgehensweise, um die Algorithmen zu untersuchen, beschränkt sich meist auf die Anwendung von Standardverfahren oder Verfahrenserweiterungen auf Benchmark-Probleme oder Probleme der realen Welt („Real-world Problems“). Im Gegensatz dazu werden beispielsweise in den Arbeiten (Stützle und Dorigo, 2001) und (Merkle und Middendorf, 2001d) Erweiterungen und Verfahrensmodifikationen anhand sehr „einfacher“ Probleme untersucht, um die erfolgreiche Anwendbarkeit von ACO-Algorithmen auf „komplexeren“ Problemen zu erklären. Die Verbesserungen durch die in dieser Arbeit vorgestellten Verfahren werden jeweils, soweit möglich, auf unterschiedlich schweren Problemen belegt. In Kapitel 3 und 4 werden wir Methoden zur besseren Nutzung der in der Pheromonmatrix gespeicherten Information vorstellen. Neben empirischen Studien auf NP-vollständigen Problemen wird dabei die jeweilige Methode auch immer auf einfache Permutationsprobleme, die in diesem Kapitel definiert werden, angewendet. Ein Vorteil der Untersuchungen anhand einfacher Probleme ist, dass Effekte, die auch bei komplexeren Optimierungsproblemen auftreten, leichter zu analysieren und besser zu verstehen sind. Während wir die Verfahren in Kapitel 5 und 6 auf praxisnahe Probleme anwenden (Planungsprobleme mit Nebenbedingungen und multikriterielle Opti-

mierungsprobleme), wird das zu untersuchende Optimierungsproblem in Kapitel 7 so weit vereinfacht, bis wir analytische Aussagen über die Dynamik der ACO-Algorithmen mit Hilfe eines numerisch exakten Modells machen können. Einfache Probleme, die in diesem Kapitel eingeführt werden, dienen hierzu als Ausgangspunkt für Anwendung und Analyse.

In Abschnitt 2.2 wird zunächst definiert, was wir im Rahmen dieser Arbeit unter einfachen Permutationsproblemen verstehen. In Abschnitt 2.3 wird eine erste Instanz eines einfachen Problems vorgestellt. Der in Abschnitt 1.1 eingeführte ACO-Algorithmus wird auf diesem Problem untersucht. Eine wichtige Beobachtung wird die Verzerrung der Auswahlwahrscheinlichkeiten sein. Wir werden im Abschnitt 2.4 zeigen, dass dieser Effekt zu einem unerwünschten Optimierungsverhalten führen kann. In Abschnitt 2.5 wird eine Zusammenfassung dieses Kapitels gegeben.

2.2 Definition einfacher Probleme

Wenn wir in dieser Arbeit das Optimierungsverhalten von Ameisenalgorithmen auf einfachen Problemen untersuchen, werden wir immer Instanzen von Permutationsproblemen analysieren, welche wie folgt definiert werden.

Gegeben seien n Elemente $1, 2, \dots, n$ und eine Kostenmatrix $C = [c(i, j)]$ der Größe $n \times n$ mit ganzzahligen Kosten $c(i, j) \geq 0$. Es sei \mathcal{P}_n die Menge aller Permutationen von $(1, 2, \dots, n)$. Für eine Lösungspermutation π berechnen wir die Kosten $c(\pi)$ wie folgt.

$$c(\pi) = \sum_{i=1}^n c(i, \pi(i)) \quad (2.1)$$

Gesucht ist eine Permutation $\pi \in \mathcal{P}_n$ mit minimalen Kosten, d.h. $c(\pi) = \min\{c(\pi') \mid \pi' \in \mathcal{P}_n\}$.

Dieses Problem ist in der Literatur als lineares Zuordnungsproblem oder auch als gewichtetes perfektes Zuordnungsproblem in bipartiten Graphen bekannt und kann mit der in (Kuhn, 1955) vorgeschlagenen „Ungarischen Methode“ mit Laufzeit $O(n^3)$ optimal gelöst werden.

2.3 Die Probleminstanz P_A

Für die erste von uns untersuchte einfache Probleminstanz soll gelten, dass für jede der Entscheidungen eines ACO-Algorithmus immer höhere Kosten entstehen, je mehr die Platzierung des gewählten Elements von der für dieses

Element einzigen optimalen Platzierung abweicht. Die Kostenmatrix wird wie folgt definiert. Für jedes Element (i, j) der Kostenmatrix $c(i, j)$ steigen die Kosten mit dem Abstand zur Hauptdiagonalen. Wir wählen $c(i, j) = |i - j| + 1$. Die einzige optimale Lösung π_{opt} für diese Problem Instanz erhalten wir, wenn alle Elemente auf der Hauptdiagonalen platziert werden. Hierbei entstehen die Kosten $c(\pi_{\text{opt}}) = n$. Die Kostenmatrix der Größe $n = 50$ ist in Abbildung 2.1 dargestellt. Wir bezeichnen die Instanz mit P_A (Abweichung).

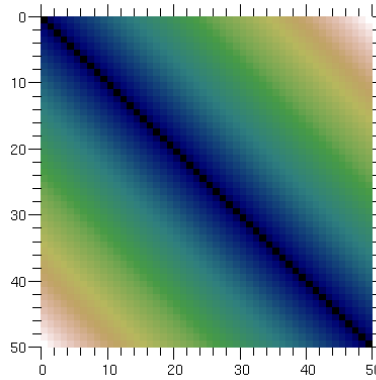


Abbildung 2.1: Kostenmatrix der Problem Instanz P_A (Abweichung); hellere Farben bedeuten höhere Kosten

Als Instanz eines sehr einfachen Planungsproblems interpretiert, entspricht P_A einem Ein-Maschinen-Planungsproblem mit n Jobs, deren Laufzeit jeweils $p_j = 1$ beträgt. Jeder Job j hat genau den Fälligkeitstermin j . Die Kosten, die pro Job entstehen (Lagerungs- oder Verspätungskosten), werden durch die absolute Abweichung des Fertigstellungszeitpunktes zum Fälligkeitstermin berechnet.

Im Folgenden werden wir den in Kapitel 1 auf Seite 5 vorgestellten ACO-Algorithmus auf das Problem P_A anwenden. Die methodischen Erweiterungen für ACO-Algorithmen werden in dieser Arbeit häufig anhand NP-vollständiger Planungsprobleme untersucht. Daher wählen wir als Entscheidungsreihenfolge bei der Lösungskonstruktion einer Ameise jene, die sich intuitiv aus der Einplanungsreihenfolge bei der Konstruktion eines Belegungsplans für ein Ein-Maschinen-Planungsproblem ergibt. Die i -te Entscheidung entspricht somit der Wahl eines Elements an der i -ten Stelle der Lösungspermutation. Falls nicht explizit anders spezifiziert, sind alle Ergebnisse, die in dieser Arbeit für einfache Problem Instanzen vorgestellt werden, Mittelwerte aus 25 Läufen. Jeder Lauf wird beendet, falls über mindestens 500 Generationen hinweg

keine neue beste Lösung gefunden wird, oder falls der Algorithmus 10000 Generationen gelaufen ist. Für alle Probleminstanzen einfacher Probleme wird die Größe $n = 50$ gewählt. Die Verdunstungsrate der ACO-Algorithmen ist jeweils $\rho = 0.01$. Die Ameise einer Iteration mit der besten Lösungsqualität führt die Pheromonaktualisierung immer entsprechend der Formel $\tau_{ij} = \tau_{ij} + 1/(\text{Kosten der besten Lösung})$ durch. Die Pheromonmatrix wird immer mit $\tau_{ij} = 1/(\rho \cdot c(\pi_{\text{opt}}))$ initialisiert, wobei $c(\pi_{\text{opt}})$ die Kosten einer optimalen Permutation sind. Es sei π die beste Lösungspermutation der ersten Iteration. Man beachte, dass der Initialisierungswert der Pheromonmatrix dem maximal erreichbaren Pheromonwert entspricht, da

$$\underbrace{\frac{1}{\rho \cdot c(\pi_{\text{opt}})}}_{\text{initialer Pheromonwert}} \cdot \underbrace{(1 - \rho)}_{\text{Verdunstung}} + \underbrace{\frac{1}{c(\pi)}}_{\text{Aktualisierung}} \leq \frac{1}{\rho \cdot c(\pi_{\text{opt}})}$$

gilt (die Gleichheit gilt für $\pi = \pi_{\text{opt}}$), und die Pheromonwerte somit nie über den Initialisierungswert steigen können. Für die Untersuchungen auf einfachen Probleminstanzen wird keine Heuristik verwendet, d.h. es gilt $\beta = 0$.

Abbildung 2.2 zeigt den Entwicklungsstand der durchschnittlichen Pheromonmatrix zu unterschiedlichen Zeitpunkten. Jedes Element einer durchschnittlichen Pheromonmatrix zu einem gewissen Zeitpunkt ist der Durchschnittswert der Pheromonwerte der einzelnen Läufe zu diesem Zeitpunkt. Die durchschnittliche Lösungsqualität der besten Lösung nach 25 Läufen beträgt 287.4. Im Mittel wurde nach 3456 Generationen die beste Lösung gefunden. Wir halten folgende wichtige Beobachtungen fest.

1. Der verwendete ACO-Algorithmus platziert Elemente tendenziell zu früh. In den oberen 2/3 Zeilen der Pheromonmatrix sind die Pheromonwerte oberhalb der Diagonalen höher als unterhalb der Diagonalen.
2. Das Verfahren konvergiert in den ersten Zeilen schneller als in den letzten Zeilen.
3. Die Pheromonwerte sind während des gesamten Laufs, insbesondere in den letzten Zeilen, an Stellen erhöht, die relativ hohen Kosten entsprechen.

Zusammenfassend beobachten wir auf der vorgestellten einfachen Probleminstanz eine deutlich asymmetrische Erhöhung der Pheromonwerte mit dem Standard-Ameisenalgorithmus bei der Auswahl der Elemente. Dieses unerwünschte Verhalten steht in klarem Zusammenhang mit der „Verzerrung der

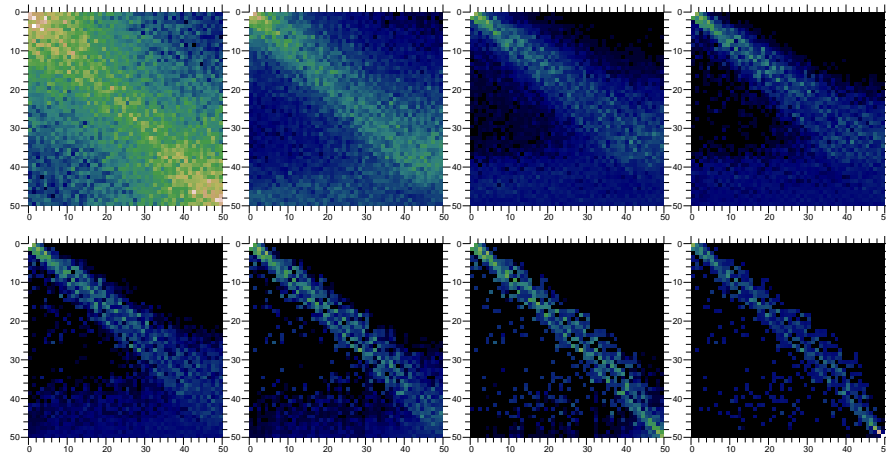


Abbildung 2.2: Durchschnittliche Pheromonmatrizen der Messläufe auf der Instanz P_A ; dargestellt sind die Pheromonmatrizen in den Generationen 500, 1000, 1500, ..., 4000; hellere Farben bedeuten höhere Pheromonwerte

Auswahlwahrscheinlichkeiten“, welche wir im nächsten Abschnitt genauer untersuchen werden.

2.4 Verzerrung der Auswahlwahrscheinlichkeiten

Wie schon zuvor erläutert, beruht die Verwendung von Pheromonen bei ACO-Algorithmen auf folgendem Prinzip. Ein Pheromonwert spiegelt das Ergebnis der Entscheidungen von guten Ameisen früherer Generationen wider. Im Falle der hier betrachteten Permutationsprobleme entspricht dies den Entscheidungen, welches Element auf welchen Platz gewählt wurde. Durch die Verdunstung verlieren die Entscheidungen früherer Generationen mehr und mehr an Bedeutung. Die Ameisen sollten die Entscheidungen entsprechend der Wahrscheinlichkeitsverteilung treffen, die sich durch die relative Menge der Pheromonwerte ergibt. Man beachte, dass diese Wahrscheinlichkeitsverteilung noch durch Heuristikwerte modifiziert werden kann.

Das folgende Beispiel zeigt ein generelles Problem, welches bei dieser Vorgehensweise auftritt. Für dieses Beispiel verwenden wir, ebenso wie bei den Untersuchungen an größeren einfachen Probleminstanzen, keine Heuristikinformation. Gegeben sei ein Problem mit $n = 3$ Elementen und Pheromonwerten entsprechend folgender Pheromonmatrix.

$$\begin{pmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \end{pmatrix} = \begin{pmatrix} \frac{1}{10} & \frac{3}{10} & \frac{3}{5} \\ \frac{3}{5} & \frac{1}{10} & \frac{3}{10} \\ \frac{3}{10} & \frac{3}{5} & \frac{1}{10} \end{pmatrix} \quad (2.2)$$

Die Zeilen- und Spaltensummen der Pheromonwerte in diesem Beispiel sind jeweils genau 1. Man beachte hierbei, dass die Multiplikation der Pheromonmatrix mit einem positiven konstanten Wert die entstehenden Wahrscheinlichkeitsverteilungen nicht verändert, da nur die Pheromonwerte relativ zur Gesamtsumme der Pheromonwerte aller wählbaren Elemente relevant sind. In der ersten Zeile wählt die Ameise die einzelnen Elemente mit den Wahrscheinlichkeiten $p_{1j} = \tau_{1j}$. In der ersten Zeile entspricht daher die Wahrscheinlichkeit, ein Element j zu wählen, genau dem relativen Anteil des Pheromonwerts τ_{1j} an der Gesamtsumme der Pheromonwerte der ersten Zeile. In der zweiten Zeile unterscheidet sich der relative Anteil der Pheromonwerte von den Wahrscheinlichkeiten, mit denen ein Element gewählt wird, da das Ergebnis der Entscheidung der ersten Zeile berücksichtigt werden muss. Die Wahrscheinlichkeit, dass beispielsweise Element 1 auf Platz 2 gewählt wird, falls Element 2 auf Platz 1 gewählt wurde, beträgt $(\tau_{21}/(\tau_{21} + \tau_{23}))$. Die Wahrscheinlichkeit Element 1 auf Platz 2 zu wählen ist somit $p_{21} = p_{12}(\tau_{21}/(\tau_{21} + \tau_{23})) + p_{13}(\tau_{21}/(\tau_{21} + \tau_{22})) = 5/7 > 3/5$. Analog rechnen wir $p_{22} = 1/9 > 1/10$ und $p_{23} = 7/40 < 3/10$. Für dieses kleine Beispiel können wir diese Wahrscheinlichkeiten noch einfach explizit ausrechnen, bei größeren Beispielen ist das nur mit sehr viel Rechenaufwand möglich. Wir werden daher bei den Untersuchungen die Wahrscheinlichkeitsverteilung durch eine Häufigkeitsverteilung annähern. Die Nichtentsprechung der sich aus den relativen Anteilen der Pheromonwerte ergebenden Wahrscheinlichkeitsverteilungen und der tatsächlich beobachteten Verteilungen bezeichnen wir als *Verzerrung der Auswahlwahrscheinlichkeiten*.

Im Folgenden zeigen wir, dass die Verzerrung der Auswahlwahrscheinlichkeiten bei größeren Pheromonmatrizen sehr stark werden kann. Bei Pheromonmatrizen, die während eines Optimierungslaufs eines ACO-Algorithmus auftreten, wird diese systematische Verzerrung zu beobachten sein. Exemplarisch werden wir eine typische Pheromonmatrix untersuchen, um den erwähnten Effekt zu zeigen. Wir werden die Verfahrenserweiterungen der folgenden Kapitel auch jeweils mit der gleichen Untersuchungsmethode basierend auf der gleichen typischen Pheromonmatrix untersuchen. Unsere Pheromonmatrix sei eine solche, die erhöhte Pheromonwerte auf Elementen der Hauptdiagonalen

und auf Elementen nahe der Hauptdiagonalen hat. Die Matrix habe die Größe $n = 30$, und die Pheromonwerte seien wie folgt definiert. Für $i \in [3, 28]$ sei $\tau_{ii} = 2/5$, $\tau_{i,i-1} = \tau_{i,i+1} = 1/5$, $\tau_{i,i-2} = \tau_{i,i+2} = 1/10$ und $\tau_{ij} = 1/20$ falls $j \notin \{i-2, i-1, i, i+1, i+2\}$. In den Zeilen $i \in \{1, 2, 29, 30\}$ passen wir die Pheromonwerte an den Rändern so an, dass die Zeilen- und Spaltensummen der Pheromonmatrix identisch bleiben. Wir definieren hierfür $\tau_{11} = 11/20$, $\tau_{22} = 2/5$ und $\tau_{21} = \tau_{12} = 1/4$ und passen die entsprechenden symmetrischen Stellen für $i \in \{29, 30\}$ auch an. Die Zeilen- und Spaltensummen sind somit jeweils $9/4$.

Um eine Häufigkeitsverteilung zu berechnen, lassen wir eine Generation von $m = 100000$ Ameisen Lösungen mit Hilfe dieser konstruierten Pheromonmatrix erzeugen. Es sei m_{ij} die Anzahl der Ameisen, die in dieser Generation Element j auf Platz i wählen. Da in jeder Zeile die Summe der Pheromonwerte $9/4$ beträgt, müssten, bei Beibehaltung der Wahrscheinlichkeiten, die sich durch die relativen Verhältnisse der Pheromonwerte ergeben, $m_{ij}^* := \tau_{ij} \cdot (4/9) \cdot 100000$ Ameisen Element j auf Platz i wählen. Es sei $d_{ij} := m_{ij} - m_{ij}^*$. Die *Matrix der positiven Abweichungen* $D^+ := [\max\{0, d_{ij}\}]$ und die *Matrix der negativen Abweichungen* $D^- := [\min\{0, d_{ij}\}]$ sind in Abbildung 2.3 dargestellt. Man beachte, dass die den Werten zugeordneten Farben für jede Matrix zwischen dem minimalen und maximalen Matrixwert skaliert werden und daher ein direkter Vergleich der Farben zwischen den zwei Matrizen nicht sinnvoll ist.

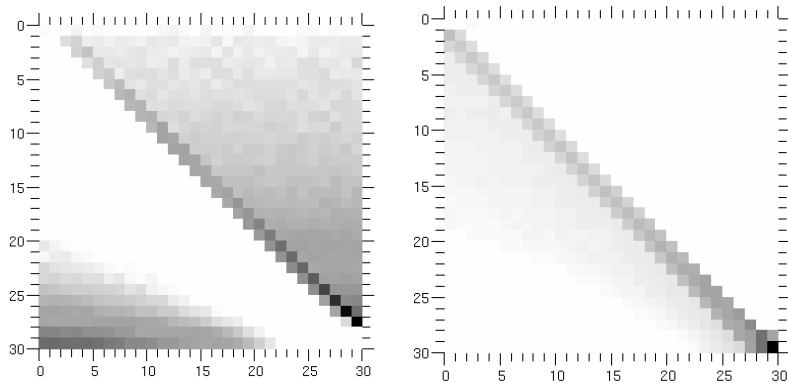


Abbildung 2.3: links: Matrix der positiven Abweichungen D^+ : weiß=0, schwarz=4246; rechts: Matrix der negativen Abweichungen D^- : weiß=0, schwarz=-22450

In Abbildung 2.3 ist zu erkennen, dass eine systematische Verzerrung der

Auswahlwahrscheinlichkeiten bei den Entscheidungen der Ameisen existiert. Die negativen Werte d_{ij} kommen im Wesentlichen entlang der Diagonalen vor. Das bedeutet, dass die Ameisen die Entscheidungen entlang der Diagonalen zu selten treffen. Die positiven d_{ij} Werte kommen oberhalb der Diagonalen und auch in der linken unteren Ecke vor. Der Grund für die erhöhten Werte oberhalb der Diagonalen sind die erhöhten Pheromonwerte entlang der Diagonalen der Pheromonmatrix. Diese verringern die Wahrscheinlichkeiten dafür, dass ein Element unterhalb der Diagonalen gewählt wird (und erhöhen daher die Pheromonwerte rechts neben der Diagonalen).

Die erhöhten Werte in der linken unteren Ecke können folgendermaßen erklärt werden. Angenommen eine Ameise wählt ein Element mit kleinem Index zufallsbedingt nicht auf einem der Plätze, auf denen sich die dem Element zugehörigen hohen Pheromonwerte befinden. Dann wird dieses Element oft erst am Ende der Lösungskonstruktion gewählt, da bei den vorherigen Entscheidungen noch Elemente mit höheren Pheromonwerten vorhanden sind.

Man beachte, dass die beobachtete Verzerrung problemunabhängig und inhärent verfahrensbedingt ist. Die Pheromonmatrix, die wir zur Veranschaulichung der Verzerrung gewählt haben, tritt genau so während eines Optimierungslaufs wahrscheinlich nie auf. Dennoch kann man erwarten, dass sich beispielsweise bei der Optimierung der in Abschnitt 2.3 eingeführten Probleminstanz P_A ähnliche Pheromonmatrizen während der Optimierung ergeben. Für die Probleminstanz P_A verschlechtert die Verzerrung der Auswahlwahrscheinlichkeiten das Optimierungsverhalten deutlich, da viele Ameisen in einer Generation Entscheidungen mit hohen Kosten fällen (Elemente mit kleinem Index werden bei der Lösungskonstruktion erst spät gewählt). Bei gegebener Probleminstanz lassen sich auch *gewichtete positive* und *gewichtete negative Abweichungen* berechnen, indem man die positiven und negativen Abweichungen jeweils mit den entsprechenden Kosten multipliziert. Wirkt sich also beispielsweise die Verzerrung so aus, dass eine kostengünstige Entscheidung zu oft getroffen wird, dann führt das zu einer hohen gewichteten positiven Abweichung. Beim Vergleich unterschiedlicher Verfahren auf einfachen Probleminstanzen werden wir in den nächsten Kapiteln auch jeweils die gewichteten Abweichungen untersuchen.

2.5 Zusammenfassung

In diesem Kapitel haben wir eine Untersuchungsmethode eingeführt, um Effekte von Verfahrensänderungen und -erweiterungen zu visualisieren und zu

analysieren. Die Methode beruht darauf, Probleme so zu definieren, dass sie eine einfache Struktur besitzen, aber charakteristische Eigenschaften komplexerer Optimierungsprobleme widerspiegeln und somit eine vereinfachte und aussagekräftige Analyse von Ameisenalgorithmen ermöglichen. Wir werden bei der Vorstellung neuer Methoden in den nächsten Kapiteln immer wieder die entsprechenden Verfahrensänderungen auch auf geeignet gewählten einfachen Probleminstanzen untersuchen.

Ein weiteres wichtiges Ergebnis dieses Kapitels ist die Beobachtung einer systematischen Verzerrung der Auswahlwahrscheinlichkeiten, die mittels einfacher Probleme besonders deutlich zu erkennen ist. Bei der Elementauswahl des Standard-Ameisenalgorithmus spiegeln die Pheromonwerte nicht die tatsächlich beobachteten Auswahlwahrscheinlichkeiten wider. Das Ausmaß der Verzerrung wird bei der Vorstellung neuer Methoden in den folgenden Kapiteln immer wieder verwendet werden, um qualitative Aussagen zu einer Methode zu machen. Im letzten Teil dieser Arbeit werden wir anhand eines deterministischen Modells analytisch nachweisen, dass die Verzerrung der Auswahlwahrscheinlichkeiten entscheidenden Einfluss auf die Güte und die Dynamik von Ameisenalgorithmen hat.

Kapitel 3

Globale Pheromonauswertung

Beim Standard-Ameisenalgorithmus, wie er in Kapitel 1 vorgestellt wurde, wird jeweils genau eine Zeile der Pheromonmatrix verwendet, um die Wahrscheinlichkeiten zu berechnen, dass einzelne Elemente gewählt werden. Diese Pheromonauswertungsmethode bezeichnen wir als *lokale Pheromonauswertung*. In diesem Kapitel werden wir zwei Verfahren einführen, bei denen zur Berechnung der Wahrscheinlichkeitsverteilungen die in der Matrix gespeicherte Information besser genutzt wird. Die erste dieser *globalen Pheromonauswertungsmethoden*, die wir mit *Summen-Pheromonauswertung* bezeichnen, wird in Abschnitt 3.1 eingeführt. Diese Methode wurde erstmals in (Merkle und Middendorf, 2000) vorgestellt. Inzwischen wurde sie beispielsweise in (Teich et al., 2001) für ein Jobshop Planungsproblem und in (Rajendran und Ziegler, 2001) für ein Flowshop Planungsproblem verwendet. Für eine allgemeine Klasse von Planungsproblemen wurde die Methode in (Blum und Sampels, 2002) untersucht und hat gute Ergebnisse erzielt. In (Bautista und Pereira, 2002) erzielte die Summen-Pheromonauswertung für das Assembly Line Balancing Problem in einem Vergleich von zwölf ACO-Algorithmus Varianten die besten Ergebnisse. Die erfolgreiche Anwendung auf das Problem der Projektplanung unter beschränkten Ressourcen wurde in (Merkle et al., 2000a) vorgestellt und wird ausführlich in Kapitel 5 erörtert. Die zweite globale Pheromonauswertungsmethode wird in Abschnitt 3.2 vorgestellt und mit *relative Pheromonauswertung* bezeichnet. Teile der vorgestellten Resultate für die relative Pheromonauswertung wurden in (Merkle und Middendorf, 2002b) veröffentlicht.

3.1 Summen-Pheromonauswertung

Die bisher in der Literatur üblichen ACO-Algorithmen berechnen die Wahrscheinlichkeiten von Einzelentscheidungen in einem lokalen Sinne. Ein einzelner Pheromonwert spiegelt aufgrund vorheriger Lösungskonstruktionen wider, wie viel versprechend eine Entscheidung ist. In diesem Abschnitt werden wir eine Methode vorstellen, bei denen Pheromonwerte „früherer“ Entscheidungen der Lösungskonstruktion Einfluss auf die Berechnung der Wahrscheinlichkeiten von Einzelentscheidungen haben. Des Weiteren werden wir eine Kombination der lokalen und der Summen-Pheromonauswertung vorstellen. Nach der Beschreibung der Methode in Abschnitt 3.1.1 werden wir die Summen-Pheromonauswertung in Abschnitt 3.1.2 auf einfache Probleme, und in Abschnitt 3.1.3 auf ein NP-vollständiges Planungsproblem anwenden.

3.1.1 Motivation und Vorgehensweise

Für Probleme, bei denen die Zeilen der Pheromonmatrix den Plätzen in der Lösungspermutation entsprechen, bedeutet ein hoher Pheromonwert τ_{ij} , dass es vorteilhaft ist, Element j auf Platz i zu wählen. Nehmen wir an, wir haben ein Problem mit der folgenden Eigenschaft. Wenn ein Element nicht auf Platz i , sondern in der Nähe von Platz i gewählt wird, dann erhöhen sich die Kosten der Lösung im Allgemeinen mit zunehmender Abweichung vom Platz i . Des Weiteren nehmen wir an, eine Ameise wählt zufallsbedingt bei der Auswahl des Elements für Platz i das Element mit dem Index h (statt des Elements mit dem Index j , welches einen höheren Pheromonwert habe). Um nun immer noch eine Lösungspermutation mit niedrigen Kosten zu erhalten, ist es sicherlich von Vorteil, das Element j möglichst bald (d.h. auf einem der nächsten Plätze) zu wählen, da durch die Änderung um wenige Plätze in der Lösungspermutation auch nur mit einer geringen Veränderung der resultierenden Lösungsqualität zu rechnen ist. Oft ist es so, dass die in ACO-Algorithmen verwendete Heuristikinformation η_{lj} für $l > i$ in diesem Falle unterstützend wirkt. Wenn jedoch die Pheromonwerte τ_{lj} klein sind, weil bisher keine guten Lösungen gefunden wurden, die Element j auf einem Platz $l > i$ gewählt haben, dann bleibt das Produkt der Pheromoninformation und der Heuristikinformation $(\eta_{lj})^\alpha \cdot (\tau_{lj})^\beta$, und somit die entsprechende Wahrscheinlichkeit das Element zu wählen, klein. Es werden daher eher andere Elemente gewählt, bis j gegen Ende der Lösungskonstruktion mit evtl. sehr hohen Kosten gewählt werden muss. In einem solchen Falle steigen die Kosten

der Lösungspermutation stark, und die gefundene Lösung ist schlecht.

Um dieses Problem zu beheben, stellen wir eine Methode vor, bei der die Pheromonwerte τ_{ij} auch einen Einfluss auf Entscheidungen der Plätze $l > i$ haben. Anstatt nur die Pheromonwerte τ_{ij} bei der Berechnung der Wahrscheinlichkeitsverteilung einer Entscheidung zu verwenden, berechnen wir die Summe aller Pheromonwerte von der ersten Zeile bis zur Zeile der aktuellen Entscheidung. Falls wir die in Abschnitt 1.4 vorgestellte Pseudorandom Proportional Rule verwenden, dann wählt eine Ameise das nächste Element j aus der Menge \mathcal{S} aller noch verfügbaren Elemente auf Platz i mit Wahrscheinlichkeit q_0 so, dass folgender Ausdruck maximal ist.

$$\left(\sum_{k=1}^i (\tau_{kj}) \right)^\alpha \cdot (\eta_{kj})^\beta \quad (3.1)$$

Mit Wahrscheinlichkeit $1 - q_0$ wählt die Ameise das Element entsprechend der Wahrscheinlichkeiten, die wie folgt berechnet werden.

$$p_{ij} = \frac{\left(\sum_{k=1}^i (\tau_{kj}) \right)^\alpha \cdot (\eta_{ij})^\beta}{\sum_{h \in \mathcal{S}} \left(\sum_{k=1}^i (\tau_{kh}) \right)^\alpha \cdot (\eta_{ih})^\beta} \quad (3.2)$$

Die Vorgehensweise bei der Berechnung der Wahrscheinlichkeiten für die Summen-Pheromonauswertung ist in Abbildung 3.1 dargestellt.

Um früheren Entscheidungen unterschiedlich starke Gewichte bei der Berechnung der Auswahlwahrscheinlichkeiten zu geben, führen wir die *gewichtete Summen-Pheromonauswertung* ein. Mit einem zusätzlichen Parameter $\gamma > 0$ steuern wir den Einfluss. Die Auswahlwahrscheinlichkeiten werden durch

$$p_{ij} = \frac{\left(\sum_{k=1}^i (\gamma^{i-k} \cdot \tau_{kj}) \right)^\alpha \cdot (\eta_{ij})^\beta}{\sum_{h \in \mathcal{S}} \left(\sum_{k=1}^i (\gamma^{i-k} \cdot \tau_{kh}) \right)^\alpha \cdot (\eta_{ih})^\beta} \quad (3.3)$$

berechnet. Falls $\gamma = 1$ ist, erhalten wir die reine Summen-Pheromonauswertung, d.h. jeder der Pheromonwerte erhält den gleichen Einfluss. Falls $\gamma < 1$ ($\gamma > 1$) ist, dann erhalten die Pheromonwerte früherer Entscheidungen weniger (mehr) Einfluss. Für $\gamma = 0$ erhalten wir die lokale Pheromonauswertung.

Für die Kombination der lokalen und der Summen-Pheromonauswertung berechnen wir die Auswahlwahrscheinlichkeiten wie in Formel (1.1) auf Seite 6, jedoch werden zuvor die τ_{ij} durch die Werte τ_{ij}^* ersetzt, die wie folgt berechnet

$$[\tau_{ij}] = \begin{pmatrix} \begin{array}{c|c|c|c} \tau_{11} & & & \tau_{1n} \\ \tau_{21} & & & \tau_{2n} \\ \vdots & & & \vdots \\ \tau_{k1} & \tau_{k2} & \dots & \tau_{kn} \\ \tau_{n1} & & & \tau_{nn} \end{array} \end{pmatrix}$$

Abbildung 3.1: Darstellung der Summen-Pheromonauswertung : Die Wahrscheinlichkeit Element j auf Platz k zu wählen, ergibt sich durch die Summe der dunkler unterlegten Werte dividiert durch die Summe aller farblich unterlegten Pheromonwerte. Die farblich nicht hervorgehobenen Spalten sind solche, für die bei einem Platz kleiner als k ein Element gewählt wurde.

werden.

$$\tau_{ij}^* := c \cdot x_i \cdot \tau_{ij} + (1 - c) \cdot y_i \cdot \sum_{k=1}^i \gamma^{i-k} \tau_{kj} \quad (3.4)$$

Hierbei sind $x_i := \sum_{h \in \mathcal{S}} \sum_{k=1}^i \gamma^{i-k} \tau_{kh}$ und $y_i := \sum_{h \in \mathcal{S}} \tau_{ih}$ Faktoren, die den relativen Einfluss der lokalen Pheromonauswertung und der Summen-Pheromonauswertung korrigieren. Man beachte, dass ohne diese Korrektur der Einfluss der Summen-Pheromonauswertung im Allgemeinen viel stärker wäre. Für $c = 1$ erhalten wir die reine lokale Pheromonauswertung und für $c = 0$ die reine gewichtete Summen-Pheromonauswertung. Für $c = 0.5$ haben beide Auswertungsmethoden den gleichen Einfluss.

Zunächst untersuchen wir die Verzerrung der Auswahlwahrscheinlichkeiten für die neue Pheromonauswertungsmethode. Hierfür wird das gleiche Experiment wie in Abschnitt 2.4 durchgeführt. Die gewählte Pheromonmatrix und die Definition von m_{ij}^* sind identisch zum Experiment aus Abschnitt 2.4. Wir wählen auch wieder $m = 100000$ Ameisen zur Berechnung der Abweichungen. Es sei m_{ij}^{Sum} die Anzahl der Ameisen, die bei Verwendung der Summen-Pheromonauswertung Element j auf Platz i gewählt haben, und es sei $d_{ij}^{\text{Sum}} := m_{ij}^{\text{Sum}} - m_{ij}^*$. Die Matrix der positiven Abweichungen $D^{\text{Sum}+} := [\max\{0, d_{ij}^{\text{Sum}}\}]$

und die Matrix der negativen Abweichungen $D^{\text{Sum}^-} := [\min\{0, d_{ij}^{\text{Sum}}\}]$ sind in Abbildung 3.2 dargestellt. Wir können wieder eine Verzerrung der Auswahlwahrscheinlichkeiten beobachten. Es ist jedoch deutlich zu erkennen, dass die Abweichung eher in der Nähe der Stellen auftritt, an denen die Pheromonwerte der exemplarisch gewählten Pheromonmatrix erhöht wurden. Insbesondere bei einer problemabhängigen Betrachtung mittels der maximalen gewichteten Abweichung wird der Unterschied deutlich. Zur Gewichtung der Abweichungen verwenden wir die durch das Problem P_A definierte Kostenmatrix $[c(i, j)]$ mit der Kostenfunktion $c(i, j) = |i - j| + 1$. Die maximale gewichtete positive Abweichung der Summen-Pheromonauswertung sei das Maximum aller Werte $c(i, j) \cdot \max\{0, d_{ij}^{\text{Sum}}\}$ mit $i, j \in [1, 30]$. Während bei der lokalen Pheromonauswertung die maximale gewichtete positive Abweichung 88823 ist, sinkt der Wert durch die Summen-Pheromonauswertung auf 16784. Dies kann zumindest als Indiz dafür gewertet werden, dass mit der neuen Pheromonauswertungsmethode auf einer bestimmten Klasse von Problemen bessere Ergebnisse erzielt werden.

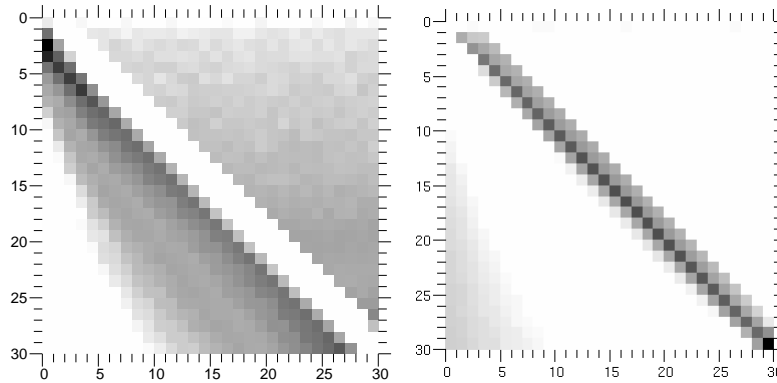


Abbildung 3.2: links: Matrix D^{Sum^+} : weiß=0, schwarz= 4690; rechts: Matrix D^{Sum^-} : weiß=0, schwarz=-16422

In den nächsten Abschnitten werden wir zunächst die reine Summen-Pheromonauswertung auf einem einfachen Permutationsproblem untersuchen. Danach werden wir die gewichtete Variante und die Kombination mit der lokalen Pheromonauswertung auf einem NP-vollständigen Problem betrachten.

3.1.2 Einfache Probleminstanzen

Summen-Pheromonauswertung für P_A

Die in Abschnitt 2.3 vorgestellte Probleminstanz P_A hat die Eigenschaft, dass eine geringe Veränderung bezüglich der Platzierung eines Elements in der Lösungspermutation nur eine geringe Veränderung der resultierenden Kosten zur Folge hat. Nach der Untersuchung der Verzerrung der Auswahlwahrscheinlichkeiten versprechen wir uns auf einem Problem mit einer solchen Struktur ein verbessertes Optimierungsverhalten. Die Entwicklung der durchschnittlichen Pheromonmatrizen (Mittelwert aus 25 Messläufen) bei Verwendung der Summen-Pheromonauswertung ist in Abbildung 3.3 dargestellt (für die lokale Pheromonauswertung vgl. Abbildung 2.2 auf Seite 19). Die Parametereinstellungen waren die gleichen wie bei Verwendung der lokalen Pheromonauswertung (vgl. Abschnitt 2.3). Die erreichte durchschnittliche Lösungsqualität bei Verwendung der Summen-Pheromonauswertung beträgt 51.9 (mit der lokalen Pheromonauswertung 287.4, der beste zu erreichende Wert ist 50). Die beste Lösung wurde durchschnittlich in Generation 2932 gefunden (mit lokaler Pheromonauswertung nach 3456 Generationen).

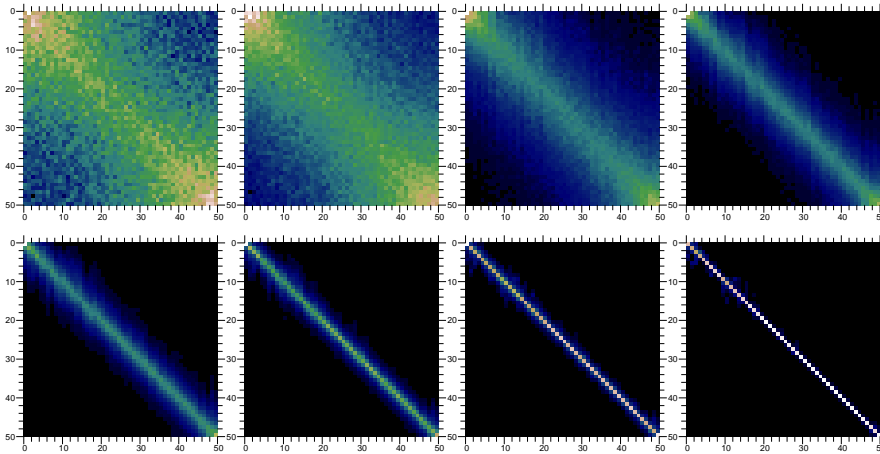


Abbildung 3.3: Durchschnittliche Pheromonmatrizen der Messläufe auf der Probleminstanz P_A ; dargestellt sind die Pheromonmatrizen in den Generationen 400, 800, 1200, ..., 3200 bei Verwendung der Summen-Pheromonauswertung; hellere Farben bedeuten höhere Pheromonwerte

Auf dem oben erläuterten Problem bestätigt sich somit die Vermutung, dass die Summen-Pheromonauswertung deutlich bessere Ergebnisse liefert als

die lokale Pheromonauswertung. In Abbildung 3.3 ist zu beobachten, dass die erhöhten Pheromonwerte bei Anwendung der neuen Auswertungsmethode sehr symmetrisch zur Hauptdiagonalen liegen und umso stärker erhöht sind, je näher der Platz der aktuellen Entscheidung an dem optimalen Platz des aktuell betrachteten Elements liegt. Während des Laufs fokussiert das Verfahren mehr und mehr auf die optimalen Entscheidungen.

Im Gegensatz zur lokalen Pheromonauswertung stellen wir Folgendes fest. Zum einen sind keine asymmetrisch erhöhten Pheromonwerte oberhalb der Diagonalen und keine hohen Pheromonwerte an Stellen mit hohen Kosten zu beobachten, zum anderen konvergiert das Verfahren im Gegensatz zur lokalen Pheromonauswertung gleichmäßig schnell in allen Zeilen.

Die deutlich verbesserte Lösungsqualität, die schnellere und vor allem gleichmäßigere Konvergenz sowie die Beseitigung unerwünschter Effekte bei Verwendung der lokalen Pheromonauswertung (hohe Pheromonwerte für Entscheidungen mit hohen Kosten) deuten darauf hin, dass die Summen-Pheromonauswertung bei geeigneter Problemstruktur auch auf schwierigeren Problemen erfolgreich angewendet werden kann.

Bevor wir die Summen-Pheromonauswertung auf einem NP-vollständigen Problem untersuchen, werden wir eine weitere einfache Probleminstanz vorstellen. Wir werden das Problem so konstruieren, dass mit der lokalen Pheromonauswertung im Vergleich zur Summen-Pheromonauswertung ein besseres Optimierungsverhalten zu erwarten ist.

Eine einfache Probleminstanz — P_D

Wir wählen die Kostenmatrix des Problems P_D so, dass es für jedes Element genau eine Platzierung gibt, die niedrige Kosten erzeugt. Alle anderen Plätze erzeugen gleich hohe Kosten, die größer sind als die Kosten der optimalen Platzierung. Wir setzen $c(i, j) = 1$ falls $i = j$ ist und $c(i, j) = 2$ sonst. Die Instanz erhält die Bezeichnung P_D (Diagonale). Die sehr einfache Kostenmatrix ist für $n = 30$ in Abbildung 3.4 dargestellt.

Die Entwicklung der Pheromonmatrizen bei Verwendung der lokalen und der Summen-Pheromonauswertung ist in den Abbildungen 3.5 und 3.6 dargestellt. Die durchschnittlich erreichte beste Lösungsqualität beträgt 50.28 bei der lokalen Pheromonauswertung. Die beste Lösung wird durchschnittlich nach 1377 Generationen gefunden. Wenn die Summen-Pheromonauswertung zur Berechnung der Auswahlwahrscheinlichkeiten verwendet wird, dann liegt die durchschnittlich beste Lösungsqualität bei 76.12. Die beste Lösung wird

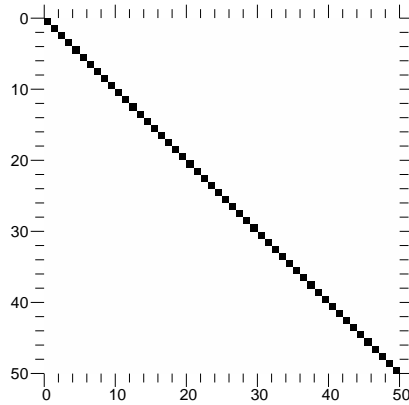


Abbildung 3.4: Kostenmatrix der Probleminstanz P_D : die Kosten auf der Diagonalen sind $c(i, i) = 1$, ansonsten ($i \neq j$) gilt $c(i, j) = 2$

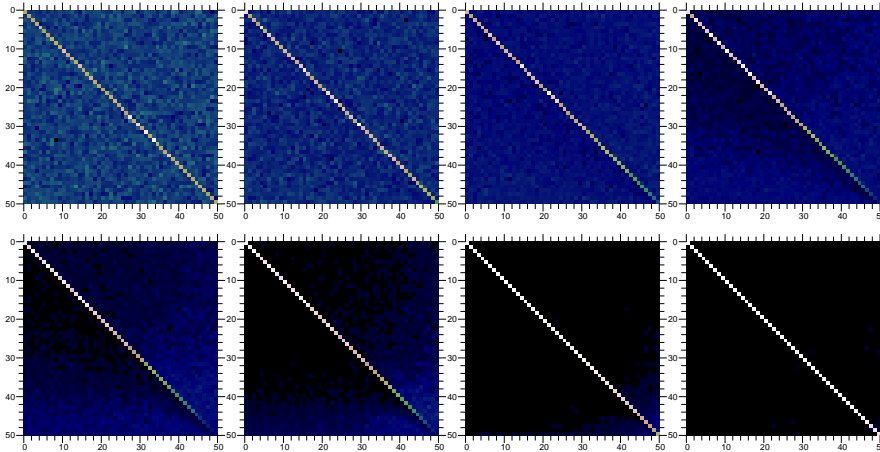


Abbildung 3.5: Durchschnittliche Pheromonmatrizen der Messläufe auf der Probleminstanz P_D ; dargestellt sind die Pheromonmatrizen in den Generationen 200, 400, 600, ..., 1600 bei Verwendung der lokalen Pheromonauswertung; hellere Farben bedeuten höhere Pheromonwerte

bei der Summen-Pheromonauswertung durchschnittlich nach 6248 Generationen gefunden.

Wenn ein Element i bei der Probleminstanz P_D nicht auf den optimalen Platz i gewählt wurde, dann ist es aus Sicht der Lösungsqualität unwichtig, auf welchem Platz ungleich i in der Lösungspermutation das Element stattdessen platziert wurde, da alle nicht-optimalen Entscheidungen Kosten 2 verursachen. Treten während eines Messlaufs hohe Pheromonwerte in der Nähe der

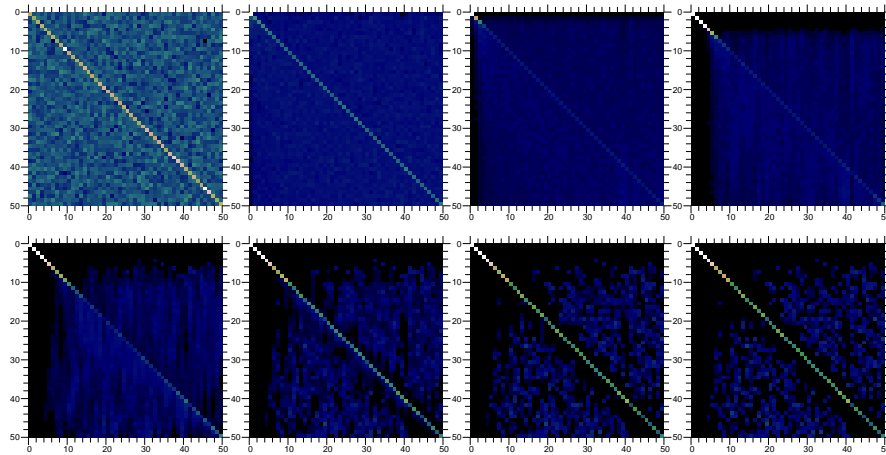


Abbildung 3.6: Durchschnittliche Pheromonmatrizen der Messläufe auf der Probleminstanz P_D ; dargestellt sind die Pheromonmatrizen in den Generationen 100, 500, 1000, 2000, 4000, 6000, 8000 und 10000 bei Verwendung der Summen-Pheromonauswertung; hellere Farben bedeuten höhere Pheromonwerte

Diagonalen auf (also nur wenig oberhalb oder wenig unterhalb), dann hat die Summen-Pheromonauswertung die Eigenschaft, dass mit hoher Wahrscheinlichkeit die optimale Entscheidung gefunden wird. Daher beobachten wir in den durchschnittlichen Pheromonmatrizen in Abbildung 3.6 am Ende der Optimierung, dass keine erhöhten Pheromonwerte in der Nähe der Diagonalen auftauchen. Allerdings ist das Optimierungsverhalten auf dem gewählten Problem im Vergleich zur lokalen Pheromonauswertung erwartungsgemäß schlecht (vgl. Abschnitt 3.1.1).

3.1.3 Ein NP-vollständiges Problem

In diesem Abschnitt werden wir die Summen-Pheromonauswertung, die gewichtete Summen-Pheromonauswertung und die Kombination mit der lokalen Pheromonauswertung auf ein NP-vollständiges Problem anwenden. Das Problem ist in der Literatur unter dem Namen „Single Machine Total Weighted Tardiness Problem“ (SMTWTP) bekannt. Es handelt sich dabei um ein Ein-Maschinen-Planungsproblem, bei dem für n gegebene Jobs ein Belegungsplan einer Maschine mit minimalen Kosten gefunden werden soll. Jeder Job j , $1 \leq j \leq n$ hat eine Laufzeit p_j , einen Fälligkeitstermin d_j und ein Gewicht für die Überschreitung des Fälligkeitstermins w_j . Gesucht ist ein unterbrechungs-

freier Ein-Maschinen-Belegungsplan (also eine Permutation der n Jobs), welcher die Zielfunktion $T = \sum_{j=1}^n w_j \cdot \max\{0, C_j - d_j\}$ minimiert. Dabei sei C_j der Fertigstellungszeitpunkt des Jobs j . T heißt die gewichtete Summe der Verspätungen des Plans. Der ungewichtete Fall des Problems ($w_j = 1$ für alle $j \in \{1, \dots, n\}$) ist in der Literatur unter dem Namen Single Machine Total Tardiness Problem (SMTTP) bekannt.

SMTTP ist schwach NP-vollständig (Du und Leung, 1990), und SMTWTP ist NP-vollständig (Lawler, 1977) im strengen Sinne. Ein pseudopolynomieller Algorithmus für den Spezialfall von SMTWTP, bei dem beim Vergleich zweier beliebiger Jobs aus einer kürzeren Laufzeit eines Jobs ($p_j < p_h$) ein mindestens genauso großes Gewicht bei der Überschreitung des Fälligkeitstermins folgt (d.h. $w_j \geq w_h$), wurde in (Lawler, 1977) vorgestellt. Man beachte, dass aus diesem Ergebnis auch direkt folgt, dass SMTTP pseudopolynomiell lösbar ist. In (Crauwels et al., 1998) ist ein Überblick über unterschiedliche Heuristiken für SMTWTP gegeben. Ein Ameisenalgorithmus für SMTWTP wurde auch in (den Besten et al., 2000) vorgestellt. Einer der wichtigsten Aspekte der Arbeit ist die Kombination einer lokalen Suche mit einem Ameisenalgorithmus. Die hervorragenden Ergebnisse der Arbeit beruhen genau auf der Kombination dieser Verfahren. Die hier vorgestellten Ergebnisse sollen die, durch die Tests auf den einfachen Probleminstanzen zu erwartende, erfolgreiche Anwendbarkeit der Summen-Pheromonauswertung auf einem schwierigen Problem belegen. Daher werden wir den ACO-Algorithmus zunächst auch nicht mit einer lokalen Suche oder sonstigen Erweiterungen versehen und nur die bisher vorgestellten Methoden verwenden. Dass die Summen-Pheromonauswertung sehr gute Ergebnisse im Vergleich mit zahlreichen anderen Verfahren für ein viel untersuchtes NP-vollständiges Problem liefert, wird in Kapitel 5 gezeigt. Die erfolgreiche Kombination der Summen-Pheromonauswertung mit einer lokalen Suche wurde in (Merkle et al., 2000c) auch für SMTWTP gezeigt. Wie in der Arbeit (den Besten et al., 2000) konnte für alle der untersuchten Probleminstanzen einer bekannten Benchmark-Bibliothek die bisher beste bekannte Lösung gefunden werden.

Heuristik für SMTWTP

Für die Tests auf einfachen Permutationsproblemen haben wir keine Heuristik eingesetzt, da diese die Interpretation der Ergebnisse bezüglich Konvergenzverhalten und Lösungsqualität nur erschwert und vom eigentlichen Fokus der Untersuchungen abgelenkt hätte. Metaheuristiken erweisen sich aber oftmals

als sehr gute Optimierungsverfahren, weil die Verwendung von problemspezifischen Heuristiken meist sehr einfach möglich ist. In (Bauer et al., 1999) wurde ein Ameisenalgorithmus für SMTTP vorgestellt. Die Heuristikwerte η_{ij} des Algorithmus wurden mit $\eta_{ij} = \frac{1}{\max\{\mathcal{T}+p_j, d_j\}}$ berechnet, wobei \mathcal{T} die Gesamtlaufzeit aller bis zum Platz i schon geplanten Jobs ist. Obwohl diese Heuristik als einfache Prioritätsregel durchaus eine sinnvolle Lösungspermutation erzeugt, ist die direkte Anwendung für ACO-Algorithmen ohne weiteres nicht sinnvoll. Bei Verwendung dieser Heuristik werden die Werte $\max\{\mathcal{T}+p_j, d_j\}$ am Ende des Plans immer größer. Daher sind die Unterschiede der Heuristikwerte auf den hohen Plätzen der Lösungspermutation im Allgemeinen sehr gering, und die Heuristik hat nur einen geringen Einfluss auf die Berechnung der Auswahlwahrscheinlichkeiten. Um diesen Effekt zu verhindern, modifizieren wir die Heuristikwerte wie folgt.

$$\eta_{ij} = \frac{1}{\max\{\mathcal{T} + p_j, d_j\} - \mathcal{T}} \quad (3.5)$$

Da wir die gewichtete Variante der Problemstellung SMTTP betrachten, multiplizieren wir die rechte Seite der Gleichung (3.5) mit dem Wert w_j . Man beachte, dass Jobs mit kleinen gewichteten Laufzeiten p_j/w_j einen hohen Heuristikwert haben, wenn $\mathcal{T} + p_j \geq d_j$ gilt. Aus einer Menge der noch wählbaren Jobs, die alle jeweils ihren Fälligkeitstermin überschritten haben (oder durch Einplanung überschritten werden), werden somit Jobs mit kürzeren Laufzeiten und höheren Gewichten bevorzugt. Dies ist auch erwünscht, denn 1.) durch eine Einplanung eines Jobs mit hoher Laufzeit würden alle sowieso schon verspäteten Jobs noch mehr verspätet, und 2.) von Jobs, die schon verspätet sind, bevorzugt man jene, deren Kosten mit sich erhöhender Gesamtlaufzeit stark wachsen. Aus einer Menge von Jobs, die alle noch nicht verspätet sind, werden die Jobs bevorzugt, denen ein kleiner Wert $\frac{w_j}{d_j - \mathcal{T}}$ zugeordnet ist. Auch dies ist erwünscht, denn 1.) werden so Jobs bevorzugt, deren Fälligkeitstermin sehr nahe ist (und die somit bald Kosten verursachen könnten), und 2.) werden Jobs bevorzugt, die ein hohes Gewicht haben (und somit später durch ihre hohen Gewichte auch keine hohe Kosten mehr verursachen können, wenn sie jetzt schon eingeplant werden).

Probleminstanzen und Parameter

Zur Untersuchung verwenden wir die 125 Probleminstanzen für SMTWTP, die in einer Standard-Bibliothek für Optimierungsprobleme (ORLIB, vgl. Beasley 1990) enthalten sind. Die Instanzen wurden jeweils wie folgt erzeugt. Jeder der

Jobs $j \in [1, 100]$ hat eine ganzzahlige Laufzeit p_j , die gleichverteilt aus dem Intervall $[1, 100]$ gewählt wurde. Die Gewichte w_j wurden gleichverteilt aus dem Intervall $[1, 10]$ gewählt. Die Fälligkeitstermine d_j wurden gleichverteilt aus dem Intervall

$$\left[\sum_{j=1}^{100} p_j \cdot \left(1 - \text{TF} - \frac{\text{RDD}}{2}\right), \sum_{j=1}^{100} p_j \cdot \left(1 - \text{TF} + \frac{\text{RDD}}{2}\right) \right] \quad (3.6)$$

gewählt. Der Wert RDD (relative range of due dates) bestimmt die Länge des Intervalls, aus denen die Fälligkeitstermine gewählt werden. Der Wert TF (tardiness factor) gibt die relative Position des Zentrums des Intervalls der Fälligkeitstermine im Intervall $[0, \sum_{j=1}^{100} p_j]$ an. Für jede der Kombinationen (TF, RDD) mit $\text{TF} \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ und $\text{RDD} \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ wurden 5 Probleminstanzen erzeugt.

Pheromon- und Heuristikwerte werden bei der Berechnung der Auswahlwahrscheinlichkeiten gleich gewichtet, d.h. es wird $\alpha = 1$ und $\beta = 1$ gewählt. Die Verdunstungsrate beträgt $\rho = 0.01$. Für die Pseudo-random Proportional Rule wird $q_0 = 0.9$ gewählt, d.h. bei 90% aller Entscheidungen planen wir als nächsten Job denjenigen, der das maximale Produkt aus Pheromonwert und Heuristikwert hat. Die Anzahl der Ameisen ist $m = 20$. In einer Generation aktualisieren jeweils die beste Ameise der Generation und die global beste Ameise die Pheromonmatrix. Der Aktualisierungswert einer Ameise ist jeweils $\frac{\rho}{2 \cdot T^*}$, wobei T^* die Lösungsqualität der Ameise ist. Jeder Lauf wurde nach 4000 Generationen beendet. Alle Messungen wurden auf einem Pentium III 500MHz Prozessor durchgeführt. Ein Lauf über 4000 Generationen dauert ca. 100 Sekunden.

Ergebnisse

Die Ergebnisse für unterschiedliche Kombinationen der Werte c (Einfluss der lokalen Pheromonauswertung) und γ (Einfluss der Pheromonwerte früherer Entscheidungen) sind in Tabelle 3.2 und Abbildung 3.7 dargestellt. Angegeben sind die durchschnittlichen prozentualen Abweichungen von den bisher besten bekannten Lösungen. Die geringste prozentuale Abweichung wurde bei $c = 0.6$ und $\gamma = 0.9$ erzielt, d.h. die lokale Pheromonauswertung hat einen leicht stärkeren Einfluss auf die Berechnung der Auswahlwahrscheinlichkeiten als die Summen-Pheromonauswertung. Pheromonwerte wurden mit 0.9^k gewichtet, wenn die dem Pheromonwert zugehörige Entscheidung k Plätze zurück liegt. Wir beobachten, dass für einen stärker werdenden Einfluss der

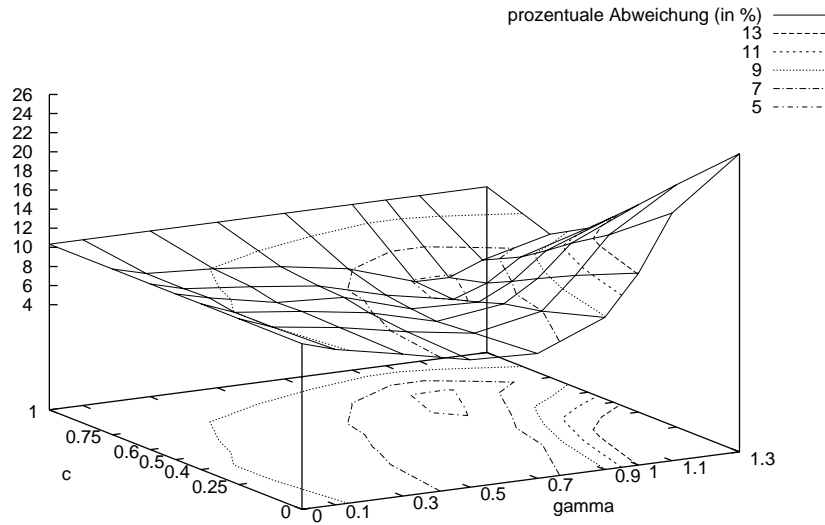


Abbildung 3.7: Ergebnisse des ACO-Algorithmus für SMTWTP für unterschiedliche Werte von c und γ ; dargestellt ist die durchschnittliche prozentuale Abweichung von den bisher besten bekannten Lösungen

lokalen Pheromonauswertung höhere (aber nicht zu hohe) Werte für γ zu bevorzugen sind. Beispielsweise erzielen wir das beste Ergebnis bei der reinen Summen-Pheromonauswertung ($c = 0$), wenn wir $\gamma = 0.7$ wählen. Bei einem starken Einfluss der lokalen Pheromonauswertung ($c = 0.75$) erzielen wir die besten Ergebnisse mit $\gamma = 1.0$.

Mit der reinen Summen-Pheromonauswertung wurde eine durchschnittliche prozentuale Abweichung von 13.1% von den bisher besten bekannten Lösungen erzielt. Dies ist schlechter als das Ergebnis, welches mit der lokalen Pheromonauswertung erzielt wurde (10.3%). Ein zu hoher Wert für γ hat einen negativen Einfluss wenn c klein wird und somit die Summen-Pheromonauswertung einen starken Einfluss hat.

Die hier vorgestellten Ergebnisse beziehen sich alle auf den Aspekt der globalen Nutzung der Pheromoninformation. Es sei an dieser Stelle erwähnt, dass mit dem hier vorgestellten Algorithmus in Kombination mit einer lokalen Suche und mehr Rechenzeit für alle Probleminstanzen die bisher beste Lösung gefunden wurde. Für die lokale Suche wurde eine einfache 2-opt Strategie verwendet. Die besten Ergebnisse wurden immer mit einer Kombination aus

Tabelle 3.2: Ergebnisse ACO-SMTWTP für unterschiedliche Werte von c und γ ; die Ergebnisse sind Mittelwerte der 125 Probleminstanzen; maximal 4000 Generationen mit $m = 20$ Ameisen; beste Ergebnisse jeder Spalte sind fett; bestes Ergebnis in der Tabelle ist unterstrichen

γ	c					
	0.0	0.25	0.4	0.5	0.6	0.75
0.0	10.3	10.3	10.3	10.3	10.3	10.3
0.1	9.3	9.0	8.9	9.4	9.4	9.4
0.3	7.8	8.0	8.2	8.1	8.7	9.0
0.5	6.3	6.7	7.1	7.7	7.8	8.3
0.7	6.1	5.6	5.3	5.5	5.9	7.1
0.9	8.9	7.0	6.2	5.3	<u>4.6</u>	4.9
1.0	13.1	10.2	8.4	7.0	5.8	4.8
1.1	18.9	14.0	11.6	9.8	8.1	6.2
1.3	24.2	18.3	14.7	12.4	10.2	8.0

lokaler und Summen-Pheromonauswertung erzielt. Während z.B. in (Crauwels et al., 1998) mit einer Tabusuche nur für 103 der 125 Probleminstanzen die bisher besten bekannten Lösungen gefunden werden konnten, wurden auch in der Arbeit (den Besten et al., 2000) mit einem ACO-Algorithmus mit lokaler Pheromonauswertung und einer komplexeren lokalen Suche alle bisher besten Ergebnisse erzielt. Somit zählen ACO-Algorithmen für das vorgestellte Ein-Maschinen-Planungsproblem zu den besten Verfahren überhaupt.

3.2 Relative Pheromonauswertung

In diesem Abschnitt stellen wir eine weitere globale Pheromonauswertungsmethode vor. Nach einer Einführung und der Untersuchung der Verzerrung der Auswahlwahrscheinlichkeiten für die relative Pheromonauswertung in Abschnitt 3.2.1 werden wir die Methode sowohl auf einfachen Problemen (Abschnitt 3.2.2) als auch auf einem NP-vollständigen Problem (Abschnitt 3.2.3) untersuchen.

3.2.1 Motivation und Vorgehensweise

Die relative Pheromonauswertung wird durch folgende Beobachtung motiviert. Angenommen ein Element j wurde während der Lösungskonstruktion noch nicht ausgewählt, obwohl ein Großteil des gesamten Pheromons der Spalte j auf den Pheromonwerten der ersten $i - 1$ Zeilen $(\tau_{1j}, \dots, \tau_{i-1,j})$ liegt. In diesem Fall sind alle Werte τ_{kj} mit $k \geq i$ klein. Daher ist es unwahrscheinlich, dass die Ameise das Element j auf Platz i wählt, wenn einem anderen Element in der i -ten Zeile ein hoher Pheromonwert zugeordnet ist. Dies gilt selbst dann noch, wenn τ_{ij} der größte der restlichen Pheromonwerte $\tau_{ij}, \dots, \tau_{nj}$ ist. Für manche Optimierungsprobleme kann es jedoch wichtig sein, auf welchem der Plätze $k \geq i$ Element j gewählt wird, obwohl hinsichtlich der kleinen Pheromonwerte alle Platzierungen nicht sehr viel versprechend erscheinen. Um bei der Berechnung der Auswahlwahrscheinlichkeiten einen Pheromonwert in Relation zur noch verfügbaren Pheromonmenge der restlichen Entscheidungen zu setzen, führen wir die relative Pheromonauswertung ein. Zur Berechnung der Auswahlwahrscheinlichkeiten verwenden wir nicht die einzelnen Pheromonwerte direkt, sondern wir gewichten den Pheromonwert umso mehr, je kleiner die Summe aller Pheromonwerte der noch nicht entschiedenen Plätze ist. Die Auswahlwahrscheinlichkeiten berechnen sich bei der relativen Pheromonauswertung wie folgt. Es sei

$$r_{ij} = \frac{\sum_{k=i}^n \tau_{kj}}{\sum_{k=1}^n \tau_{kj}}$$

der Anteil des Pheromons aller noch nicht entschiedenen Plätze an der gesamten Pheromonmenge einer Spalte. Dann berechnen wir

$$p_{ij} = \frac{\tau_{ij}^* \cdot \eta_{ij}}{\sum_{h \in \mathcal{S}} \tau_{ih}^* \cdot \eta_{ih}} \quad \text{wobei} \quad \tau_{ij}^* := \frac{1}{(r_{ij})^\gamma} \cdot \tau_{ij} \quad (3.7)$$

Mit dem Parameter γ steuern wir die Stärke der Normierung der Pheromonwerte. Für $\gamma = 0$ erhalten wir die lokale Pheromonauswertung. Die Variante, die wir für $\gamma = 1$ erhalten, bezeichnen wir als reine relative Pheromonauswertung.

Bei Verwendung der relativen Pheromonauswertung machen wir eine interessante Beobachtung. Falls wir wie üblich alle der besten Lösungspermutation zugehörigen Stellen in der Pheromonmatrix mit dem gleichen Wert aktualisieren (vgl. Algorithmus 1.1 auf Seite 5), dann werden kleine Pheromonwerte relativ gesehen stärker erhöht als große Pheromonwerte. Man beachte, dass

dieser Effekt auch bei der lokalen Pheromonauswertung auftritt und einen positiven Einfluss auf das Optimierungsverhalten haben kann, da kleine Pheromonwerte bezüglich der Auswahlwahrscheinlichkeit relativ stark anwachsen können, wenn die zugehörigen Entscheidungen Teil guter Lösungen sind. Im Gegensatz zur lokalen Pheromonauswertung kann es aber bei der relativen Pheromonauswertung vorkommen, dass zwei gleich große Pheromonwerte bei der Aktualisierung in unterschiedlichem Maße von der Aktualisierung profitieren. Je kleiner die Summe der Pheromonwerte der restlichen Plätze in der Pheromonmatrix ($\tau_{ij} + \dots + \tau_{nj}$) ist, desto größer ist der relative Einfluss der Aktualisierung an der Stelle τ_{ij} . Um den Einfluss dieses Effektes genauer zu untersuchen, führen wir eine Modifikation der relativen Pheromonauswertung ein, welche diesen Effekt umgeht. Da sich sowohl die Pheromonauswertung als auch die Pheromonaktualisierung von der Standardvorgehensweise unterscheidet, nennen wir die Methode relative Pheromonauswertung mit angepasster Pheromonaktualisierung. Bei dieser Methode wird die Menge des Pheromons bei der Aktualisierung so angepasst, dass der Wert — relativ zur Gesamtsumme der Pheromonwerte der noch zu entscheidenden Plätze — an jeder zu aktualisierenden Stelle identisch ist. Je größer die Restsumme der Pheromonwerte der noch zu entscheidenden Plätze ist, desto größer wird auch der Aktualisierungswert sein. Des Weiteren sollte es für alle Elemente τ_{lj} mit $l < i$ bei der Berechnung der Auswahlwahrscheinlichkeiten in der nächsten Iteration keinen Unterschied machen, in welcher Zeile $i > l$ die Aktualisierung durchgeführt wurde. Daher werden — nach der Erhöhung eines Werts τ_{ij} — alle Pheromonwerte τ_{lj} mit $l \geq i$ mit einem Faktor $\omega \geq 1$ so multipliziert, dass die Gesamtsumme der Aktualisierung auf allen Elementen τ_{lj} mit $l \geq i$ insgesamt genau $\frac{1}{\Delta}$ ist. Dabei wählen wir wie bei der Untersuchung der lokalen und der Summen-Pheromonauswertung $\Delta = \text{Kosten der besten Lösung} + 1$. Formal wird die Pheromonaktualisierung einer Entscheidung wie folgt in zwei Schritten durchgeführt. Nach der Berechnung von r_{ij} setzen wir zunächst

$$\tau_{ij} := \tau_{ij} + \frac{1}{\Delta} \cdot (r_{ij})^\gamma \quad (3.8)$$

und danach berechnen wir

$$\omega = \frac{\sum_{l=i}^n \tau_{lj} + \frac{1}{\Delta} (1 - (r_{ij})^\gamma)}{\sum_{l=i}^n \tau_{lj}}$$

und setzen

$$\tau_{lj} := \begin{cases} \omega \cdot \tau_{lj} & \text{für } l \geq i \\ \tau_{lj} & \text{sonst} \end{cases} \quad (3.9)$$

Wie man leicht verifizieren kann, ist die aktualisierte Pheromonmenge einer Spalte jeweils $\frac{1}{\Delta}$. In der ersten Zeile wird auf dem ausgewählten Element einer Zeile $\frac{1}{\Delta}$ aktualisiert, und die restlichen Pheromonwerte der Spalte bleiben unverändert. In der zweiten Zeile wird im Allgemeinen auf dem ausgewählten Element zunächst ein Wert kleiner als $\frac{1}{\Delta}$ aktualisiert, danach werden alle Pheromonwerte der entsprechenden Spalte um einen Faktor so erhöht, dass die Gesamtmenge an aktualisiertem Pheromon genau $\frac{1}{\Delta}$ ist. In Abbildung 3.8 ist zur Veranschaulichung der Aktualisierung bei Verwendung der relativen Pheromonauswertung mit angepasster Pheromonaktualisierung eine Pheromonmatrix eines Laufs nach nur einer Aktualisierung dargestellt. Die Pheromonmatrix wurde an allen Stellen mit dem gleichen Wert initialisiert, diese Stellen sind nach einem Schritt alle gleich groß und in schwarzer Farbe dargestellt. Deutlich erkennbar sind die höheren Aktualisierungswerte auf den Plätzen mit kleinem Index, da bei diesen Plätzen die Pheromon-Restsumme der Spalte noch hoch ist. Mit steigendem Index wird der Aktualisierungswert an den der Lösung entsprechenden Stellen geringer. Damit die in einer Spalte insgesamt hinzugefügte Pheromonmenge gleich bleibt, werden die restlichen Werte der jeweiligen Spalte mit zunehmendem Index einer Entscheidung immer mehr erhöht.

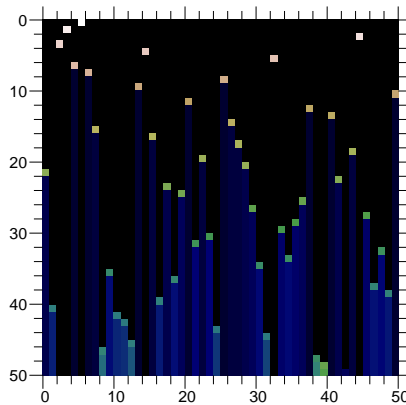


Abbildung 3.8: Pheromonmatrix bei Verwendung der relativen Pheromonauswertung mit angepasster Pheromonaktualisierung nach einer Aktualisierung; hellere Farben bedeuten höhere Pheromonwerte

Für die Untersuchung der Verzerrung der Auswahlwahrscheinlichkeiten wählen wir die üblichen Parameter (vgl. Abschnitt 2.4 und Abschnitt 3.1.1 für die Untersuchung der Verzerrung für die lokale und die Summen-Pheromonauswertung). Es sei m_{ij}^{Rel} die Anzahl der Ameisen, die bei Verwen-

dung der relativen Pheromonauswertung Element j auf Platz i gewählt haben, und es sei $d_{ij}^{\text{Rel}} := m_{ij}^{\text{Rel}} - m_{ij}^*$. Die Matrix der positiven Abweichungen $D^{\text{Rel}+} := [\max\{0, d_{ij}^{\text{Rel}}\}]$ und die Matrix der negativen Abweichungen $D^{\text{Rel}-} := [\min\{0, d_{ij}^{\text{Rel}}\}]$ sind in Abbildung 3.9 dargestellt. Die maximale positive Abweichung sinkt im Vergleich zur lokalen Pheromonauswertung von 4246 auf 1733, die minimale negative Abweichung steigt von -22450 auf -3681 . Noch deutlicher fällt der Vergleich bei den Summen aller Abweichungen aus. Bei der lokalen Pheromonauswertung ist die Summe aller Abweichungen 351103, bei der relativen Pheromonauswertung ist die Summe aller Abweichungen 71877. Man beachte, dass die Summe aller positiven Abweichungen bis auf den Vorzeichenwechsel identisch zur Summe aller negativen Abweichungen ist. In Abbildung 3.9 ist zu erkennen, dass, ähnlich wie bei der Summen-Pheromonauswertung, die starken Abweichungen eher an Stellen nahe der Hauptdiagonalen auftreten. Dies ist auch sehr deutlich beim Vergleich der problemspezifischen gewichteten Abweichung zu erkennen. Die maximale gewichtete positive Abweichung der lokalen Pheromonauswertung für die Problem Instanz P_A beträgt 88823, und die maximale gewichtete positive Abweichung der relativen Pheromonauswertung beträgt mit 4335 unter 5% der maximalen gewichteten positiven Abweichung der lokalen Pheromonauswertung. Es sei an dieser Stelle wieder deutlich darauf hingewiesen, dass die Untersuchung der Verzerrung für eine fest gewählte Pheromonmatrix nur als Indiz und als Erklärungsansatz für ein verbessertes Optimierungsverhalten gewertet werden darf.

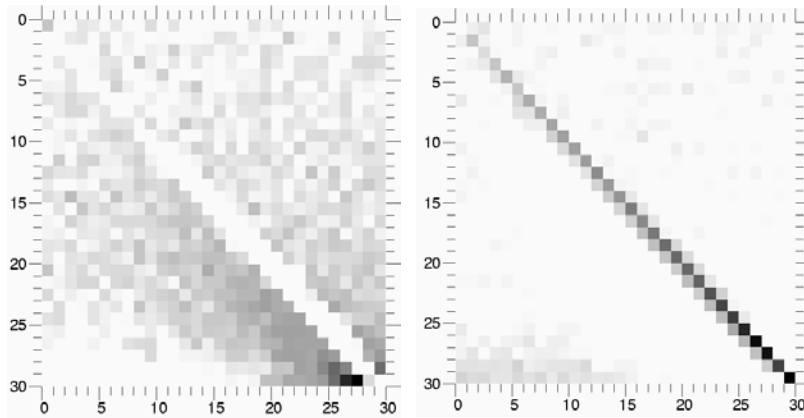


Abbildung 3.9: links: Matrix $D^{\text{Rel}+}$: weiß=0, schwarz=1733; rechts: Matrix $D^{\text{Rel}-}$: weiß=0, schwarz=-3681

In den nächsten Abschnitten werden wir die beiden Varianten der relativen Pheromonauswertung wieder sowohl auf ein einfaches Problem als auch auf ein NP-vollständiges Problem anwenden.

3.2.2 Einfache Probleminstanzen

Eine einfache Probleminstanz — P_{MF}

Zur Untersuchung der relativen Pheromonauswertung wählen wir ein Problem, bei dem es für einzelne Elemente mehrere Plätze in der Lösungspermutation gibt, die minimale Kosten erzeugen. Der Kostenanteil, den ein Element zu den Gesamtkosten beiträgt, ist der minimale Abstand zu allen optimalen Platzierungen des Elements (um 1 erhöht). Die Kostenmatrix des Problems ist in Abbildung 3.10 dargestellt. Jedes Element hat mindestens 2, aber maximal 3 optimale Platzierungsmöglichkeiten, die jeweils Kosten 1 erzeugen. Für die Probleminstanz existieren mehrere optimale Lösungen mit Kosten 50.

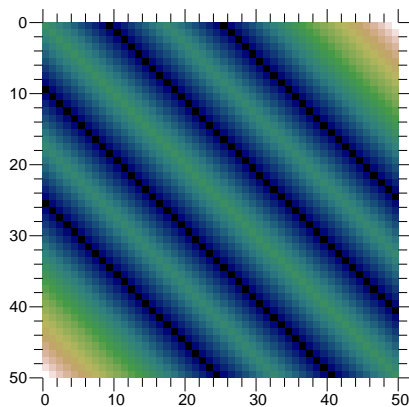


Abbildung 3.10: Kostenmatrix der Probleminstanz P_{MF} (mehrere Fälligkeitstermine); hellere Farben bedeuten höhere Kosten

Als Planungsproblem interpretiert entspricht dieses Problem einem Ein-Maschinen-Planungsproblem mit n Jobs, deren Laufzeit jeweils $p_j = 1$ beträgt. Jeder Job j hat mehrere Fälligkeitstermine. Die Kosten, die pro Job entstehen, werden durch die absolute minimale Abweichung zu den Fälligkeitsterminen berechnet. Wir bezeichnen die Probleminstanz mit P_{MF} (mehrere Fälligkeitstermine).

Die Entwicklung der Pheromonmatrizen bei Verwendung der lokalen Pheromonauswertung, der Summen-Pheromonauswertung und der beiden relativen Pheromonauswertungsmethoden ist jeweils in den Abbildungen 3.11, 3.12,

3.13 und 3.14 dargestellt. Die durchschnittlich erreichte Lösungsqualität beträgt 109.0 nach durchschnittlich 3200 Generationen bei Verwendung der lokalen Pheromonauswertung, 55.4 nach durchschnittlich 2657 Generationen bei Verwendung der relativen Pheromonauswertung und 159.6 nach durchschnittlich 5980 Generationen bei Verwendung der Summen-Pheromonauswertung. Bei Verwendung der relativen Pheromonauswertung mit angepasster Pheromonaktualisierung wird eine durchschnittliche Lösungsqualität von 71.4 nach durchschnittlich 2951 Generationen erreicht.

Wie bei der Problem Instanz P_A beobachten wir bei der lokalen Pheromonauswertung eine frühe Konvergenz in den ersten Zeilen. Für die Problem Instanz P_{MF} führt dieses Verhalten zu einer schlechten Lösungsqualität, da es viele Lösungspermutationen gibt, die zwar auf den Plätzen mit kleinen Indizes sehr gut sind, jedoch dann nicht zu guten Lösungen erweitert werden können, weil die restlichen Entscheidungen mit hohen Indizes eher hohe Kosten verursachen.

Bei der relativen Pheromonauswertung mit angepasster Aktualisierung ist die ungleichmäßige Konvergenz nicht zu beobachten, da relativ kleine Pheromonwerte in den unteren Zeilen (also solche mit hohem Index) unter Umständen zu hohen Auswahlwahrscheinlichkeiten führen können. Daher kann eine Entscheidung, die sich auf Plätzen mit kleinem Index in der Anfangsphase des Algorithmus durchgesetzt hat, besser als bei der Verwendung der lokalen Pheromonauswertung wieder revidiert werden.

Bei der relativen Pheromonauswertung mit angepasster Aktualisierung ist im Gegensatz zu den anderen Pheromonauswertungsmethoden zu beachten, dass durch die Anpassung die einzelnen Zeilensummen einer Pheromonmatrix nicht mehr identisch bleiben. Eine Entscheidung der besten Ameise verursacht eine Pheromonaktualisierung auf allen Plätzen, die einen höheren (oder gleich hohen) Index haben als der Index der Entscheidung. In Zeilen mit hohem Index wird im Allgemeinen mehr aktualisiert, da bei diesen Zeilen für nahezu jedes Element der zugehörige Pheromonwert erhöht wird (vgl. Gleichung (3.9) und Abbildung 3.8). Daher werden die Zeilensummen in den letzten Zeilen insbesondere am Anfang eines Messlaufs hoch sein.

Die angepasste Variante der relativen Pheromonauswertung wirkt sich im Vergleich zur normalen relativen Pheromonauswertung weder auf die erreichte Lösungsqualität noch auf die Konvergenzgeschwindigkeit positiv aus, liefert jedoch im Vergleich zur lokalen Pheromonauswertung bessere Ergebnisse. Die schlechteste Auswertungsmethode ist auf der Problem Instanz P_{MF} die (reine) Summen-Pheromonauswertung.

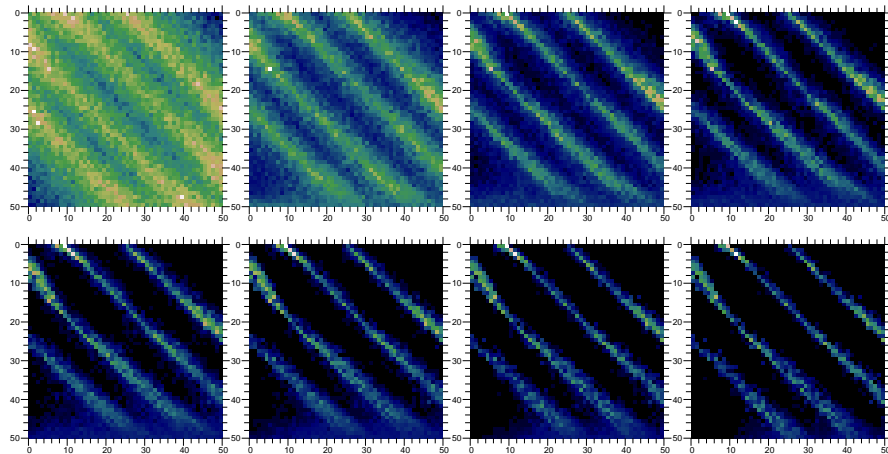


Abbildung 3.11: Durchschnittliche Pheromonmatrizen der Messläufe auf der Probleminstanz P_{MF} ; dargestellt sind die Pheromonmatrizen in den Generationen 400, 800, ..., 3200 bei Verwendung der lokalen Pheromonauswertung; hellere Farben bedeuten höhere Pheromonwerte

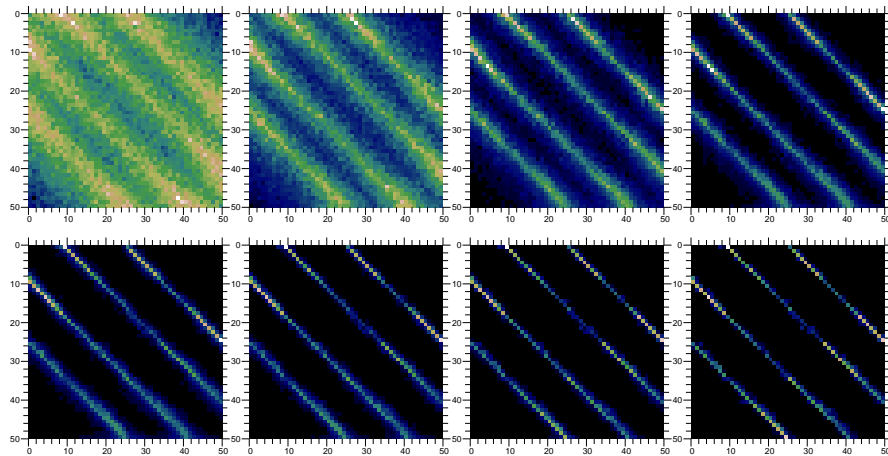


Abbildung 3.12: Durchschnittliche Pheromonmatrizen der Messläufe auf der Probleminstanz P_{MF} ; dargestellt sind die Pheromonmatrizen in den Generationen 400, 800, ..., 3200 bei Verwendung der relativen Pheromonauswertung; hellere Farben bedeuten höhere Pheromonwerte

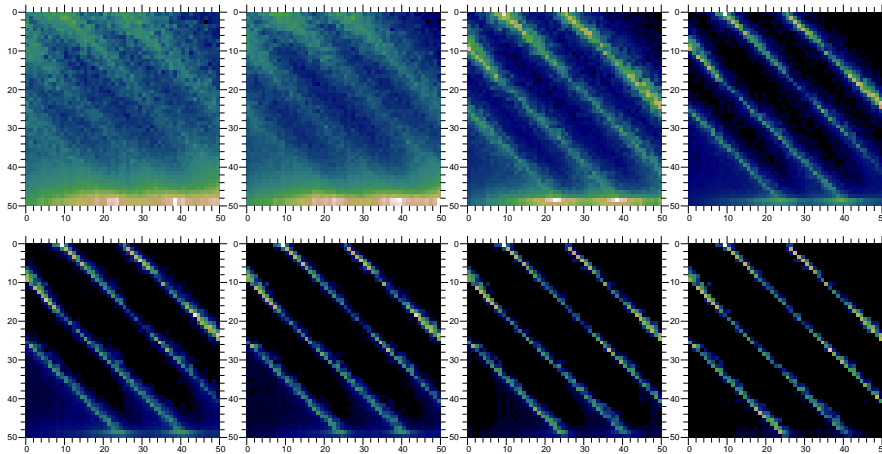


Abbildung 3.13: Durchschnittliche Pheromonmatrizen der Messläufe auf der Probleminstanz P_{MF} ; dargestellt sind die Pheromonmatrizen in den Generationen 100, 500, 1000, 1500, 2000, 2500, 3000, 3500 bei Verwendung der relativen Pheromonauswertung mit angepasster Pheromonaktualisierung; hellere Farben bedeuten höhere Pheromonwerte

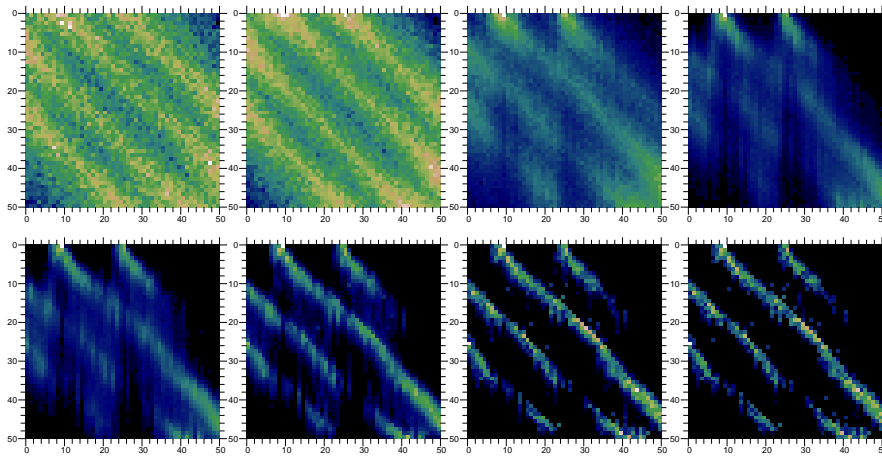


Abbildung 3.14: Durchschnittliche Pheromonmatrizen der Messläufe auf der Probleminstanz P_{MF} ; dargestellt sind die Pheromonmatrizen in den Generationen 100, 500, 1000, 2000, 3000, 4000, 6000, 8000 bei Verwendung der Summen-Pheromonauswertung; hellere Farben bedeuten höhere Pheromonwerte

3.2.3 Ein NP-vollständiges Problem

Als NP-vollständiges Testproblem zur Untersuchung der relativen Pheromonauswertung verwenden wir ein Ein-Maschinen-Planungsproblem, bei dem das Ziel die Minimierung von Verfrühungs- bzw. Lagerungskosten ist, die entstehen, wenn Jobs zu früh fertig gestellt werden. Wir betrachten ein Problem mit mehreren Fälligkeitsterminen, d.h. die entstehenden Kosten eines Jobs sind immer die Lagerungskosten bis zum nächsten Fälligkeitstermin. Motiviert ist dieses Problem durch eine Situation, bei der hergestellte Güter beispielsweise nur zum Monatsanfang (oder auch zu unregelmäßigeren Terminen) von einem Spediteur abgeholt werden und daher entsprechende Lagerungskosten entstehen. Das Problem erhält die Bezeichnung SMTEMDP (single machine total earliness multiple due date problem). Man beachte, dass schon der Fall mit nur einem Fälligkeitstermin ein NP-vollständiges Problem ist.

Formal haben wir n Jobs, wobei Job j , $1 \leq j \leq n$ eine Laufzeit p_j hat. Jedem Job sind n_j viele Fälligkeitstermine $d_{j_1} \geq \dots \geq d_{j_{n_j}}$ zugeordnet. Wir suchen einen unterbrechungsfreien Ein-Maschinen-Belegungsplan, welcher die Zielfunktion $E = \sum_{j=1}^n \min\{d_{j_k} - C_j \mid d_{j_k} \geq C_j\}$ minimiert, wobei C_j der Fertigstellungszeitpunkt des Jobs j ist. Um für jeden Job die Kosten berechnen zu können, gehen wir davon aus, dass der letzte Fälligkeitstermin der Summe aller Laufzeiten entspricht, d.h. $d_{j_{n_j}} = \sum_{j=1}^n p_j$.

Heuristik für SMTEMDP

Als Heuristik für SMTEMDP verwenden wir

$$\eta_{ij} = \frac{1}{\min\{d_{j_k} - (\mathcal{T} + p_j) \mid d_{j_k} \geq \mathcal{T} + p_j\} + 1} \quad (3.10)$$

wobei \mathcal{T} die Gesamtlaufzeit aller bis zum Platz $i - 1$ schon geplanten Jobs ist. Die Idee dieser Heuristik ist, dass Ameisen die Jobs bevorzugen sollen, die beim Einplanen des nächsten Jobs möglichst kleine Kosten verursachen.

Probleminstanzen und Parameter

Die Testinstanzen für SMTEMDP wurden wie folgt erzeugt. Die Probleminstanzgröße ist jeweils $n = 100$. Jeder Job hat eine ganzzahlige Laufzeit p_j , die zufällig gleichverteilt aus dem Intervall $[1, 10]$ gewählt wurde. Die ganzzahligen Fälligkeitstermine wurden zufällig gleichverteilt aus dem Intervall $[1, \sum_{j=1}^n p_j]$ gewählt. Wir haben jeweils 4 Instanzen mit 2, 3, 5 und 10 Fälligkeitsterminen (plus einen Pseudo-Fälligkeitstermin am Ende des Plans) erzeugt. Die beim

Tabelle 3.3: Vergleich von vier unterschiedlichen Pheromonauswertungsmethoden und unterschiedlich vielen Fälligkeitsterminen (FT); angegeben sind die durchschnittlich erzielten Kosten (bestes Ergebnis fett) für SMTEMDP ohne Verwendung einer Heuristik ($\beta = 0$); geklammert angegeben ist die durchschnittliche Anzahl an Generationen, bei der die beste Lösung gefunden wurde; relative* PA bezeichnet die relative Pheromonauswertung mit angepasster Pheromonaktualisierung

$\beta = 0.0$				
# FT	lokale PA	Summen-PA	relative PA	relative* PA
2	3782.5 (5025)	4281.0 (8314)	1710.0 (6994)	3097.7 (5300)
3	2441.1 (5068)	3496.4 (8433)	1182.7 (7001)	2087.2 (5368)
5	1793.2 (5082)	2420.8 (8266)	862.4 (7363)	1547.4 (5675)
10	964.2 (5537)	1317.0 (8527)	512.9 (7453)	781.2 (5731)

ACO-Algorithmus gewählten Parameter sind $\alpha = 1$, $\gamma = 1$, $\rho = 0.01$ und die Anzahl der Ameisen pro Generation ist $m = 10$. Jeder Lauf wurde für jede Parameterkombination 10 mal wiederholt und nach 15000 Generationen gestoppt. Die Aktualisierung wurde in jeder Generation von der besten Ameise der Generation durchgeführt, der Aktualisierungswert betrug $\frac{1}{E^*}$, wobei E^* die Lösungsqualität der Ameise war. Um die beiden relativen Pheromonauswertungen mit und ohne Heuristik zu untersuchen, wurden alle Tests mit $\beta = 1$ und mit $\beta = 0$ durchgeführt.

Ergebnisse

Die Ergebnisse für unterschiedlich viele Fälligkeitstermine sind in den Tabellen 3.3 und 3.4 dargestellt.

In allen getesteten Fällen erzielte der ACO-Algorithmus mit Verwendung der relativen Pheromonauswertung die besten Ergebnisse. Die Ergebnisse sind deutlich besser als bei Verwendung der lokalen Pheromonauswertung. Die Verbesserung beträgt zwischen 46.8% und 54.8% ohne Verwendung einer Heuristik und zwischen 38.4% und 41.9% bei Verwendung der Heuristik. Die durchschnittliche Anzahl der Generationen, bis die beste Lösung gefunden wurde, ist ohne Verwendung einer Heuristik maximal 44.9% und bei Verwendung einer Heuristik maximal 24.1% größer als bei Verwendung der lokalen Pheromonauswertung.

Tabelle 3.4: Vergleich von vier unterschiedlichen Pheromonauswertungsmethoden und unterschiedlich vielen Fälligkeitsterminen (FT); angegeben sind die durchschnittlich erzielten Kosten (bestes Ergebnis fett) für SMTEMDP mit Verwendung der Heuristik ($\beta = 1$); geklammert angegeben ist die durchschnittliche Anzahl an Generationen, bei der die beste Lösung gefunden wurde; relative* PA bezeichnet die relative Pheromonauswertung mit angepasster Pheromonaktualisierung

$\beta = 1.0$				
# FT	lokale PA	Summen-PA	relative PA	relative* PA
2	755.1 (3261)	534.1 (5062)	460.2 (3292)	498.7 (3260)
3	555.0 (2783)	550.5 (5933)	341.7 (3453)	389.3 (3277)
5	353.6 (3207)	271.6 (6579)	211.3 (3979)	254.4 (3730)
10	236.9 (3864)	199.2 (7622)	137.7 (4401)	158.4 (3833)

In allen getesteten Fällen war die angepasste Variante nach der relativen Pheromonauswertung die zweitbeste Methode. Die durchschnittliche Anzahl der Generationen, bis die beste Lösung gefunden wurde, war bei der angepassten Variante immer niedriger. Die Reihenfolge bezüglich der Lösungsqualität ist auf diesem Problem identisch zur Reihenfolge, die wir auf der einfachen Problem Instanz P_{MF} beobachtet haben. Die angepasste relative Pheromonauswertung hat im Vergleich zur reinen relativen Pheromonauswertung auf keinem der untersuchten Probleme bessere Ergebnisse erzielt. Dies ist ein Hinweis darauf, dass die lokale Pheromonauswertung verbessert werden könnte, indem man die Pheromonaktualisierung für einzelne Entscheidungen unterschiedlich stark macht. Üblicherweise verwenden ACO-Algorithmen für alle Entscheidungen die gleiche Menge an Pheromon für die Aktualisierung.

Ohne Verwendung der Heuristik ($\beta = 0$) liefert die Summen-Pheromonauswertung die deutlich schlechtesten Ergebnisse. Dies ist nicht überraschend, denn die Problemstruktur passt nicht zur Summen-Pheromonauswertung (vgl. Ergebnisse der Summen-Pheromonauswertung auf der einfachen Problem Instanz P_{MF} in Abbildung 3.14). Wird eine Heuristik verwendet, dann konzentriert der ACO-Algorithmus mit Summen-Pheromonauswertung die Suche auf wenige Plätze (entsprechend wenige Fälligkeitstermine) und ist besser als die lokale Pheromonauswertung (jedoch werden ca. doppelt so viele Iterationen benötigt).

Abbildungen 3.15 und 3.16 zeigen die Entwicklung der durchschnittlichen

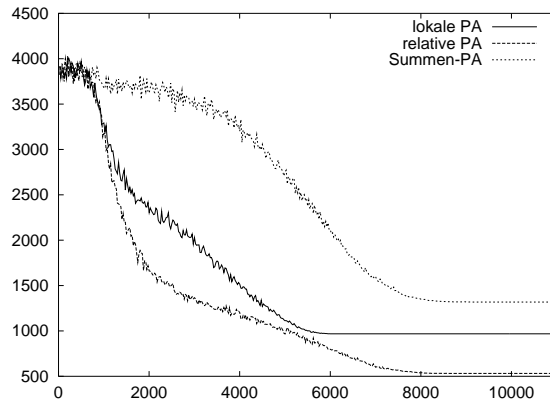


Abbildung 3.15: Durchschnittliche Lösungsqualität bei 10 Fälligkeitsterminen, $\beta = 0$ und drei unterschiedlichen Methoden zur Pheromonauswertung (PA)

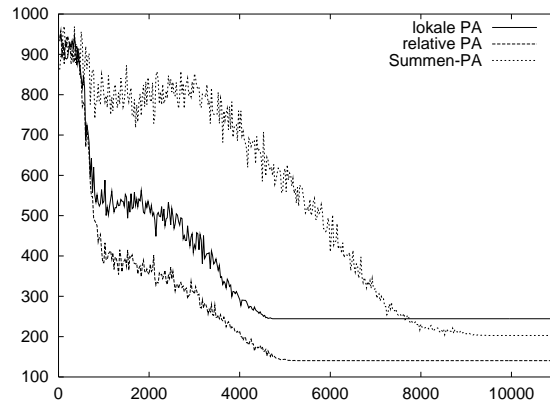


Abbildung 3.16: Durchschnittliche Lösungsqualität bei 10 Fälligkeitsterminen, $\beta = 1$ und drei unterschiedlichen Methoden zur Pheromonauswertung (PA)

Lösungsqualität im Falle von 10 Fälligkeitsterminen. Nur in der Anfangsphase verhalten sich die Algorithmen etwa gleich. Danach erhalten wir mit der Summen-Pheromonauswertung meist die schlechtesten Ergebnisse. Nur bei $\beta = 1$ werden mit der Summen-Pheromonauswertung am Ende, wenn der ACO-Algorithmus mit lokaler Pheromonauswertung schon konvergiert ist, noch bessere Ergebnisse erzielt. Dies bestätigt die Vermutung, dass die Verwendung der reinen Summen-Pheromonauswertung nicht für Planungsprobleme mit vielen Fälligkeitsterminen geeignet ist. Die relative Pheromonauswertung liefert nach

der Anfangsphase sowohl für $\beta = 0$ als auch für $\beta = 1$ während des gesamten Laufs im Mittel die besten Ergebnisse.

3.3 Zusammenfassung

In diesem Kapitel wurden zwei globale Pheromonauswertungsmethoden vorgestellt: die Summen-Pheromonauswertung und die relative Pheromonauswertung. Die beiden Methoden wurden bezüglich der Verzerrung der Auswahlwahrscheinlichkeiten untersucht. Für beide Varianten konnte eine geringere Verzerrung als bei der lokalen Pheromonauswertung festgestellt werden. Das Optimierungsverhalten wurde anhand einfacher Probleminstanzen untersucht. Die Resultate zeigten, dass sowohl die Summen-Pheromonauswertung als auch die relative Pheromonauswertung für bestimmte Problemklassen bessere Ergebnisse als die lokale Pheromonauswertung des Standard ACO-Algorithmus liefert. Das verbesserte Optimierungsverhalten wurde anhand von Untersuchungen an NP-vollständigen Problemen bestätigt.

Kapitel 4

Entscheidungsreihenfolgen

Die in den vorherigen Kapiteln verwendete Entscheidungsreihenfolge für die Lösungskonstruktion bei NP-vollständigen Ein-Maschinen-Planungsproblemen war immer die „intuitive“, d.h. die erste (oder i -te) Entscheidung einer Ameise entsprach auch immer dem ersten (oder i -ten) Job, der auf eine Maschine eingeplant wurde. Dies ist auch die in der Literatur übliche Vorgehensweise, um mit ACO-Algorithmen Planungsprobleme zu lösen. In diesem Kapitel wollen wir von dieser anschaulichen aber nicht unbedingt notwendigen Verfahrensweise abweichen. Wir werden untersuchen, wie sich unterschiedliche Entscheidungsreihenfolgen auf das Optimierungsverhalten von ACO-Algorithmen auswirken. In Abschnitt 4.1 werden wir unter anderem die Methode der zufällig gewählten Entscheidungsreihenfolge einführen und hinsichtlich der Verzerrung der Auswahlwahrscheinlichkeiten untersuchen. Des Weiteren werden wir ein algorithmisches Gerüst angeben, welches die unterschiedlichen Pheromonauswertungsmethoden und die verschiedenen Vorgehensweisen bezüglich der Entscheidungsreihenfolge vereinigt. In Abschnitt 4.2 werden die Varianten auf einfachen Permutationsproblemen und in Abschnitt 4.3 auf einem NP-vollständigen Problem untersucht.

4.1 Motivation und Vorgehensweise

In Kapitel 2 haben wir eine deutliche Verzerrung der Auswahlwahrscheinlichkeiten bei einem ACO-Algorithmus mit lokaler Pheromonauswertung und einer, wie in der Einleitung schon erwähnt, „intuitiven“ Entscheidungsreihenfolge beobachtet. Diese Verzerrung war bei dem in Abschnitt 2.4 vorgestellten Beispiel insbesondere auf Plätzen mit hohem Index zu erkennen, da auf den letzten Plätzen der Lösungspermutation die Menge der noch auszuwählenden

Elemente immer kleiner wird. Im letzten Kapitel wurde die Verzerrung durch eine globale Auswertung der Pheromone zur Berechnung der Auswahlwahrscheinlichkeiten deutlich abgeschwächt. In diesem Abschnitt werden wir eine Methode vorstellen, mit der wir das Verhalten eines ACO-Algorithmus hinsichtlich der Verzerrung der Auswahlwahrscheinlichkeiten deutlich verbessern können. Die neue Methode ist auf Probleme anwendbar, bei denen zwischen den einzelnen Elementen keine Vorrangbeziehungen bestehen. Solche Beziehungen gibt es beispielsweise bei Planungsproblemen, wenn einer Probleminstanz ein Präzedenzgraph zugeordnet ist, der angibt, welche Jobs vor oder nach welchen anderen Jobs eingeplant werden dürfen, um einen zulässigen Plan zu erzeugen. Alle in dieser Arbeit bisher untersuchten Probleme, sowohl die einfachen als auch die NP-vollständigen, erfüllen die Bedingung, dass keine Vorrangbeziehungen zwischen den Elementen oder Jobs bestehen.

Die grundsätzliche Idee der neuen Methode ist die folgende. Jede Ameise erzeugt zunächst eine Zufallspermutation der Zahlen $(1, \dots, n)$, wobei n wie üblich die Anzahl der Lösungskonstruktionsschritte ist. Die Entscheidung, auf welchem der noch vorhandenen Plätze der Lösungspermutation als nächstes ein Element gewählt wird, ergibt sich dann entsprechend der erzeugten Zufallspermutation. Der Vorteil der Verwendung einer solchen Zufallspermutation ist, dass auf jedem Platz der Lösungspermutation mit gleicher Wahrscheinlichkeit die erste (oder auch k -te) Entscheidung bei der Lösungskonstruktion gefällt wird (also ein Element gewählt wird). Im Gegensatz zu den Untersuchungen der lokalen und der Summen-Pheromonauswertung erwarten wir daher eine gleichmäßigere Verzerrung der Auswahlwahrscheinlichkeiten. Wie wir bei den Untersuchungen NP-vollständiger Probleme für die globalen Pheromonauswertungsmethoden gesehen haben, ist bei ACO-Algorithmen für Planungsprobleme der Heuristikwert η_{ij} oft von der bisher erzeugten partiellen Lösung der Ameise abhängig, da der Fertigstellungszeitpunkt des zuletzt eingeplanten Jobs oft in die Berechnung des Heuristikwerts η_{ij} eingeht. Wenn wir eine zufällige Permutation als Entscheidungsreihenfolge wählen, dann können wir im Allgemeinen nach der Einplanung eines Jobs dessen Fertigstellungszeitpunkt nicht wissen. Daher ist die direkte Verwendung eines Heuristikwerts zur Berechnung der Auswahlwahrscheinlichkeiten nur dann möglich, wenn der Heuristikwert nicht von der partiellen Lösung einer Ameise abhängig ist. Bei den Untersuchungen auf einfachen Probleminstanzen setzen wir wie bei den analogen Tests in den vorherigen Abschnitten keine Heuristik ein. Bei der Untersuchung eines NP-vollständigen Problems werden wir in Abschnitt 4.3 eine Methode angeben, mit der wir auch bei Verwendung von Zufallspermu-

tationen eine Heuristik verwenden können. Teile der Resultate dieses Kapitels sind in (Merkle und Middendorf, 2001c) und (Merkle und Middendorf, 2002f) erschienen.

Zunächst untersuchen wir, wie in den letzten Kapiteln auch, die Verzerrung der Auswahlwahrscheinlichkeiten. Wir verwenden wie üblich die in Abschnitt 2.4 definierte Pheromonmatrix der Größe $n = 30$ und lassen wieder $m = 100000$ Ameisen Lösungspermutationen mit dieser Pheromonmatrix erzeugen. Jedoch verwenden wir hier für jede Ameise eine Zufallspermutation, welche die Reihenfolge der Entscheidungen festlegt. Es sei m_{ij}^{Ran} die Anzahl der Ameisen, die bei diesem Verfahren Element j auf Platz i gewählt haben, und es sei $d_{ij}^{\text{R}} := m_{ij}^{\text{Ran}} - m_{ij}^*$. Die Matrix der positiven $D^{\text{Ran}+} := [\max\{0, d_{ij}^{\text{Ran}}\}]$ Abweichungen und die Matrix der negativen Abweichungen $D^{\text{Ran}-} := [\min\{0, d_{ij}^{\text{Ran}}\}]$ sind in Abbildung 4.1 dargestellt.

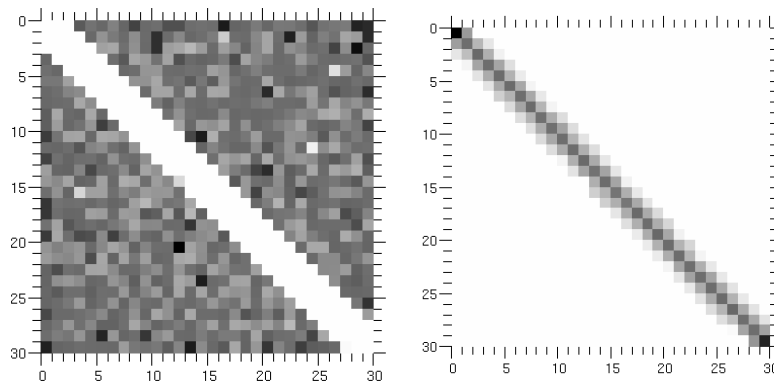


Abbildung 4.1: links: Matrix $D^{\text{Ran}+}$: weiß=0, schwarz=271; rechts: Matrix $D^{\text{Ran}-}$: weiß=0, schwarz=-3113

Wie beim Standard ACO-Algorithmus auch, beobachten wir die negativen Werte d_{ij}^{Ran} entlang der Hauptdiagonalen. Die zugehörigen Entscheidungen werden also zu selten gewählt, um keine Verzerrung der Auswahlwahrscheinlichkeiten zu erhalten. Die positiven Abweichungen sind aber, im Gegensatz zum Standard ACO-Algorithmus und auch im Gegensatz zu den Varianten mit globaler Pheromonauswertung, in der gesamten Matrix etwa gleich verteilt. Es besteht keine deutliche Verzerrung in einem bestimmten Gebiet der Pheromonmatrix. Das ist ein Vorteil gegenüber dem Standard ACO-Algorithmus, da die gleichmäßigere Verteilung darauf hindeutet, dass Ameisen mit zufällig gewählter Entscheidungsreihenfolge bessere Lösungen finden können.

Der Vergleich der positiven und negativen Abweichung gibt einen weite-

ren Hinweis auf die Stärke der Ameisen mit Zufallspermutation. Beim Standard ACO-Algorithmus war die Summe der positiven (oder bis auf den Vorzeichenwechsel negativen) Abweichungen 351103. Dies entsprach etwa 11.7% aller 3000000 Entscheidungen. Bei den Ameisen mit Zufallspermutation ist die Summe aller positiven Abweichungen 96416. Das sind nur etwa 3.2% aller 3000000 Entscheidungen. Die gewichtete positive (negative) Abweichung aller Entscheidungen bezogen auf die Instanz P_A (vgl. Abbildung 2.1 auf Seite 17) ist für den Standard ACO-Algorithmus 4297654 (negativ: -1206284). Die durchschnittliche Abweichung jeder dieser Entscheidungen ist somit 12.2 (negativ: -3.4). Bei Ameisen mit Zufallspermutation ist die gewichtete positive Abweichung aller Entscheidungen bezogen auf die Probleminstanz P_A 1231973 (negativ: -133052). Dies entspricht einer durchschnittlichen Abweichung von 12.7 (negativ: -1.4). Die durchschnittliche Abweichung ist bei Verwendung einer Zufallspermutation also ähnlich wie bei Verwendung der Permutation $(1, \dots, n)$. Der maximale positive (minimale negative) Wert für d_{ij} ist 4246 (negativ: -22450) bei Verwendung der üblichen Entscheidungsreihenfolge. Beim ACO-Algorithmus mit Zufallspermutation ist die maximale positive (minimale negative) Abweichung 271 (negativ: -3113). Der Vergleich zeigt deutlich, dass die Verwendung von Zufallspermutationen bei der Lösungskonstruktion die Verzerrung der Auswahlwahrscheinlichkeiten deutlich abschwächt.

Wie in den vorherigen Abschnitten, in denen wir die Verzerrung der Auswahlwahrscheinlichkeiten anhand einer Generation mit 100000 Ameisen ohne Heuristik untersucht haben, sei auch an dieser Stelle darauf hingewiesen, dass die Ergebnisse mit Vorsicht interpretiert werden müssen. Wenn das durchschnittliche Verhalten der Ameisen einer geringeren systematischen Verzerrung unterliegt, muss das nicht notwendigerweise bedeuten, dass der Algorithmus zu einem besseren Optimierungsverhalten führt.

In (Michels und Middendorf, 1999) wurde erstmals vorgeschlagen, sowohl *Vorwärts-Ameisen* als auch *Rückwärts-Ameisen* zur Erzeugung der Lösungen eines Optimierungsproblems zu verwenden. Mit Vorwärts-Ameisen bezeichnen wir Ameisen, bei denen die Entscheidungsreihenfolge $(1, 2, \dots, n)$ verwendet wird. Mit Rückwärts-Ameisen bezeichnen wir Ameisen, die immer die Entscheidungsreihenfolge $(n, n - 1, \dots, 1)$ bei der Lösungskonstruktion verwenden, d.h. als erste Entscheidung wird die der untersten Zeile der Pheromonmatrix entsprechende Entscheidung gefällt. In Kombination mit den unterschiedlichen Methoden der Pheromonauswertung erhalten wir den Algorithmus 4.2, der auf Seite 58 angegeben ist. Die Eingaben des Algorithmus

sind, neben den nicht explizit aufgeführten Eingabeparametern für die Steuerung des Einflusses der Pheromon- und Heuristikwerte α, β und der Verdunstungsrate ρ , der Parameter X für die Methode der Pheromonauswertung und der Parameter y für die Festlegung der Entscheidungsreihenfolge. Bei $X = L$ wird die lokale Pheromonauswertung verwendet, $X = S$ bezeichnet die Summen-Pheromonauswertung, und $X = R$ bzw. $X = R^*$ bezeichnen die relative Pheromonauswertung mit und ohne Anpassung der Pheromonaktualisierung. Bei $y = f$ werden Vorwärts-Ameisen verwendet, $y = r$ kehrt die Entscheidungsreihenfolge um, und bei $y = p$ wird für jede Ameise eine Zufallspermutation gewählt, welche die Entscheidungsreihenfolge festlegt. Man beachte, dass nicht alle möglichen Kombinationen aus Entscheidungsreihenfolge und Pheromonauswertung Sinn machen. Die Summen-Pheromonauswertung ist für Probleme gedacht, bei denen wir im Allgemeinen umso höhere Kosten für eine Entscheidung haben, je mehr diese von einer sehr guten Platzierung abweicht. Wie wir schon gesehen haben, bleiben bei der Summen-Pheromonauswertung Werte in der Pheromonmatrix pro Spalte in einem Gebiet aufeinander folgender Plätze hoch. Wenn nun nach einer Entscheidung auf Platz i nicht die Entscheidung auf Platz $i + 1$, sondern auf einem zufällig bestimmten Platz gefällt wird, dann führt die Summen-Pheromonauswertung zu einem nicht gewünschten Verhalten. Die Summen-Pheromonauswertung werden wir daher nicht in Kombination mit zufällig gewählten Entscheidungsreihenfolgen untersuchen. Wenn wir die Summen-Pheromonauswertung mit Rückwärts-Ameisen kombinieren, dann summieren wir die Pheromonwerte beginnend mit der letzten Zeile bis zur Zeile, in der die aktuelle Entscheidung getroffen wird. Die lokale und die relative Pheromonauswertung lassen sich mit allen Varianten der Entscheidungsreihenfolge kombinieren. Die relative Pheromonauswertung mit angepasster Aktualisierung lässt sich wie die Summen-Pheromonauswertung nicht mit zufällig gewählten Entscheidungsreihenfolgen kombinieren. Die Anpassung führt dazu, dass die Aktualisierung in den Zeilen unterschiedlich stark ist. Bei dieser Vorgehensweise wird davon ausgegangen, dass die Entscheidungsreihenfolge in der nächsten Generation gleich bleibt. Da dies bei einer zufällig gewählten Entscheidungsreihenfolge nicht der Fall ist, werden wir diese Kombination nicht untersuchen. Die Kombination der relativen Pheromonauswertung mit Rückwärts-Ameisen ist wie bei der Summen-Pheromonauswertung möglich.

Die Entscheidungsreihenfolge wird in Kombination mit den unterschiedlichen Methoden zur Pheromonauswertung im nächsten Abschnitt anhand einfacher Probleminstanzen genauer untersucht.

Algorithmus 4.2 erweiterter ACO-Algorithmus für Permutationsprobleme

Zusätzliche Parameter:

Pheromonauswertungsmethode: $X \in \{L, S, R, R^*\}$

Entscheidungsreihenfolge: $y \in \{f, r, p\}$

```

1: Initialisiere alle Pheromonwerte  $\tau_{ij}$ 
2: repeat
3:   for Ameise  $k \in \{1, \dots, m\}$  do
4:     if  $y = p$  then
5:       Permutiere für diese Ameise alle Zeilen der Pheromonmatrix und
       der Kostenmatrix entsprechend einer zufällig gewählten Permuta-
       tion der Zahlen  $1, \dots, n$ 
6:     end if
7:     if  $y = r$  then
8:       Kehre für diese Ameise die Reihenfolge der Zeilen in der Phero-
       monmatrix und in der Kostenmatrix um
9:     end if
10:     $S := \{1, 2, \dots, n\}$            //Menge der verfügbaren Elemente
11:    for  $i = 1$  to  $n$  do
12:      wähle Element  $j \in S$  mit Wahrscheinlichkeit  $p_{ij}$  entsprechend  $X$ 
13:       $S := S - \{j\}$ 
14:    end for
15:  end for
16:  for all  $(i,j)$  do
17:     $\tau_{ij} := (1 - \rho) \cdot \tau_{ij}$            //Verdunstung
18:  end for
19:  if  $X = R^*$  then
20:    führe eine Aktualisierung entsprechend der Gleichungen (3.8) und
    (3.9) auf Seite 40 durch
21:  else
22:    for all  $(i,j) \in$  beste Lösung do
23:       $\tau_{ij} := \tau_{ij} + \Delta(\rho, \text{Kosten der besten Lösung})$  //Aktualisierung
24:    end for
25:  end if
26: until Abbruchkriterium erreicht

```

4.2 Einfache Probleminstanzen

4.2.1 Die Probleminstanzen P_A und P_{MF}

Zunächst werden wir schon vorgestellte Probleminstanzen mit den uns bekannten Pheromonauswertungsmethoden in Kombination mit einer zufällig gewählten Entscheidungsreihenfolge kombinieren. Wir verwenden hierfür die beiden Probleminstanzen P_A und P_{MF} . In Tabelle 4.1 und 4.2 sind die Ergebnisse der unterschiedlichen Verfahren für die beiden Probleminstanzen P_A und P_{MF} aufgeführt. Wie üblich, sind die Ergebnisse auf den einfachen Probleminstanzen Mittelwerte aus 25 Testläufen. Die Summen-Pheromonauswertung liefert für die Probleminstanz P_A sehr gute Ergebnisse (vgl. Abschnitt 3.1.2). Es ist deutlich zu erkennen, dass sowohl die lokale Pheromonauswertung als auch die relative Pheromonauswertung so stark von der zufällig gewählten Entscheidungsreihenfolge profitieren können, dass nahezu immer die Optimallösung gefunden wird.

Tabelle 4.1: Ergebnisse für die Instanz P_A ; angegeben sind die durchschnittliche Lösungsqualität der besten Lösung und die durchschnittliche Generation, in der diese Lösung gefunden wurde

Auswertung		Probleminstanz P_A	
		bestes Ergebnis	Generation
lokale PA	Vorwärts	287.4	3456
	Zufall	50.4	3547
relative PA	Vorwärts	70.1	2421
	Zufall	50.1	3415
relative* PA	Vorwärts	192.56	3919
Summen-PA	Vorwärts	51.9	2932

In Abbildung 4.2 sind die durchschnittlichen Pheromonmatrizen für die lokale und die relative Pheromonauswertung bei Verwendung einer Zufallspermutation dargestellt. Wir beobachten bei beiden Verfahren keine ungleichmäßige Konvergenz und keine Verstärkung von Pheromonwerten an kostengünstigen Stellen. Obwohl sich die lokale und die relative Pheromonauswertung in Kombination mit Vorwärts-Ameisen bezüglich der Berechnung der Auswahlwahrscheinlichkeiten stark unterscheiden, führt die Kombination mit einer Zufallspermutation als Entscheidungsreihenfolge auf diesen beiden Pro-

Tabelle 4.2: Ergebnisse für die Instanz P_{MF} ; angegeben sind die durchschnittliche Lösungsqualität der besten Lösung und die durchschnittliche Generation, in der diese Lösung gefunden wurde

Auswertung		Probleminstanz P_{MF}	
		bestes Ergebnis	Generation
lokale PA	Vorwärts	109.0	3200
	Zufall	50.0	4493
relative PA	Vorwärts	55.4	2657
	Zufall	50.2	3941
relative* PA	Vorwärts	71.4	2951
Summen-PA	Vorwärts	159.6	5980

bleminstanzen zu sehr ähnlichen Ergebnissen bezüglich Lösungsqualität und Konvergenzverhalten. In Abbildung 4.2 ist nahezu kein Unterschied der durchschnittlichen Pheromonmatrizen auf der Probleminstanz P_A zu erkennen.

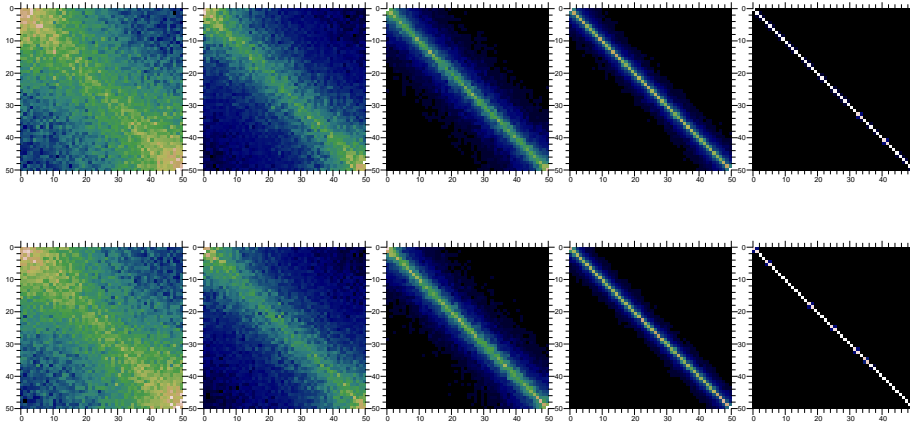


Abbildung 4.2: Durchschnittliche Pheromonmatrizen zweier Messläufe auf der Probleminstanz P_A ; dargestellt sind die Pheromonmatrizen in den Generationen 500, 1000, 2000, 3000, 4000 und 5000 bei Verwendung der lokalen Pheromonauswertung (oben) und der relativen Pheromonauswertung (unten) mit zufällig gewählter Entscheidungsreihenfolge; hellere Farben bedeuten höhere Pheromonwerte

4.2.2 Die Problem Instanz P_{MA}

Um den Einfluss von Vorwärts- und Rückwärts-Ameisen auf einfachen Problem Instanzen genauer zu untersuchen, führen wir eine neue einfache Problem Instanz ein. Wir modifizieren die Kostenmatrix der Problem Instanz P_A derart, dass die Entscheidungen auf den ersten $k < n$ Plätzen jeweils gleiche Kosten verursachen. Wir wählen die Kosten $c(i, j) = 1$ für alle $1 \leq i \leq 10$. Die optimale Lösung des Problems hat bei der Problemgröße $n = 50$ die Kosten 50. Man beachte, dass viele (genau $10!$) unterschiedliche optimale Lösungen existieren. Bei einer Lösungspermutation mit optimaler Lösungsqualität sind alle Elemente j mit $j > 10$ auf Platz j . Die 10 übrigen Elemente der ersten 10 Plätze können dann beliebig vertauscht werden, ohne dass die Kosten der Lösung verändert werden. Die Kostenmatrix für das Problem ist in Abbildung 4.3 abgebildet.

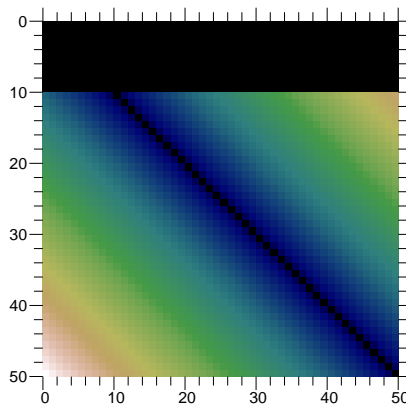


Abbildung 4.3: Kostenmatrix der Problem Instanz P_{MA} (modifizierte Abweichung); hellere Farben bedeuten höhere Kosten; die Kosten in den ersten 10 Zeilen der Matrix betragen bei allen Elementen jeweils 1

Die Resultate in Tabelle 4.3 zeigen, dass ähnlich wie bei der Problem Instanz P_A das beste Ergebnis mit der Summen-Pheromonauswertung, und das zweitbeste Ergebnis mit der relativen Pheromonauswertung mit Vorwärts-Ameisen gefunden wurde. Die einzelnen Methoden verhalten sich bezüglich der Einplanungsreihenfolgen ganz unterschiedlich. Während die Resultate der lokalen und der Summen-Pheromonauswertung stark von der Entscheidungsreihenfolge beeinflusst sind, ist die relative Pheromonauswertung bezüglich dieses Kriteriums auf dem gewählten Problem robuster. Die relative Pheromonauswertung ohne Anpassung liefert in Kombination mit Vorwärts-Ameisen im

Mittel sogar bessere Ergebnisse. Die relative Pheromonauswertung mit Anpassung der Aktualisierung liefert bessere Ergebnisse als der Algorithmus mit lokaler Pheromonauswertung, die Lösungsqualität ist jedoch im Vergleich zu den besten Varianten schlecht. Diese Untersuchung bezüglich der Entscheidungsreihenfolge bei der Elementauswahl könnte erklären, weshalb in der Arbeit (den Besten et al., 1999) die Beobachtung gemacht wurde, dass bei einem Planungsproblem mit Fälligkeitsterminen der verwendete ACO-Algorithmus (mit lokaler Pheromonauswertung) auf Instanzen mit vielen späten Fälligkeitsterminen ein schlechteres Optimierungsverhalten zeigt als auf Instanzen mit frühen Fälligkeitsterminen. Denn es scheint bei der lokalen und bei der Summen-Pheromonauswertung besser zu sein, wichtige Entscheidungen (solche, die bei der Auswahl des nächsten Elements sowohl hohe als auch niedrige Kosten verursachen können) früh zu fällen. Man beachte hierbei allerdings, dass für unsere Untersuchungen auf den einfachen Probleminstanzen im Gegensatz zu (den Besten et al., 1999) weder eine zusätzliche lokale Suche, noch Heuristikinformation verwendet wird.

Tabelle 4.3: Ergebnisse für die Instanz P_{MA} ; angegeben sind die durchschnittliche Lösungsqualität der besten Lösung und die durchschnittliche Generation, in der diese Lösung gefunden wurde

Auswertung		Probleminstanz P_{MA}	
		bestes Ergebnis	Generation
lokale PA	Vorwärts	270.5	3490
	Rückwärts	160.5	1668
relative PA	Vorwärts	62.3	2256
	Rückwärts	70.2	2345
relative* PA	Vorwärts	161.8	2934
	Rückwärts	101.5	4476
Summen-PA	Vorwärts	254.2	2791
	Rückwärts	50.6	2604

In Abbildung 4.4 sind für die lokale Pheromonauswertung die durchschnittlichen Pheromonmatrizen unter Verwendung von Vorwärts- und Rückwärts-Ameisen dargestellt. Die hohen Kosten, die bei Verwendung von Vorwärts-Ameisen in den letzten Zeilen zu erkennen sind, entstehen bei Verwendung der Rückwärts-Ameisen nicht, da bei den wichtigen Entscheidungen in den letzten Zeilen die Auswahlmenge an noch verfügbaren Elementen groß ist.

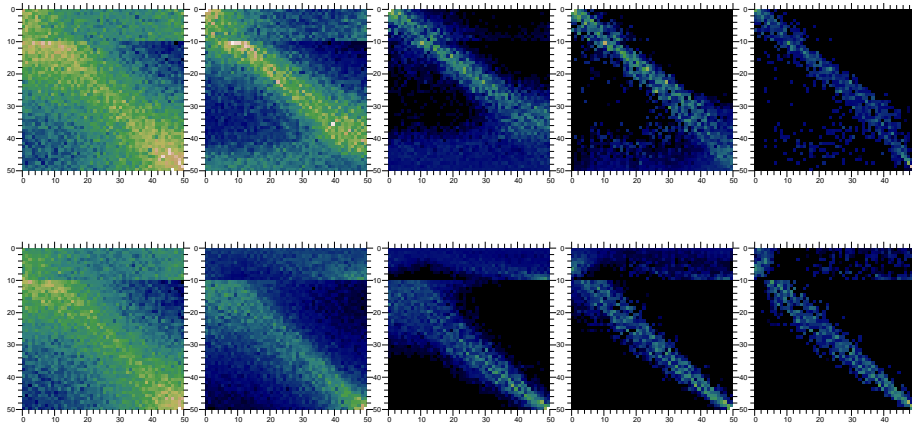


Abbildung 4.4: Durchschnittliche Pheromonmatrizen zweier Messläufe auf der Probleminstanz P_{MA} ; dargestellt sind die Pheromonmatrizen in den Generationen 500, 1000, 2000, 3000 und 4000 bei Verwendung der lokalen Pheromonauswertung und Vorwärts-Ameisen (oben) und Rückwärts-Ameisen (unten); hellere Farben bedeuten höhere Pheromonwerte

In Abbildung 4.5 ist das unterschiedliche Verhalten von Vorwärts- und Rückwärts-Ameisen für die Summen-Pheromonauswertung dargestellt. Zwar verhindert die Summen-Pheromonauswertung mit Vorwärts-Ameisen eine kostengünstige Erhöhung der Pheromonwerte in der linken unteren Ecke der Pheromonmatrix, jedoch werden viele Elemente mit hohem Spaltenindex auf den ersten Plätzen der Lösungspermutation gewählt. Dies führt zu sehr hohen Kosten, denn falls beispielsweise nur das Element mit Index 50 auf einem der ersten 10 Plätze gewählt wird, dann ist die noch zu erreichende beste Lösungsqualität 90 (optimale Lösungsqualität: 50). Die Erhöhung um 40 Kosteneinheiten entsteht dabei, weil 40 Elemente jeweils um einen Platz später gewählt werden müssen. Es verursachen also, im Gegensatz zur lokalen Pheromonauswertung, nicht wenige Elemente jeweils hohe Kosten, sondern viele Elemente jeweils niedrige Kosten. In Kombination mit Rückwärts-Ameisen tritt das beschriebene Verhalten nicht auf, und die Ameisen finden nahezu immer die optimale Lösung.

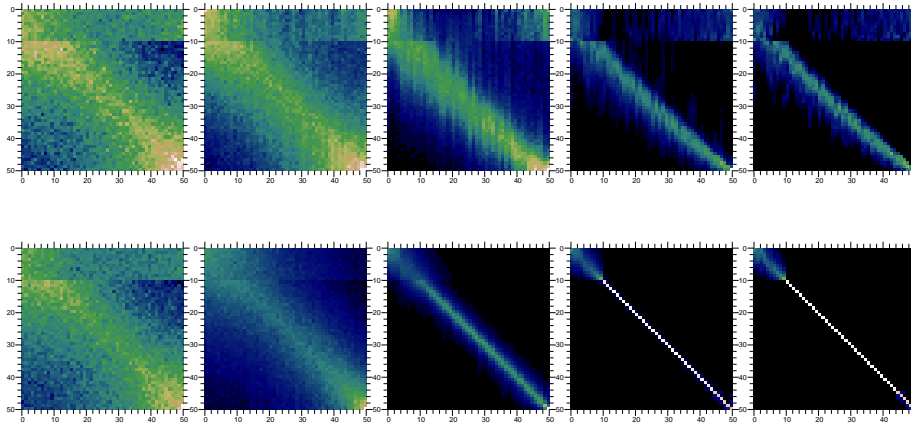


Abbildung 4.5: Durchschnittliche Pheromonmatrizen zweier Messläufe auf der Probleminstanz P_{MA} ; dargestellt sind die Pheromonmatrizen in den Generationen 500, 1000, 2000, 3000 und 4000 bei Verwendung der Summen-Pheromonauswertung und Vorwärts-Ameisen (oben) und Rückwärts-Ameisen (unten); hellere Farben bedeuten höhere Pheromonwerte

4.2.3 Die Probleminstanzen P_{P_k}

Alle bisher verwendeten einfachen Probleminstanzen (bis auf die Instanz P_D) haben die Eigenschaft, dass man sie sinnvoll als Instanzen eines sehr einfachen Planungsproblems interpretieren kann, bei denen im Allgemeinen umso höhere Kosten entstehen, je größer die Abweichung von einem optimalen Platz ist. Die drei folgenden sehr ähnlichen einfachen Probleminstanzen haben diese Eigenschaft nicht. Die Kostenmatrix der Instanzen wird nicht über absolute Abweichungen der Platzierung eines Elements von einem optimalen Platz berechnet. Es wird nur zwei unterschiedliche Kostenwerte in der Kostenmatrix geben. Jede der folgenden Probleminstanzen hat zwei ausgezeichnete optimale Lösungen. Neben diesen beiden ausgezeichneten Lösungen gibt es evtl. noch andere Lösungen mit minimalen Kosten. Die beiden ausgezeichneten optimalen Lösungen haben die Eigenschaft, dass keine der Entscheidungen der einen Lösung Teil der anderen Lösung ist, d.h. bei einer Beschreibung der optimalen Lösungen mit einer Menge von Tupeln der Form (Platz, Element), sind die beiden Mengen disjunkt. Die Kostenmatrix wird wieder so definiert, dass entlang der Hauptdiagonalen die Kosten $c(i, i) = 1$, $1 \leq i \leq n$ entstehen. Zur Diagonalen legen wir eine Parallele mit Abstand k , deren zugehörige Ent-

scheidungen dann auch jeweils die Kosten 1 verursachen. Hierfür setzen wir $c(i, j) = 1$ für alle Tupel (i, j) , für die $j - i \equiv k \pmod{n}$ für ein konstantes ganzzahliges $k \in [1, n - 1]$ gilt. Für alle anderen Elemente in der Kostenmatrix gilt $c(i, j) = 2$. Wir wählen wieder $n = 50$ und setzen für die 3 unterschiedlichen Instanzen $k = 3$, $k = 23$ und $k = 25$ und bezeichnen die Probleminstanzen entsprechend mit P_{P_3} , $P_{P_{23}}$ oder $P_{P_{25}}$ (Parallele mit Abstand $k \in \{3, 23, 25\}$).

Die lokale und die relative Pheromonauswertung liefert auf den Probleminstanzen P_{P_k} sehr gute Ergebnisse und findet fast immer die optimale Lösung (vgl. Tabelle 4.4 auf Seite 66). Die Ergebnisse bei Verwendung der Summenpheromonauswertung sind erwartungsgemäß schlecht. Mit der lokalen und der relativen Pheromonauswertung konnte auf der Probleminstanz $P_{P_{25}}$ immer die optimale Lösung gefunden werden, auf den Probleminstanzen P_{P_3} und $P_{P_{23}}$ jedoch nicht. Dies lässt sich wie folgt begründen. Die Probleminstanzen P_{P_3} und $P_{P_{23}}$ haben jeweils nur zwei optimale Lösungen: 1.) auf Platz $i \in [1, 50]$ wird Element i gewählt, 2.) auf Platz $i \in [1, 50]$ wird Element $i + k \pmod{50}$ gewählt, wobei k der Abstand der Parallelen zur Hauptdiagonalen ist. Für $n = 50$ hat die Instanz $P_{P_{25}}$ insgesamt 2^{25} optimale Lösungen: für jedes $i \in [1, 25]$ wählen wir auf Platz i und $i + 25$ entweder Elemente i und $i + 25$, oder Elemente $i + 25$ und i .

Eine weitere interessante Beobachtung ist, dass die lokale und die relative Pheromonauswertung die Elemente bei der Probleminstanz P_{P_3} mit viel größerer Wahrscheinlichkeit auf der Parallelen platzieren. Für diese Untersuchung ist in Tabelle 4.4 der Quotient D/P angegeben. D gibt die durchschnittliche Anzahl der Elemente einer Lösung an, die auf der Hauptdiagonalen gewählt wurden. P gibt die Anzahl der Elemente an, die entlang der Parallelen gewählt wurden. Bei den Probleminstanzen $P_{P_{23}}$ und $P_{P_{25}}$ tritt die ungleichmäßige Auswahl nicht auf ($D/P \approx 1.0$). Falls bei der Instanz P_{P_3} beispielsweise Element 4 auf Platz 1 gewählt wird, kann Element 4 nicht mehr auf Platz 4 gewählt werden. Daher ist es wahrscheinlicher, dass Element 7 auf Platz 4 gewählt wird. Wenn Elemente auf den ersten Plätzen entlang der Parallelen platziert werden, dann erhöht das also die Wahrscheinlichkeit, dass k Entscheidungen später eine Entscheidung auch entlang der Parallelen getroffen wird. Für kleine k hat dieser Effekt bei der Verwendung von Vorwärts-Ameisen eine starke Bedeutung. Für die lokale Pheromonauswertung ist das unterschiedliche Verhalten anhand von durchschnittlichen Pheromonmatrizen für die Probleminstanzen P_{P_3} und $P_{P_{23}}$ in Abbildung 4.6 abgebildet. Während bei der Probleminstanz P_{P_3} die Entscheidungen nahezu alle entlang der Parallelen gefällt werden, ist das bei der Instanz $P_{P_{23}}$ nicht der Fall. Wenn eine zufälli-

Tabelle 4.4: Ergebnisse für die Instanzen P_{P_k} mit $k \in \{3, 23, 25\}$; angegeben sind: die durchschnittliche Lösungsqualität der besten Lösung; die durchschnittliche Generation, in der diese Lösung gefunden wurde; das durchschnittliche Verhältnis der Anzahl von Elementen, die auf der Hauptdiagonalen gewählt wurden, zur Anzahl von Elementen, die auf der Parallelen gewählt wurden (D/P)

Auswertung	Probleminstanz P_{P_3}			
		bestes Ergebnis	Generation	(D/P)
lokale PA	Vorwärts	51.2	1770	0.03
	Zufall	52.0	3011	1.01
relative PA	Vorwärts	50.9	1653	0.06
	Zufall	52.0	3118	1.01
relative* PA	Vorwärts	50.7	1625	0.04
Summen-PA	Vorwärts	72.5	6193	4.13

Auswertung	Probleminstanz $P_{P_{23}}$			
		bestes Ergebnis	Generation	(D/P)
lokale PA	Vorwärts	53.3	1919	0.93
	Zufall	52.1	3012	0.93
relative PA	Vorwärts	52.2	2009	0.95
	Zufall	51.8	3094	1.11
relative* PA	Vorwärts	53.0	1831	0.96
Summen-PA	Vorwärts	72.4	6502	1.15

Auswertung	Probleminstanz $P_{P_{25}}$			
		bestes Ergebnis	Generation	(D/P)
lokale PA	Vorwärts	50.0	1538	0.85
	Zufall	50.0	1850	0.95
relative PA	Vorwärts	50.0	1500	0.93
	Zufall	50.0	1872	0.98
relative* PA	Vorwärts	50.0	1392	1.02
Summen-PA	Vorwärts	72.6	6504	1.01

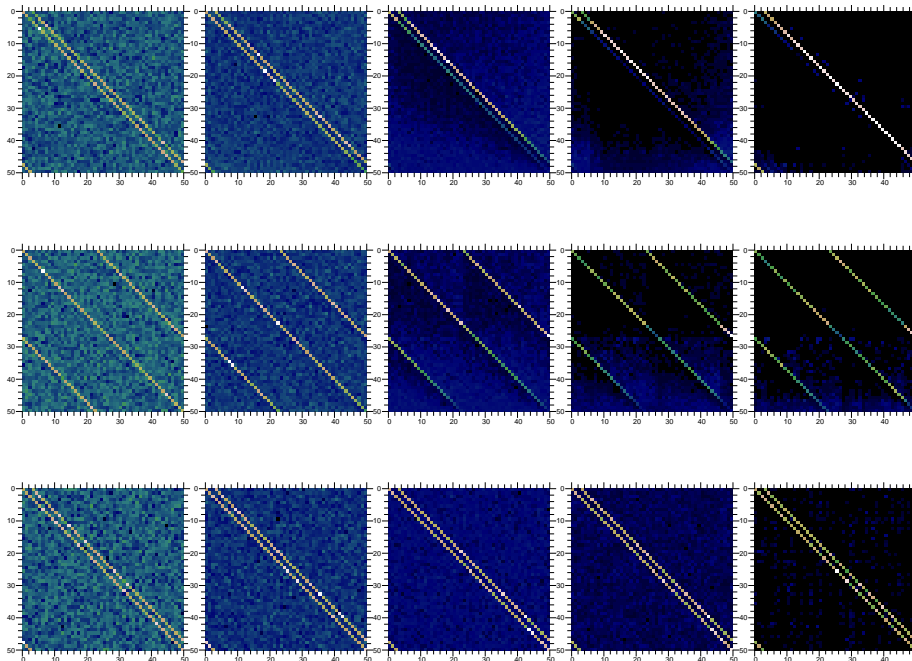


Abbildung 4.6: Durchschnittliche Pheromonmatrizen: Probleminstanz P_{P_3} , lokale Pheromonauswertung, Vorwärts-Ameisen (oben); Probleminstanz $P_{P_{2,3}}$, lokale Pheromonauswertung, Vorwärts-Ameisen (Mitte); Probleminstanz P_{P_3} , lokale Pheromonauswertung, zufällig gewählte Entscheidungsreihenfolge (unten); dargestellt sind jeweils die Pheromonmatrizen in den Generationen 100, 500, 1000, 1500 und 2000 (oben und Mitte) und 100, 500, 1000, 1500 und 3500 (unten)

ge Entscheidungsreihenfolge gewählt wird, ist auch auf P_{P_3} eine gleichmäßige Verteilung zu beobachten.

4.3 Ein NP-vollständiges Problem

Wir haben im vorherigen Abschnitt gesehen, dass die Verwendung einer zufällig gewählten Permutation für die Einplanungsreihenfolge zu einem deutlich besseren Optimierungsverhalten führen kann. Jedoch wurde schon darauf hingewiesen, dass die Verwendung einer Heuristik bei der Berechnung der Auswahlwahrscheinlichkeiten bei einer Zufallspermutation nicht oder nur eingeschränkt möglich ist. Wir werden bei den Untersuchungen der Methoden auf einem schwierigen Problem wieder ein Planungsproblem untersuchen, für

welches eine Heuristikinformation genutzt wird, die auf einer Prioritätsregel basiert. Generell existieren für viele Planungsprobleme sehr gute Prioritätsregeln, die — evtl. angepasst — wichtig für das Optimierungsverhalten von ACO-Algorithmen sind. Um von den Vorteilen der Vorwärts-Ameisen bezüglich der möglichen Verwendung einer Heuristik und zusätzlich von den Vorteilen einer zufälligen Entscheidungsreihenfolge zu profitieren, kombinieren wir in diesem Abschnitt beide Ansätze. Ein Teil der Ameisen wird durch eine Einplanung mit der Entscheidungsreihenfolge $(1, \dots, n)$ von einer Prioritätsregel profitieren können, ein anderer Teil wird Lösungen mit einer zufälligen Entscheidungsreihenfolge ohne jegliche auf Prioritätsregeln basierende Heuristikinformation erzeugen. Man beachte, dass durch die Verwendung dieser beiden unterschiedlichen Typen von Ameisen in einer Generation im Allgemeinen nicht der gewünschte Effekt eintritt. Oft wird es dann nämlich so sein, dass der eine Typ Ameise den anderen Typ innerhalb einer Generation dominiert. Die Aktualisierung wird dann nur durch die bessere Ameisenart durchgeführt, die schlechtere Ameisenart ist größtenteils überflüssig (wir gehen davon aus, dass nur die beste Ameise jeder Generation die Pheromonmatrix aktualisiert). Wir lösen dieses Problem, indem wir eine andere Art der Kombination wählen. Innerhalb einer Generation verwenden wir immer nur die gleiche Ameisenart, in unterschiedlichen Generationen verwenden wir unterschiedliche Einplanungsreihenfolgen für alle Ameisen der jeweiligen Generation. Eine Konkurrenz findet somit nur zwischen Ameisen der gleichen Art statt.

Als NP-vollständiges Testproblem wählen wir ein Ein-Maschinen-Planungsproblem, für welches die Summe aus gewichteten Verfrühungen und Verspätungen minimiert werden soll. Das Problem wird mit SMTWDP bezeichnet (single machine total weighted deviation problem). Formal haben wir n Jobs gegeben, wobei jeder Job j , $1 \leq j \leq n$ eine Laufzeit p_j , einen Fälligkeitstermin d_j und zwei Gewichte h_j und w_j hat. Gesucht ist ein unterbrechungsfreier Belegungsplan der Jobs, sodass $D = \sum_{j=1}^n (h_j \cdot \max\{0, d_j - C_j\} + w_j \cdot \max\{0, C_j - d_j\})$ minimiert wird. C_j ist hierbei der Fertigstellungszeitpunkt des Jobs j . Der Wert $h_j \cdot \max\{0, d_j - C_j\}$ ist die gewichtete Verfrühung eines Jobs und $w_j \cdot \max\{0, C_j - d_j\}$ ist dessen gewichtete Verspätung. Schon das Problem, bei dem nur gewichtete Verspätungen betrachtet werden, ist NP-vollständig im strengen Sinne (vgl. Abschnitt 3.1.3). Eine Übersicht von Arbeiten, bei denen Ein-Maschinen-Planungsprobleme mit Verfrühungen und Verspätungen betrachtet werden, ist in (Wu et al., 2000) gegeben.

4.3.1 Heuristik für SMTWDP

Bei der Berechnung der Heuristikwerte gehen wir zunächst von der Verwendung von Vorwärts-Ameisen aus. Die Werte η_{ij} werden in Anlehnung an die in (Ow und Morton, 1998) vorgestellte Heuristik berechnet. Falls noch Jobs einzuplanen sind, die ihren Fälligkeitstermin schon überschritten haben oder im nächsten Schritt überschreiten werden (ausgehend von dem bis zum aktuellen Platz erzeugten partiellen Plan), wählt eine Vorwärts-Ameise den nächsten Job immer aus der Menge dieser Jobs. Für jeden dieser Jobs gilt, dass die Kosten immer höher werden, je später er eingeplant wird. Aus der Menge dieser Jobs bevorzugen wir solche, die eine kurze Laufzeit und ein hohes Gewicht für die Verspätungskosten haben (vgl. Heuristik für SMTWTP in Abschnitt 3.1.3). \mathcal{T} sei wie üblich der Fertigstellungszeitpunkt des zuletzt eingeplanten Jobs (dies entspricht der Länge des partiellen Plans). Dann berechnen wir die Heuristikwerte für alle Jobs, für die $d_j \leq \mathcal{T} + p_j$ gilt, durch

$$\eta_{ij} = \frac{w_j}{p_j}$$

Somit werden Jobs mit kurzen Laufzeiten und hohem Gewicht der Verspätungskosten bevorzugt. Einige der Jobs, welche ihren Fälligkeitstermin nicht überschreiten, wenn sie als nächstes eingeplant werden, überschreiten den Fälligkeitstermin evtl. sehr bald danach. Solche Jobs werden wir auch bei der Auswahl des nächsten Jobs betrachten. Jedem Job j , der eine Laufzeit hat, die kleiner als die durchschnittliche Laufzeit \bar{p} aller noch einzuplanenden Jobs ist, und für den der Fälligkeitstermin den Zeitpunkt $\mathcal{T} + p_j$ maximal um $(\bar{p} - p_j) \cdot \frac{w_j}{w_j + h_j}$ überschreitet, wird ein positiver Heuristikwert zugeordnet. Man beachte, dass dieser Wert für die Jobs größer ist, deren Gewicht der Verspätungskosten im Vergleich zu dem Gewicht der Kosten der Verfrühungen relativ hoch ist. Die zugeordneten Heuristikwerte fallen linear mit zunehmendem Fälligkeitstermin von $\frac{w_j}{p_j}$ auf 0 im Intervall $[\mathcal{T} + p_j, \mathcal{T} + p_j + (\bar{p} - p_j) \cdot \frac{w_j}{w_j + h_j}]$. Formal berechnen wir also für jeden Job mit $d_j \in [\mathcal{T} + p_j, \mathcal{T} + p_j + \max\{0, \bar{p} - p_j\} \cdot \frac{w_j}{w_j + h_j}]$ den Heuristikwert

$$\eta_{ij} = \left(1 - \frac{d_j - \mathcal{T} - p_j}{\max\{0, \bar{p} - p_j\} \cdot \frac{w_j}{w_j + h_j}}\right) \cdot \frac{w_j}{p_j}$$

Die Berechnung der Heuristikwerte, falls mindestens ein Job noch nicht eingeplant ist, für den $d_j \leq \mathcal{T} + p_j + \max\{0, \bar{p} - p_j\} \cdot \frac{w_j}{w_j + h_j}$ gilt, ist in Abbildung 4.7 veranschaulicht.

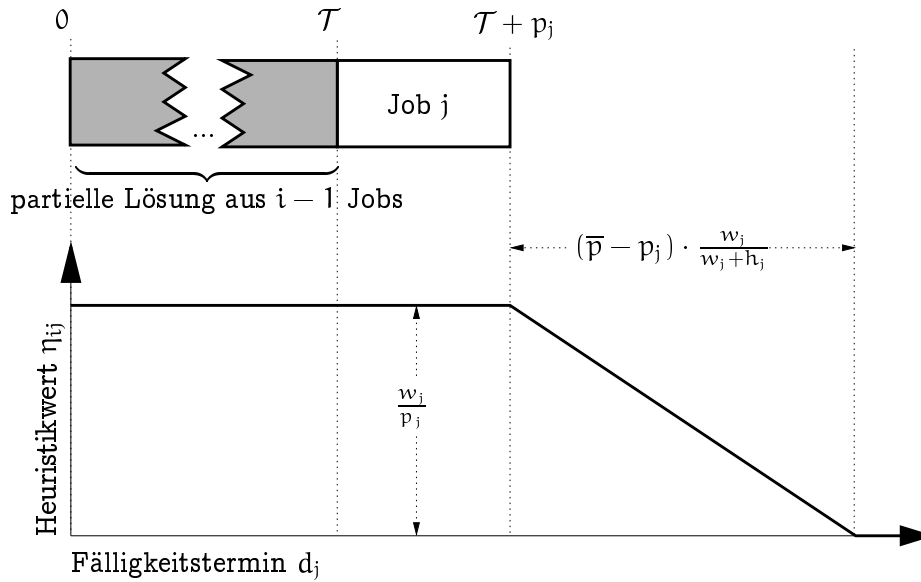


Abbildung 4.7: Heuristikwert η_{ij} des Jobs j mit $p_j < \bar{p}$ in Abhängigkeit von d_j bei Verwendung von Vorwärts-Ameisen, falls mindestens ein Job noch nicht eingeplant ist, dem entsprechend dieser Vorgehensweise ein Heuristikwert $\eta_{ij} > 0$ zugeordnet wird

Wenn alle noch einzuplanenden Jobs einen Fälligkeitstermin $d_j > T + p_j + (\bar{p} - p_j) \cdot \frac{w_j}{w_j + h_j}$ haben, dann bevorzugen wir Jobs mit langen Laufzeiten p_j und einem niedrigen Gewicht für die Verfrühungskosten w_j . Wir berechnen dann alle Heuristikwerte mit

$$\eta_{ij} = \frac{p_j}{h_j}$$

Rückwärts-Ameisen, die zunächst entscheiden welcher Job zum Zeitpunkt $\sum_{j=1}^n p_j$ endet (und somit auf dem letzten Platz in der Lösungspermutation gewählt wird), gehen prinzipiell symmetrisch zu dem eben erläuterten Verfahren bei Vorwärts-Ameisen vor. Zunächst wird ein Job aus der Menge aller Jobs gewählt, die ihren Fälligkeitstermin nicht überschritten haben (für die somit Verfrühungskosten anfallen), sofern es solche Jobs gibt. Von diesen bevorzugen wir die kürzeren und jene, die ein hohes Gewicht der Verfrühungskosten haben. Man beachte, dass wir auch bei der Verwendung von Rückwärts-Ameisen von einer Überschreitung des Fälligkeitstermins sprechen, wenn der Fertigstellungszeitpunkt des Jobs zeitlich nach dem Fälligkeitstermin liegt. Es sei T nun die Summe der Laufzeiten aller noch nicht eingeplanten Jobs. Formal

berechnen wir dann für alle Jobs, für die $d_j \geq \mathcal{T}$ gilt, die Heuristikwerte

$$\eta_{ij} = \frac{h_j}{p_j}$$

Analog zu Vorwärts-Ameisen betrachten wir auch Jobs, die einen Fälligkeitstermin knapp vor \mathcal{T} haben. Allen Jobs, die eine Laufzeit haben, die kleiner als der Durchschnitt aller noch einzuplanenden Jobs ist, und deren Fälligkeitstermin nicht vor $\mathcal{T} - (\bar{p} - p_j) \cdot \frac{h_j}{w_j + h_j}$ liegt, wird ein positiver Heuristikwert zugeordnet. Wir berechnen also formal für alle Jobs, deren Fälligkeitstermin d_j im Intervall $\left[\mathcal{T} - \max\{0, \bar{p} - p_j\} \cdot \frac{h_j}{w_j + h_j}, \mathcal{T} \right]$ liegt, den Heuristikwert

$$\eta_{ij} = \left(1 - \frac{\mathcal{T} - d_j}{\max\{0, \bar{p} - p_j\} \cdot \frac{h_j}{w_j + h_j}} \right) \cdot \frac{h_j}{p_j}$$

Falls keinem der noch einzuplanenden Jobs ein Heuristikwert wie eben erläutert zugeordnet wird, dann bevorzugen wir solche mit langen Laufzeiten p_j und einem niedrigen Gewicht für die Verspätungskosten w_j . Wir berechnen dann alle Heuristikwerte mit

$$\eta_{ij} = \frac{p_j}{w_j}$$

4.3.2 Probleminstanzen und Parameter

Die Testinstanzen für SMTWDP haben die Größe $n = 100$. Die ganzzahligen Laufzeiten der Jobs wurden zufällig gleichverteilt aus dem Intervall $[10, 100]$ gewählt, die reellwertigen Gewichte der Verfrühungskosten wurden gleichverteilt aus dem Intervall $[1 \cdot e, 2 \cdot e]$ gewählt, und die Verspätungskosten wurden gleichverteilt aus dem Intervall $[1 \cdot t, 2 \cdot t]$ gewählt. Mögliche Werte für (e, t) waren $(5, 1)$, $(3, 1)$, $(1, 1)$, $(1, 3)$ und $(1, 5)$. Die ganzzahligen Fälligkeitstermine wurden gleichverteilt aus dem Intervall gewählt, welches in Gleichung (3.6) auf Seite 36 definiert wurde. Dabei wurde $RDD = 0.4$ und $TF \in \{0.2, 0.4, 0.6, 0.8\}$ gewählt. Durch $RDD = 0.4$ liegen die Fälligkeitstermine in einem Bereich, der 40% des Intervalls $[0, \sum_{j=0}^{100} p_j]$ abdeckt. Für jede der 20 möglichen Kombinationen aus TF und (e, t) wurden 15 unterschiedliche Testinstanzen erzeugt.

Die Verdunstungsrate beträgt $\rho = 0.01$. Die beste Ameise der Iteration aktualisiert die Pheromonmatrix immer mit $\frac{1}{D}$, wobei D die Kosten des Plans sind. Die Anzahl der Ameisen ist immer $m = 20$, falls wir nicht explizit darauf hinweisen, dass wir eine andere Anzahl von Ameisen verwenden. Pheromon-

und Heuristikwerte werden bei der Berechnung der Auswahlwahrscheinlichkeiten gleich gewichtet, d.h. es wird $\alpha = 1$ und $\beta = 1$ gewählt. Jeder Lauf wurde nach 20000 Generationen gestoppt.

Aufgrund der unterschiedlichen Verwendung von Vorwärts- und Rückwärts-Ameisen, wählen wir folgende Bezeichnungen. V-A beschreibt einen ACO-Algorithmus, der nur Vorwärts-Ameisen verwendet, bei R-A werden nur Rückwärts-Ameisen verwendet, und Z-A bezeichnet einen Algorithmus, bei dem für jede Ameise immer eine zufällig gewählte Einplanungsreihenfolge verwendet wird. Wenn wir Vorwärts-Ameisen in Kombination mit der (ungewichteten) Summen-Pheromonauswertung verwenden, dann schreiben wir ΣV -A und entsprechend ΣR -A bei Rückwärts-Ameisen. Kombinationen der zufälligen Einplanungsreihenfolge mit Vorwärts- oder Rückwärts-Ameisen bezeichnen wir mit VZ-A, RZ-A, ΣVZ -A und ΣRZ -A.

4.3.3 Ergebnisse

Ein Vergleich der Algorithmen V-A, R-A, Z-A, ΣV -A und ΣR -A ist in Abbildung 4.8 dargestellt. Es ist interessant, dass der Algorithmus Z-A, der keine Heuristikinformation verwendet, im Vergleich mit den anderen Verfahren sehr gut abschneidet. Das gute Optimierungsverhalten bei Verwendung einer zufällig gewählten Entscheidungsreihenfolge, welches wir bei den einfachen Probleminstanzen ohne die Verwendung einer problemspezifischen Heuristik beobachtet haben, scheint sich also bei diesem Optimierungsproblem zu bestätigen. In der Hälfte aller Fälle hat Z-A die besten Ergebnisse erzielt. Z-A verhält sich besonders gut bei Probleminstanzen, bei denen die Fälligkeitstermine in der Mitte des gesamten Planungsintervalls liegen ($TF = 0.4$ und $TF = 0.6$). ΣV -A war die beste Variante bei Probleminstanzen mit späten Fälligkeitsterminen ($TF = 0.2$), und ΣR -A war die beste Variante bei Probleminstanzen mit frühen Fälligkeitsterminen ($TF = 0.8$). Für Probleme mit $TF \in \{0.4, 0.6\}$ beobachten wir, dass die Algorithmen V-A und ΣV -A bei Problemen, bei denen die Verfrühungskosten hoch sind, besser abschneiden. Analog gilt, dass R-A und ΣR -A bei Problemen, bei denen die Verspätungskosten hoch sind, bessere Ergebnisse erzielen. Die bevorzugte Einplanungsreihenfolge ist also deutlich von den Parametern der jeweiligen Probleminstanz abhängig.

Der Vergleich der Algorithmen, die Ameisen mit zwei unterschiedlichen Einplanungsreihenfolgen verwenden (VZ-A, RZ-A, ΣVZ -A und ΣRZ -A) zu den entsprechenden Verfahren ohne die zufällige Einplanungsreihenfolge (V-A, R-A, ΣV -A und ΣR -A), zeigt, dass mit den kombinierten Verfahren in allen

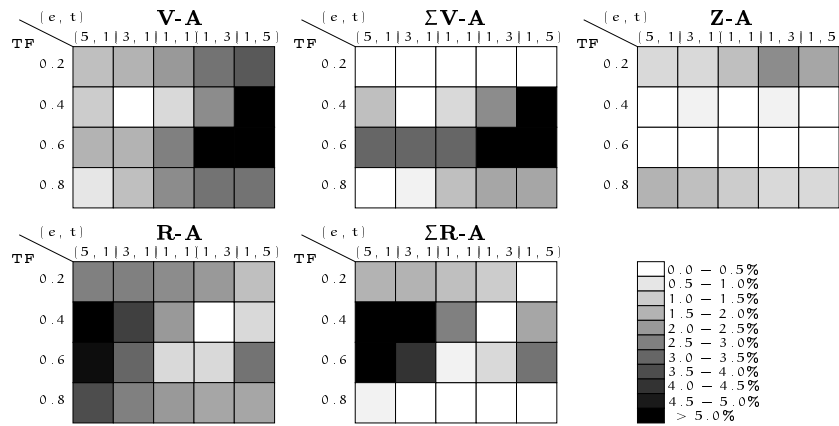


Abbildung 4.8: Ergebnisse für V-A, R-A, ΣV -A, ΣR -A und Z-A: Für jede Kombination TF und (e, t) ist die mittlere durchschnittliche Verschlechterung im Vergleich mit der jeweils besten der fünf Ergebnisse dargestellt. Ein weißes Quadrat bedeutet, dass der zugehörige Algorithmus für die entsprechende Kombination TF und (e, t) der beste der fünf verglichenen Algorithmen ist oder maximal 0.5% vom besten Algorithmus abweicht.

Fällen bessere Ergebnisse erzielt wurden. Ein Vergleich der vier Verfahren und dem Verfahren Z-A ist in Abbildung 4.9 gegeben.

Wenn die Fälligkeitstermine sehr früh oder sehr spät sind ($TF = 0.8$ oder $TF = 0.2$), dann liefern ΣVZ -A und ΣRZ -A die besten Ergebnisse. Es ist interessant, dass die Algorithmen, bei denen die Summen-Pheromonauswertung verwendet wird, auch von der Kombination mit einer zufällig gewählten Entscheidungsreihenfolge profitieren können (mit ΣV -A und ΣR -A wurden bessere Ergebnisse als mit Z-A erzielt). Bei den Probleminstanzen mit $TF = 0.4$ und $TF = 0.6$ ist bei den kombinierten Verfahren wieder deutlich zu erkennen, dass die zu bevorzugende Einplanungsreihenfolge stark von den Gewichten der Verfrühungs- und Verspätungskosten abhängt. Wie bei den Untersuchungen mit einfachen Permutationsproblemen ist vorzugsweise eine Einplanungsrichtung zu wählen, bei der die Entscheidungen, die hohe Kosten verhindern können, früh gefällt werden.

Von dem Verfahren mit einer zufällig gewählten Entscheidungsreihenfolge kann aufgrund einer jeweils unterschiedlichen Entscheidungsreihenfolge jeder Ameise erwartet werden, dass die Kosten der erzeugten Lösungen einer Generation eine größere Varianz haben, als die Kosten der Lösungen, die nur mit Vorwärts-Ameisen erzeugt wurden. Dies legt nahe, dass es von Vorteil ist,

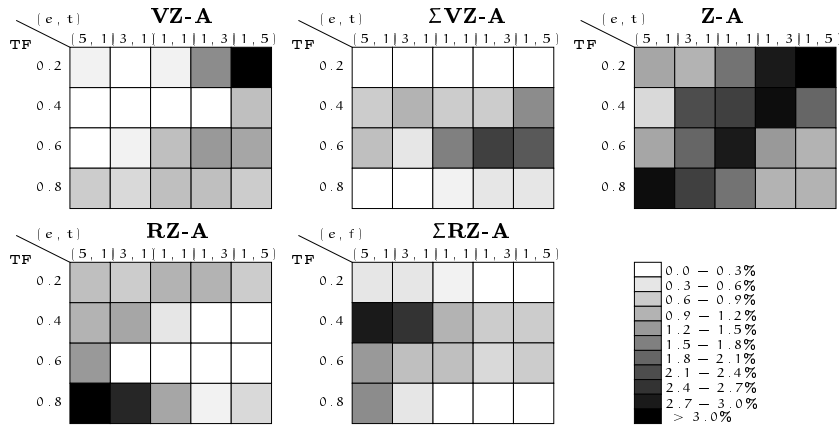


Abbildung 4.9: Ergebnisse für VZ-A, RZ-A, Σ VZ-A, Σ RZ-A und Z-A: Für jede Kombination TF und (e, t) ist die durchschnittliche Verschlechterung im Vergleich mit der jeweils besten der fünf Ergebnisse dargestellt. Man beachte, dass die Messungen der hier dargestellten Ergebnisse für Z-A identisch sind zu denen aus Abbildung 4.8. Die dunkleren Farben bei Z-A im Vergleich zur entsprechenden Grafik aus Abbildung 4.8 basieren auf der Tatsache, dass einer der anderen vier Algorithmen, die Ameisen mit verschiedenen Einplanungsreihenfolgen kombinieren, bessere Ergebnisse erzielt hat.

wenn die Konkurrenz der Ameisen in einer Generation von Ameisen mit einer zufällig gewählten Entscheidungsreihenfolge nicht zu klein ist. Der Einfluss der Anzahl der Ameisen pro Generation für V-A und Z-A ist in Abbildung 4.10 dargestellt. Wir haben $TF = 0.4$ und $(e, t) = (1, 1)$ gewählt. Mit V-A wurden für 15 oder weniger Ameisen pro Generation die besseren Ergebnisse erzielt. Wenn die Konkurrenz innerhalb einer Generation steigt (mehr als 15 Ameisen pro Generation), dann wurde jeweils mit Z-A das bessere Ergebnis erzielt.

In Tabelle 4.5 sind die durchschnittlichen Kosten des jeweils besten der neun getesteten Verfahren dargestellt. Es ist deutlich zu sehen, dass jeweils ein kombiniertes Verfahren am besten war.

4.4 Zusammenfassung

In diesem Kapitel haben wir für ACO-Algorithmen eine Methode eingeführt, die auf einer zufällig gewählten Entscheidungsreihenfolge beruht. Die Methode hat bei der Untersuchung der Verzerrung der Auswahlwahrscheinlichkeiten deutlich besser die in der Pheromonmatrix gespeicherte Information reflek-

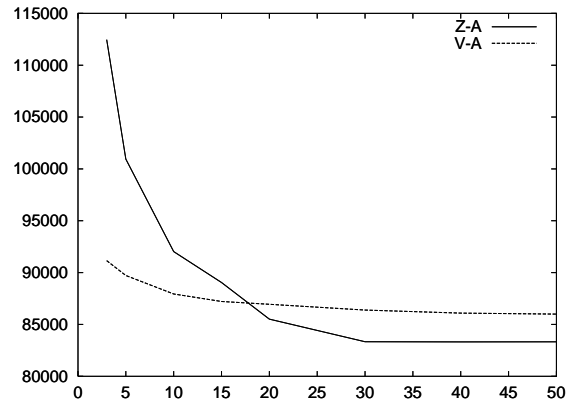


Abbildung 4.10: Durchschnittliche Summe der gewichteten Abweichungen für V-A und Z-A bei einer unterschiedlichen Anzahl von Ameisen; Parameter der Probleminstanzen: $TF = 0.4$ und $(e, t) = (1, 1)$

Tabelle 4.5: Durchschnittliche beste gewichtete Abweichung; dargestellt ist jeweils das Ergebnis des besten der 5 Verfahren V-A, R-A, ΣV -A, ΣR -A, Z-A, VZ-A, RZ-A, ΣVZ -A und ΣRZ -A für unterschiedliche Kombinationen (e, t) und TF

	(5,1)	(3,1)	(1,1)	(1,3)	(1,5)
TF=0.2	ΣRZ -A 739,122	ΣRZ -A 443,851	ΣRZ -A 148,453	ΣRZ -A 149,288	ΣRZ -A 149,834
TF=0.4	VZ-A 349,575	VZ-A 218,472	VZ-A 86,738	VZ-A 124,358	RZ-A 160,203
TF=0.6	VZ-A 152,589	RZ-A 121,241	RZ-A 87,874	RZ-A 226,185	RZ-A 363,407
TF=0.8	ΣVZ -A 155,756	ΣVZ -A 155,734	ΣRZ -A 154,640	ΣRZ -A 462,516	ΣRZ -A 770,590

tiert. Die in Kapitel 3 eingeführten globalen Pheromonauswertungsmethoden wurden mit Vorwärts-Ameisen, Rückwärts-Ameisen und Ameisen, die eine zufällig gewählte Entscheidungsreihenfolge wählen, kombiniert. Die Verfahren wurden zunächst auf einfachen Permutationsproblemen untersucht. Dabei wurden insbesondere bei der zufälligen Wahl einer Entscheidungsreihenfolge deutlich bessere Ergebnisse erzielt. Für Probleme, bei denen die Verwendung der Heuristikinformation bei der Berechnung der Auswahlwahrscheinlichkei-

ten wichtig ist, wurde eine Kombination von Verfahren vorgestellt, bei der sowohl die Vorteile einer zufälligen Entscheidungsreihenfolge als auch die Vorteile einer Heuristik erhalten bleiben. Dies wurde auf einem NP-vollständigen Ein-Maschinen-Planungsproblem empirisch nachgewiesen.

Kapitel 5

Projektplanung unter beschränkten Ressourcen

Viele in der Praxis vorkommende Planungsprobleme können als ressourcenbeschränkte Projektplanungsprobleme formuliert werden. In den letzten Jahren wurden vielfältige Untersuchungen zu diesem NP-vollständigen Problem publiziert. Eine Übersicht ist z.B. in (Brucker et al., 1999) und (Weglarz, 1999) gegeben. Neben den wichtigsten Referenzen und den Definitionen zahlreicher Problemvarianten, findet sich dort unter anderem auch ein Überblick über deterministische und zufallsgesteuerte Optimierungsverfahren für das Problem.

In diesem Kapitel werden wir nach einer formalen Definition des Optimierungsproblems einen Ameisenalgorithmus vorstellen, der bei einem Vergleich mit mehr als einem Dutzend anderer Verfahren, die zum Großteil auf dem aktuellen Stand der Forschung sind, sehr gute Ergebnisse liefert. Um die Güte des ACO-Algorithmus zu erreichen, setzen wir einerseits die in den vorherigen Kapiteln vorgestellten Methoden ein (globale Pheromonauswertung, Vorwärts- und Rückwärts-Ameisen), und werden andererseits noch zusätzliche Verbesserungen vorstellen. Im Gegensatz zu den Methoden, welche den Entscheidungsprozess bei der Konstruktion einer einzelnen Lösung verbessern, teilen wir hier den gesamten Suchprozess systematisch in Phasen auf, und führen dann phasenabhängig algorithmische Anpassungen durch.

In Abschnitt 5.1 führen wir einige notwendige Begriffe ein und definieren das Problem der Projektplanung unter beschränkten Ressourcen. Für die Erzeugung einer Lösungspermutation werden in Abschnitt 5.2 unterschiedliche Prioritätsregelverfahren erläutert. In Abschnitt 5.3 wird der ACO-Algorithmus für das Problem vorgestellt. Wir werden für den Vergleich mit anderen Methoden die größten Testprobleme einer bekannten Standardbibliothek (PSPLIB)

verwenden, die zusammen mit den Parametern des ACO-Algorithmus in Abschnitt 5.4 festgelegt werden. Die Ergebnisse der Tests werden in Abschnitt 5.5 vorgestellt. Teile der in diesem Kapitel vorgestellten Resultate sind unter anderem in (Merkle et al., 2000a,b,c; Merkle und Middendorf, 2001e) und (Merkle et al., 2002) erschienen.

5.1 Definitionen

Beim ressourcenbeschränkten Projektplanungsproblem (RCPSP) sollen Aktivitäten eines Projekts so angeordnet werden, dass die Projektdauer minimiert wird. Dabei müssen zum einen Vorrangbeziehungen zwischen den Aktivitäten eingehalten werden (d.h. eine Aktivität darf erst gestartet werden, wenn andere Aktivitäten beendet wurden), und zum anderen darf die Ressourcenanforderung pro Zeiteinheit aller sich zu einem Zeitpunkt in Ausführung befindlichen Aktivitäten einen gegebenen Maximalwert pro Ressource nicht überschreiten.

Formal sei $\mathcal{J} = \{0, \dots, n+1\}$ die Menge aller *Aktivitäten* eines Projekts. Es sei eine *Vorrangbeziehung* der Aktivitäten gegeben. Eine *Aktivitätenliste* ist eine Permutation aller Aktivitäten, sodass jede Aktivität in der Aktivitätenliste vor allen Aktivitäten steht, die gemäß der Vorrangbeziehung Nachfolger dieser Aktivität sind. \mathcal{Q} ist die Menge aller *Ressourcen*. $R_i > 0$ ist die *Ressourcenkapazität* der Ressource $i \in \mathcal{Q}$. Jede Aktivität $j \in \mathcal{J}$ hat eine *Laufzeit* p_j und *Ressourcenbedarfe* $r_{j,1}, \dots, r_{j,q}$, wobei $r_{j,i}$ der Bedarf der Ressource i pro Zeiteinheit ist, wenn sich Aktivität j in Ausführung befindet.

\mathcal{P}_j ist die Menge der direkten Vorgänger der Aktivität j . \mathcal{S}_j ist die Menge der direkten Nachfolger der Aktivität j . \mathcal{P}_j^* ist die Menge aller Vorgänger der Aktivität j . \mathcal{S}_j^* ist die Menge aller Nachfolger der Aktivität j . Aktivität 0 ist die einzige Aktivität, die keinen Vorgänger besitzt, und Aktivität $n+1$ ist die einzige Aktivität, die keinen Nachfolger hat. Die Anfangs- und Endaktivität haben keinen Ressourcenbedarf und jeweils die Laufzeit 0.

Ein *Projektplan* eines Projekts ist ein Vektor $(s_0, s_1, \dots, s_{n+1})$, wobei s_j der *Anfangszeitpunkt* der Aktivität $j \in \mathcal{J}$ ist. Falls s_i der Anfangszeitpunkt der Aktivität i ist, dann ist $f_i = s_i + p_i$ deren *Endzeitpunkt*. Der *Anfangszeitpunkt eines Projektplans* ist der minimale Anfangszeitpunkt $\min\{s_j \mid j \in \mathcal{J}\}$ aller Aktivitäten, der *Endzeitpunkt eines Projektplans* ist der maximale Endzeitpunkt $\max\{f_j \mid j \in \mathcal{J}\}$ aller Aktivitäten. Die *Dauer eines Projektplans* ist die Differenz des Endzeitpunktes und des Anfangszeitpunktes des Projektplans. Man beachte, dass s_0 der Anfangszeitpunkt eines Projektplans und f_{n+1} der Endzeitpunkt eines Projektplans ist.

Ein Projektplan ist *zulässig*, falls er folgende Bedingungen erfüllt.

1. Eine Aktivität $j \in \mathcal{J}$ kann erst dann gestartet werden, wenn alle Vorgängeraktivitäten beendet wurden, d.h. für jede Aktivität $j \in \mathcal{J}$ gilt

$$\forall i \in \mathcal{P}_j : s_j \geq f_i \quad (5.1)$$

2. Die Ressourcenbeschränkungen sind erfüllt, d.h. für jede Ressource i und für jeden Zeitpunkt t darf die Summe der Ressourcenbedarfe aller sich zum Zeitpunkt t in Ausführung befindlichen Aktivitäten die Ressourcenkapazität R_i nicht überschreiten. Somit muss für jeden Zeitpunkt t gelten

$$\forall i \in \mathcal{Q} : \sum_{j \in \mathcal{J}, s_j \leq t < f_j} r_{j,i} \leq R_i \quad (5.2)$$

Beim ressourcenbeschränkten Projektplanungsproblem soll für ein gegebenes Projekt mit Ressourcenkapazitäten, Vorrangbeziehungen der Aktivitäten, Laufzeiten der Aktivitäten und Ressourcenbedarfe der Aktivitäten ein zulässiger Projektplan mit minimaler Dauer gefunden werden. In Abbildung 5.1 ist ein Beispiel für eine RCPSP Instanz und ein zugehöriger zulässiger Projektplan dargestellt. Die Vorrangbeziehungen sind dabei durch einen *Präzedenzgraphen* definiert.

Der *späteste Anfangszeitpunkt* (LS_j) und der *späteste Endzeitpunkt* (LF_j) jeder Aktivität j kann ausgehend von einer oberen Schranke des Endzeitpunktes T des Projektplans durch eine Rückwärtsrekursion berechnet werden (Elmaghraby, 1977). Ausgehend von $LS_{n+1} = LF_{n+1} = T$ setzen wir sukzessive $LF_j = \min\{LS_i \mid i \in \mathcal{S}_j\}$ und $LS_j = LF_j - p_j$ für $j = n, \dots, 0$. Analog lässt sich der *früheste Anfangszeitpunkt* (ES_j) und der *früheste Endzeitpunkt* (EF_j) berechnen. Ausgehend von $ES_0 = EF_0 = 0$ setzen wir sukzessive $ES_j = \max\{EF_i \mid i \in \mathcal{P}_j\}$ und $EF_j = ES_j + p_j$ für alle $j = 1, \dots, n+1$. Man beachte, dass der früheste Anfangszeitpunkt ES_{n+1} der Aktivität $n+1$ eine triviale untere Schranke für die minimale Dauer eines zulässigen Projektplans ist.

Der Übersicht wegen fassen wir die Bezeichner für RCPSP in Tabelle 5.1 noch einmal zusammen.

5.2 Prioritätsregelverfahren

Bei einfachen Ein-Maschinen-Planungsproblemen, wie wir sie in den vorherigen Kapiteln untersucht haben, konnte man auf offensichtliche Weise aus

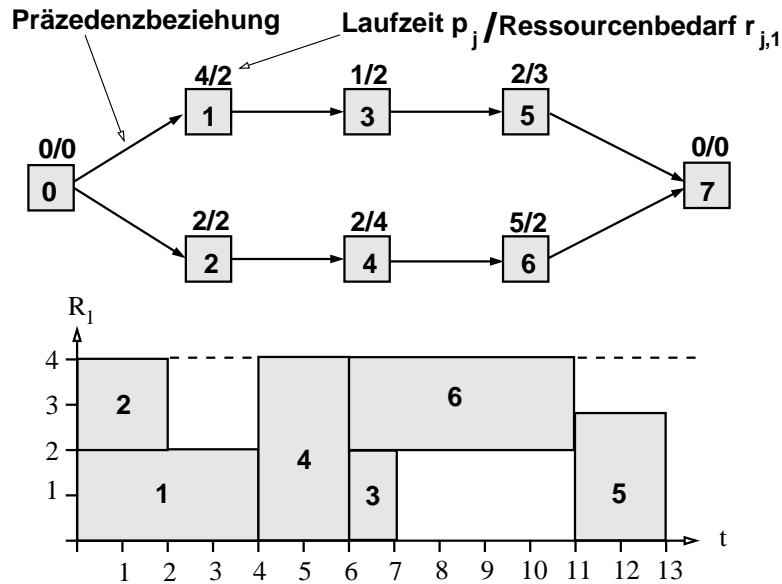


Abbildung 5.1: Präzedenzgraph einer Instanz des RCPSP mit Aktivitäten $0, \dots, 7$, einer Ressource mit Kapazität 4 und einem zulässigen Projektplan

der Lösungspermutation eine Laufzeit des Belegungsplans berechnen. Beim RCPSP kann man bei gegebener Lösungspermutation den Projektplan mit unterschiedlichen Verfahren (schedule generation schemes, SGS) erzeugen. Wir werden zwei unterschiedliche Verfahren zur Erzeugung eines Plans verwenden. Sowohl das serielle (SSGS) als auch das parallele (PSGS) Prioritätsregelverfahren sind Standardverfahren für das RCPSP (vgl. Kolisch und Hartmann, 1999).

SSGS startet mit einem partiellen Projektplan, der nur aus der Startaktivität 0 mit Start- und Endzeitpunkt 0 besteht. Es wird ein kompletter Plan in n Schritten erzeugt, wobei dem partiellen Plan in jedem Schritt genau eine Aktivität hinzugefügt wird. In jedem Schritt g wird eine Aktivität j aus der Menge der wählbaren Aktivitäten $\mathcal{E}(g)$ eingeplant. Die Menge der zum Schritt g wählbaren Aktivitäten enthält genau jene Aktivitäten, die noch nicht eingeplant sind, aber für die gilt, dass alle Aktivitäten, die im Präzedenzgraphen direkte Vorgänger sind, schon eingeplant sind. Um aus einer gegebenen Permutation von Aktivitäten einen Projektplan zu erzeugen, geht SSGS so vor, dass sukzessive jeweils die vorderste und bezüglich der Vorrangbeziehungen wählbare Aktivität $j \in \mathcal{E}(g)$ der Permutation eingeplant wird. Die Anfangszeitpunkte der Aktivitäten werden bei SSGS wie folgt berechnet. Für jede

\mathcal{J}	Menge der Aktivitäten
\mathcal{Q}	Menge der Ressourcen
R_i	Kapazität der Ressource i
$r_{j,i}$	Bedarf der Aktivität j an der Ressource i pro Zeiteinheit
p_j	Dauer der Aktivität j
\mathcal{P}_j	Menge der direkten Vorgänger der Aktivität j
\mathcal{S}_j	Menge der direkten Nachfolger der Aktivität j
\mathcal{P}_j^*	Menge aller Vorgänger der Aktivität j
\mathcal{S}_j^*	Menge aller Nachfolger der Aktivität j
s_j	Anfangszeitpunkt der Aktivität j
f_j	Endzeitpunkt der Aktivität j
F_j	maximaler Endzeitpunkt aller eingeplanten Aktivitäten $i \in \mathcal{P}_j$
LS_j	spätest möglicher Anfangszeitpunkt der Aktivität j
ES_j	frühest möglicher Anfangszeitpunkt der Aktivität j
LF_j	spätest möglicher Endzeitpunkt der Aktivität j
EF_j	frühest möglicher Endzeitpunkt der Aktivität j

Tabelle 5.1: Bezeichner für RCPSP

wählbare Aktivität $j \in \mathcal{E}(g)$ sei F_j der maximale Endzeitpunkt aller direkten Vorgänger im Präzedenzgraphen. Der Anfangszeitpunkt der Aktivität j ist der früheste Zeitpunkt im Intervall $[F_j, LS_j]$, sodass alle Ressourcenbeschränkungen erfüllt sind (siehe Algorithmus 5.3).

Algorithmus 5.3 Serielles Prioritätsregelverfahren

$\mathcal{E}(g)$: alle Aktivitäten, die entsprechend der Vorrangbeziehung (5.1) in Stufe g zulässig eingeplant werden können

- 1: **for** $g = 0$ to $n + 1$ **do**
 - 2: Berechne die Menge aller wählbaren Aktivitäten $\mathcal{E}(g)$
 - 3: Wähle ein $j \in \mathcal{E}(g)$
 - 4: Plane Aktivität j zum frühest möglichen Zeitpunkt $t \in [F_j, LS_j]$ unter Berücksichtigung der Vorrangbeziehung (5.1) und der Beschränkung aller Ressourcenkapazitäten (5.2)
 - 5: **end for**
-

Bei PSGS wird nicht über Aktivitäten, sondern über mögliche Anfangszeitpunkte der Aktivitäten iteriert (siehe Algorithmus 5.4). Auch PSGS startet

mit einem partiellen Projektplan, der nur aus der Aktivität 0 mit Anfangs- und Endzeitpunkt 0 besteht. In jedem Schritt g wird zunächst der Anfangszeitpunkt t_g der im nächsten Schritt einzuplanenden Aktivitäten berechnet, bevor diese eingeplant werden. Zunächst wird die Menge aller wählbaren Aktivitäten $\mathcal{E}(t_g)$ berechnet, welche zum Zeitpunkt t_g unter Berücksichtigung der Vorrangbeziehung und der Ressourcenkapazitäten eingeplant werden können. Der folgende Schritt wird dann solange iteriert, bis die Menge $\mathcal{E}(t_g)$ leer ist. Aus der Menge $\mathcal{E}(t_g)$ wird eine Aktivität j ausgewählt und zum Zeitpunkt t_g eingeplant. Danach wird $\mathcal{E}(t_g)$ wieder neu berechnet. Es können also während eines Schrittes auch keine oder mehrere Aktivitäten eingeplant werden. Am Ende der g -ten Iteration wird ein neuer Anfangszeitpunkt t_{g+1} der nächsten Iteration berechnet. Um aus einer gegebenen Permutation der Aktivitäten einen Projektplan zu erzeugen, geht PSGS so vor, dass jeweils die vorderste wählbare Aktivität $j \in \mathcal{E}(t_g)$ der Permutation eingeplant wird, die bezüglich Vorrangbeziehung und Ressourcenkapazitäten zulässig eingeplant werden kann.

Die Menge der Pläne, die durch SSGS erzeugt werden können, enthält auf jeden Fall die optimale Lösung der betrachteten Probleminstanz. Das bedeutet, dass eine Permutation von Aktivitäten existiert, welche mit SSGS zu einer minimalen Dauer des Projektplans führt (vgl. Kolisch und Hartmann, 1999). PSGS erzeugt immer so genannte unverzögerte Pläne. Das ist eine Teilmenge der Pläne, die von SSGS erzeugt werden (vgl. Sprecher et al., 1995). Die Menge aller unverzögerten Pläne enthält die optimale Lösung jedoch nicht notwendigerweise.

5.3 Ameisenalgorithmus für RCPSP

Um mit Hilfe eines Ameisenalgorithmus eine Instanz des RCPSP qualitativ gut zu lösen, konstruieren die Ameisen eine Permutation (Aktivitätenliste), die einen Plan mit möglichst kurzer Dauer mit Hilfe des seriellen oder des parallelen Prioritätsregelverfahrens findet. Dabei wird für einen Lauf immer für alle Ameisen das gleiche Prioritätsregelverfahren verwendet. Wie üblich erzeugen in jeder Generation m Ameisen eine Lösung. Jede Ameise wählt die einzuplanenden Aktivitäten in der Reihenfolge, in der sie bei dem seriellen oder dem parallelen Verfahren zur Erzeugung eines Plans verwendet wird. Zur Berechnung der Auswahlwahrscheinlichkeiten werden sowohl Pheromoninformation als auch Heuristikinformation verwendet. Die verwendeten Heuristiken werden im nächsten Abschnitt erläutert.

Algorithmus 5.4 Paralleles Prioritätsregelverfahren

$\mathcal{E}(t_g)$: alle Aktivitäten, die entsprechend der Vorrangbeziehung (5.1) und der Beschränkung aller Ressourcenkapazitäten (5.2) zum Zeitpunkt t_g zulässig eingeplant werden können

\mathcal{C} : Menge aller schon eingeplanten Aktivitäten

\mathcal{A}_g : $\mathcal{A}_g = \{j \in \mathcal{C} | s_j \leq t_g < f_j\}$ (aktive Menge: alle sich zum Zeitpunkt t_g in Ausführung befindlichen Aktivitäten)

- 1: $g := 0, t_g := 0, \mathcal{C} := \{0\}$
- 2: **while** $|\mathcal{J} - \mathcal{C}| > 0$ **do**
- 3: Berechne die Menge aller wählbaren Aktivitäten $\mathcal{E}(t_g)$
- 4: **while** $\mathcal{E}(t_g) \neq \emptyset$ **do**
- 5: Wähle ein $j \in \mathcal{E}(t_g)$
- 6: Plane Aktivität j mit Anfangszeitpunkt t_g ein
- 7: $\mathcal{C} := \mathcal{C} \cup \{j\}$
- 8: Aktualisiere $\mathcal{E}(t_g)$
- 9: **end while**
- 10: Berechne die zum Zeitpunkt t_g aktive Menge \mathcal{A}_g
- 11: $g := g + 1$
- 12: Berechne den minimalen Endzeitpunkt t_g aller Aktivitäten der Menge \mathcal{A}_{g-1}
- 13: **end while**

Die bei der Lösungskonstruktion zur Wahl stehenden Aktivitäten werden entsprechend der Wahrscheinlichkeitsverteilung über die Menge der wählbaren Aktivitäten \mathcal{E} berechnet. Zur Bestimmung der Wahrscheinlichkeiten wird entweder die lokale Pheromonauswertung oder die gewichtete Summen-Pheromonauswertung (vgl. Abschnitt 3.1.1) verwendet. Der Übersicht halber geben wir die Formeln zur Berechnung der Auswahlwahrscheinlichkeiten hier noch einmal an. Bei der lokalen Pheromonauswertung ergeben sich somit für $j \in \mathcal{E}$ die Wahrscheinlichkeiten wie folgt.

$$p_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{h \in \mathcal{E}} (\tau_{ih})^\alpha (\eta_{ih})^\beta} \quad (5.3)$$

Bei der gewichteten Summen-Pheromonauswertung ergeben sich folgende Wahrscheinlichkeiten

$$p'_{ij} = \frac{\left(\sum_{k=1}^i (\gamma^{i-k} \cdot \tau_{kj})\right)^\alpha \cdot (\eta_{ij})^\beta}{\sum_{h \in \mathcal{E}} \left(\sum_{k=1}^i (\gamma^{i-k} \cdot \tau_{kh})\right)^\alpha \cdot (\eta_{ih})^\beta} \quad (5.4)$$

wobei der Parameter $\gamma > 0$ den Einfluss der Pheromonwerte festlegt, die vorhergehenden Entscheidungen entsprechen.

Die beste bisher gefundene Lösung und die beste Lösung in der aktuellen Iteration werden verwendet, um die Pheromonmatrixaktualisierung durchzuführen. Wie üblich wird zuvor Pheromon entsprechend der Formel

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij}$$

verdunstet. Danach wird für jede Aktivität $j \in \mathcal{J}$ Pheromon auf das Pheromonmatrixelement τ_{ij} addiert, wobei i der Platz der Aktivität j in der Lösungspermutation der bisher besten Lösungspermutation ist. Wie bei dem in Abschnitt 3.1.3 vorgestellten Ein-Maschinen-Planungsproblem ist der Aktualisierungswert einer Ameise jeweils $\frac{\rho}{2T^*}$, wobei T^* die Projektdauer des bisher besten Plans ist. Die Pheromonmatrixaktualisierung entspricht somit der Formel

$$\tau_{ij} = \tau_{ij} + \rho \cdot \frac{1}{2T^*}$$

Die gleiche Vorgehensweise wird für die beste Lösung der aktuellen Iteration verfolgt. Dafür wird für jede Aktivität $j \in \mathcal{J}$ Pheromon auf das Pheromonmatrixelement τ_{ij} addiert, wobei i der Platz der Aktivität j in der Lösungspermutation der besten Lösung der aktuellen Iteration ist.

Der Ameisenalgorithmus wird abgebrochen, falls sich die durchschnittliche Lösungsqualität der Ameisen einer Iteration über mehrere Iterationen nicht ändert, oder falls eine vorgegebene Anzahl von Iterationen durchgeführt wurde.

5.3.1 Prioritätsregeln

Als mögliche Heuristiken des ACO-Algorithmus werden angepasste Prioritätsregeln für RCPSP verwendet. Eine Übersicht über Prioritätsregeln für RCPSP ist z.B. in (Klein, 2000) gegeben. Die Anpassungen sind wie bei der Untersuchung der Ein-Maschinen-Planungsprobleme notwendig, da zur Berechnung der Auswahlwahrscheinlichkeiten die absoluten Werte der Heuristik verwendet werden, und nicht nur die Ränge entsprechend der Sortierung, die sich

durch die Prioritätsregel-Heuristik ergibt (vgl. Abschnitt 3.1.3 und (Merkle und Middendorf, 2000)).

Die Latest Finish Time (LFT) Heuristik bevorzugt Aktivitäten mit großen Endzeitpunkten. Die relativen Unterschiede der Heuristikwerte für Aktivitäten, die erst spät gewählt werden können, sind üblicherweise klein. Daher verwenden wir die absoluten Differenzen zum Maximum der spätesten Endzeitpunkte aller wählbaren Aktivitäten als Heuristikwerte. Für die *normierte LFT Heuristik* (nLFT) werden somit die Werte η_{ij} entsprechend der folgenden Formel berechnet.

$$\eta_{ij} = \max_{k \in \mathcal{E}} LF_k - LF_j + 1$$

Die normierte Variante der Latest Start Time (LST) Heuristik wird mit *normierte LST Heuristik* (nLST) bezeichnet und wird identisch zur nLFT Heuristik definiert, nur dass anstatt der spätesten Endzeitpunkte LF die spätesten Anfangszeitpunkte LS verwendet werden. Formal definieren wir die Heuristikwerte somit wie folgt.

$$\eta_{ij} = \max_{k \in \mathcal{E}} LS_k - LS_j + 1$$

Die normierte Variante der Minimum Slack Time (MSL) Heuristik bezeichnen wir als *normierte MSL Heuristik* (nMSL) und berechnen die Heuristikwerte mit

$$\eta_{ij} = \max_{k \in \mathcal{E}} (LS_k - ES_k) - (LS_j - ES_j) + 1$$

Die Most Total Successors (MTS) Heuristik bevorzugt Aktivitäten mit einer großen Anzahl von Elementen in der Menge aller Nachfolger der jeweiligen Aktivität. Für die *normierte Most Total Successors* (nMTS) definieren wir

$$\eta_{ij} = |\mathcal{S}_j^*| - \min_{k \in \mathcal{E}} |\mathcal{S}_k^*| + 1$$

Die Greatest Rank Positional Weight All (GRPWA) Heuristik bevorzugt Aktivitäten, bei denen die Summe der Laufzeiten aller noch nachfolgenden Aktivitäten plus der Laufzeit der jeweiligen Aktivität möglichst groß ist. Für die *normierte Greatest Rank Positional Weight All* (nGRPWA) Heuristik definieren wir

$$\eta_{ij} = p_j + \sum_{i \in \mathcal{S}_j^*} p_i - \min_{k \in \mathcal{E}} \left(p_k + \sum_{i \in \mathcal{S}_k^*} p_i \right) + 1$$

Die Weighted Resource Utilization and Precedence (WRUP) Heuristik berechnet eine Prioritätsliste mit Hilfe des Anteils der genutzten Ressourcenkapazität und der Anzahl der direkten Nachfolger einer Aktivität. Für die *normierte Weighted Resource Utilization and Precedence* (nWRUP) Heuristik definieren wir

$$\eta_{ij} = \omega |\mathcal{S}_j| + (1 - \omega) \sum_{l \in Q} r_{jl}/R_l - \min_{k \in \mathcal{E}} \left(\omega |\mathcal{S}_k| + (1 - \omega) \sum_{l \in Q} r_{kl}/R_l \right) + 1$$

wobei $\omega \in [0, 1]$ ein frei wählbarer Parameter der Heuristik ist.

5.3.2 Zusätzliche Erweiterungen

Um mit dem Ameisenalgorithmus für RCPSP im Vergleich mit anderen Heuristiken sehr gute Ergebnisse zu erzielen, werden Methoden aus den vorherigen Kapiteln zur Verbesserung des Konstruktionsprozesses einer Lösung verwendet. Des Weiteren führen wir in diesem Abschnitt Erweiterungen ein, die den gesamten Suchprozess systematisch in Phasen einteilen.

Kombination der lokalen und der Summen-Pheromonauswertung

Die Summen-Pheromonauswertung wurde in Kapitel 3 eingeführt und anhand einfacher und NP-vollständiger Probleme getestet. Insbesondere hat sich gezeigt, dass die Summen-Pheromonauswertung dann erfolgreich angewendet werden kann, wenn eine kleine Veränderung der Lösungspermutation hinsichtlich der Plätze der zu wählenden Elemente nur einen geringen Qualitätsunterschied der erzeugten Lösungen zur Folge hat. Beim Problem der Projektplanung unter beschränkten Ressourcen ist diese Eigenschaft nur eingeschränkt gegeben. Aufgrund der Vorrangbeziehung dürfen manche Aktivitäten nicht zu spät geplant werden. Es ist jedoch auch wichtig, dass Gruppen von Aktivitäten so geplant werden, dass deren kumulierter Ressourcenbedarf die Ressourcenkapazitäten möglichst gut nutzt. Daher sind für manche Aktivitäten evtl. mehrere Plätze in der Lösungspermutation von Vorteil, während andere Plätze, die dazwischen liegen, von Nachteil sind. Eine Möglichkeit, das gewünschte Verhalten zu erreichen, ist die Verwendung der in Abschnitt 3.1.1 vorgestellten Kombination aus lokaler und Summen-Pheromonauswertung. Wir wiederholen hier die Vorgehensweise kurz. Mit dem Parameter c , $c \in [0, 1]$, steuern wir

den relativen Einfluss der lokalen und der Summen-Pheromonauswertung. Die Wahrscheinlichkeit, dass eine Ameise eine Aktivität j auf Platz i wählt, wird nach Gleichung (5.3) auf Seite 83 berechnet. Die Werte τ_{ij} werden durch die Werte τ_{ij}^* ersetzt, die wie folgt berechnet werden (vgl. auch Gleichung (3.4) auf Seite 28).

$$\tau_{ij}^* := c \cdot x_i \cdot \tau_{ij} + (1 - c) \cdot y_i \cdot \sum_{k=1}^i \gamma^{i-k} \tau_{kj}$$

Dabei sind $x_i := \sum_{h \in \mathcal{E}} \sum_{k=1}^i \gamma^{i-k} \tau_{kh}$ und $y_i := \sum_{h \in \mathcal{E}} \tau_{ih}$ Faktoren, um den relativen Einfluss der lokalen Pheromonauswertung und der Summen-Pheromonauswertung zu korrigieren.

Verwerfen der Elitelösung

Meist hat die Verwendung einer Elitestrategie den Vorteil, dass der Suchraum in der Nähe der bisher besten gefundenen Lösung verstärkt nach guten Lösungen abgesucht wird. Eine Elitelösung, die über mehrere Iterationen hinweg stabil war, hat einen starken Einfluss auf die Pheromonwerte, da in jeder Iteration Pheromon an den der Elitelösung entsprechenden Stellen aktualisiert wird. Dies kann dazu führen, dass der Algorithmus vorzeitig konvergiert. Insbesondere wenn in der Nachbarschaft der Elitelösung keine weiteren guten Lösungen liegen, ist diese Konzentration auf einen Teil der Suchlandschaft problematisch. Aus diesem Grund definieren wir eine maximale Anzahl g_{\max} an Generationen, in denen eine Elitelösung die Pheromonmatrix nacheinander maximal aktualisieren darf. Wenn eine Elitelösung in g_{\max} aufeinander folgenden Generationen die Pheromonmatrix aktualisiert hat, dann wird sie durch die in der aktuellen Generation beste Lösung ersetzt. Dies erfolgt selbst dann, wenn die in der aktuellen Generation gefundene Lösung schlechter ist als die bisherige Elitelösung. Wenn in der Nachbarschaft der alten Elitelösung weitere Lösungen mit einer guten Qualität liegen, dann ist die Wahrscheinlichkeit hoch, dass eine dieser guten Lösungen gefunden wird und zur neuen Elitelösung wird. Befinden sich in der Nachbarschaft keine weiteren guten Lösungen, dann ist das Verwerfen der bisherigen Elitelösung ohne große Bedeutung für das Optimierungsverhalten.

Veränderung der Parameter

Gewöhnlich sind die wichtigsten Parameter eines ACO-Algorithmus (α , β und ρ) während eines Algorithmuslaufs fix. Zahlreiche Arbeiten sind zur Thematik

erschienen, möglichst gute Parametereinstellungen für ein gegebenes Optimierungsproblem zu finden. Auch der Einfluss einer gegebenen Probleminstanz auf optimale Parametereinstellungen wurde untersucht. So wird beispielsweise in (Bottee und Bonabeau, 1998) ein genetischer Algorithmus dazu verwendet, günstige Werte für die Parameter α und β in einem ACO-Algorithmus zur Lösung des Problems des Handlungsreisenden zu finden. In Kapitel 6 wird anhand eines multikriteriellen Optimierungsproblems gezeigt, dass es vorteilhaft sein kann, innerhalb einer Iteration Ameisen mit unterschiedlichen Parameterwerten zu verwenden. Arbeiten zur Untersuchung einer Veränderung der Parameterwerte über Generationen hinweg sind bisher nicht erschienen.

Für das Problem der Projektplanung unter beschränkten Ressourcen verändern wir die Parameter β und ρ zur Laufzeit. Der Parameter β steuert den relativen Einfluss der Heuristikwerte bei der Berechnung der Auswahlwahrscheinlichkeiten. Meist hilft eine gute Heuristik dem Algorithmus in den ersten Generationen, verhältnismäßig gute Lösungen zu finden. Nach einer gewissen Zeit kann die Heuristik jedoch die Ameisen sogar davon abhalten, Entscheidungen zu treffen, die aufgrund ihrer Pheromoninformation gut sind. Eine Verbesserung der Lösungsqualität wird dadurch behindert. Dies ist insbesondere der Fall, wenn Heuristiken verwendet werden, die auf statischen Prioritätsregeln basieren (d.h. Regeln, bei denen der Heuristikwert einer Aktivität nicht vom aktuellen Stand der Planung abhängt). Ein Beispiel für eine statische Prioritätsregel ist MTS (vgl. Seite 85), bei der Aktivitäten mit einer großen Anzahl an Nachfolgern bevorzugt werden. Wir verringern den Einfluss der Heuristik dadurch, dass wir den Wert β linear zur Laufzeit auf 0 fallen lassen, und dann nur noch aufgrund der in der Pheromonmatrix gespeicherten Pheromoninformation entscheiden. Nach der Absenkung auf $\beta = 0$ bleibt β konstant.

Der Parameter ρ bestimmt die Konvergenzgeschwindigkeit des Algorithmus. Wenn wir genügend Rechenzeit verwenden können, um eine möglichst gute Lösung zu finden, dann ist es meist von Vorteil, die Verdunstungsrate ρ sehr klein zu wählen. Dadurch hat der Algorithmus die Möglichkeit, ein größeres Gebiet des Suchraumes zu durchsuchen. Eine frühe Fokussierung auf ein kleines Gebiet wird durch einen kleinen Wert ρ verhindert. Wenn jedoch die Rechenzeit begrenzt ist, dann wirkt sich ein größerer Wert für ρ im Regelfall positiv aus. Die Verdunstungsrate sollte dann so gewählt sein, dass der Algorithmus spät, aber nicht zu spät, konvergiert. Für das Problem der Projektplanung unter beschränkten Ressourcen verwenden wir daher zwei unterschiedliche Werte für ρ . Ein relativ kleiner Wert für ρ wird von Beginn an

während nahezu des gesamten Laufs verwendet. In den letzten Generationen erhöhen wir die Verdunstungsrate, um den Algorithmus am Ende intensiv in der Nähe der bisher besten gefundenen Lösung suchen zu lassen, und möglichst eine Konvergenz zu erzwingen.

Lokale Optimierung

In mehreren Arbeiten wurde gezeigt, dass eine nachgeschaltete lokale Suche auf Lösungspermutationen der Ameisen das Optimierungsverhalten von ACO-Algorithmus verbessern kann (z.B. Dorigo und Gambardella, 1997b; den Besten et al., 2000). Für das Problem der Projektplanung unter beschränkten Ressourcen untersuchen wir zwei verschiedene Methoden der lokalen Optimierung.

1. Eine lokale Suche, die in (Hartmann, 1999) in Verbindung mit einem genetischen Algorithmus vorgestellt wird, versucht, durch eine Rechts-Verschiebung von Aktivitäten in der Lösungspermutation die Lösungsqualität zu verbessern. Zur Erzeugung eines Plans aus der Lösungspermutation wird SSGS eingesetzt. In (Hartmann, 1999) werden mehrere Bedingungen bestimmt, für die die Rechts-Verschiebung einer Aktivität keine Veränderung im resultierenden Plan bewirkt. Nur wenn keines dieser Kriterien erfüllt ist, kann die Konstruktion eines Projektplans basierend auf der modifizierten Lösungspermutation zu einer Verringerung der Projektdauer führen, und der lokale Optimierungsschritt wird durchgeführt.
2. Bei Verwendung der von (Hartmann, 1999) vorgeschlagenen lokalen Suche könnte sich bei dem hier vorgestellten ACO-Algorithmus das Problem ergeben, dass eine solche Rechts-Verschiebung von Aktivitäten mit hoher Wahrscheinlichkeit schon durch die Ameisen selbst gefunden wurde. Daher betrachten wir auch eine einfache 2-opt Strategie, bei der eine Lösungspermutation neu ausgewertet wird, nachdem ein Wechsel zweier Aktivitäten durchgeführt wurde. Für ein Paar von Aktivitäten (i, j) , $i < j$, wird zunächst überprüft, ob die erzeugte Aktivitätenliste nach der Vertauschung der Aktivitäten i und j hinsichtlich der Vorrangbeziehung zulässig bleibt. Nur falls dies der Fall ist, wird die Projektdauer des neuen Projektplans berechnet. Falls diese kleiner als die des alten Projektplans ist, wird die Lösungspermutation der Ameise entsprechend modifiziert.

Bidirektionales Planen

Der Einfluss der Entscheidungsreihenfolge auf das Optimierungsverhalten wurde in Kapitel 4 untersucht. Wir haben gesehen, dass die Entscheidungsreihenfolge signifikanten Einfluss auf die Lösungsqualität haben kann. Beim Problem der Projektplanung unter beschränkten Ressourcen ist die zu bevorzugende Entscheidungsreihenfolge nicht offensichtlich. Wir werden sowohl Vorwärts- als auch Rückwärts-Ameisen zur Erzeugung von Lösungspermutationen verwenden (vgl. Abschnitt 4.1).

Die Methode, Pläne sowohl vorwärts (d.h. beginnend mit der ersten Aktivität der Lösungspermutation, die für das serielle oder parallele Prioritätsregelverfahren verwendet wird) als auch rückwärts (d.h. beginnend mit der letzten Aktivität der Lösungspermutation) zu erzeugen, wird bidirektionales Planen genannt. Bidirektionales Planen wurde für das RCPSP schon in Kombination mit unterschiedlichen zufallsgesteuerten Verfahren verwendet. In (Dorndorf et al., 2000) wird die Methode in Kombination mit einem genetischen Algorithmus verwendet. In (Valls et al., 2000) wird eine spezielle Tabusuche zur Optimierung verwendet, bei der in unterschiedlichen Algorithmusphasen entweder der Anfang oder das Ende eines Projektplans verbessert wird. Der Vorteil bidirektionalen Planens in Verbindung mit deterministischen Prioritätsregelverfahren wurde in (Klein, 2000) gezeigt.

Man beachte, dass der von Vorwärts-Ameisen mit SSGS erzeugte Projektplan sich im Allgemeinen von dem mit Rückwärts-Ameisen erzeugten Projektplan unterscheidet. In (Michels und Middendorf, 1999) wurde für das „Shortest Common Supersequence Problem“ gezeigt, dass die Verwendung von Vorwärts- und Rückwärts-Ameisen am erfolgreichsten ist, wenn die guten Lösungen, die von den beiden Ameisentypen einer Kolonie erzeugt werden, sehr ähnlich sind. Für RCPSP gilt dies meist nicht. Die Projektpläne, die von Ameisen konstruiert werden, welche die gleiche Pheromonmatrix aber unterschiedliche Einplanungsreihenfolgen verwenden, sind im Allgemeinen sehr unterschiedlich. Daher verwenden wir hier zwei Ameisenkolonien — eine mit Vorwärts-Ameisen und eine mit Rückwärts-Ameisen. Die Vorgehensweise zur Erzeugung einer Lösungspermutation bei Rückwärts-Ameisen ist identisch zu der Vorgehensweise der Vorwärts-Ameisen auf der jeweils invertierten Problem Instanz (d.h. die gerichteten Kanten im Präzedenzgraphen werden umgedreht). Beide Kolonien arbeiten unabhängig voneinander auf einer eigenen Pheromonmatrix. Nach einer vorgegebenen Anzahl an Generationen vergleichen die beiden Kolonien die Ergebnisse, indem sie beispielsweise den Durch-

schnitt der Qualität der besten Lösungen der letzten Generationen vergleichen. Man beachte, dass der Durchschnitt über die Lösungsqualität mehrerer Ameisen (und nicht etwa die bisher jeweils beste gefundene Lösung) verglichen wird, da es sinnvoll scheint, die Entscheidung in der frühen Phase des Algorithmus nicht aufgrund einer einzelnen Lösung zu treffen. Danach erzeugt nur noch die bessere der beiden Kolonien Belegungspläne.

Phasenstrukturierung

Eine den bisher in Abschnitt 5.3.2 vorgestellten Methoden übergeordnete Rolle spielt die Phasenstrukturierung des ACO-Algorithmus für RCPSP. Wir haben mehrere Erweiterungen vorgestellt, die jeweils in unterschiedlichen Phasen des Algorithmus eingesetzt werden. Zusammenfassend ergeben sich folgende fünf Phasen.

I. Phase zur Auswahl der Varianten:

Die im Abschnitt 5.3.2 auf Seite 90 vorgestellten Methoden zur Erzeugung eines Plans mit Vorwärts- und Rückwärts-Ameisen werden in dieser Phase parallel gestartet. Die Phase endet mit der Entscheidung, welche der beiden Methoden viel versprechender ist.

II. Heuristik-basierte Phase:

Man steuert den Ameisenalgorithmus durch einen zu Beginn hohen Einfluss der Heuristik in eine viel versprechende Region der Suchlandschaft. Der Einfluss der Heuristik ist in der ersten Generation am höchsten und sinkt bis zum Ende dieser Phase auf $\beta = 0$ (vgl. Abschnitt 5.3.2 auf Seite 87).

III. Hauptsuch-Phase:

Die Heuristik hat keinen Einfluss mehr. Der ACO-Algorithmus kann relativ leicht in noch viel versprechende Regionen der Suchlandschaft wechseln, konzentriert sich jedoch zunehmend auf die Suche in der bisher besten Region.

IV. Konvergenz-Phase:

Durch eine Erhöhung der Verdunstungsrate ρ wird der ACO-Algorithmus gezwungen, verstärkt in der Nähe der bisherigen Elitelösung zu suchen und zu konvergieren (vgl. Abschnitt 5.3.2 auf Seite 87).

V. Nachbearbeitungs-Phase:

Am Ende wird versucht, mit einer der in Abschnitt 5.3.2 auf Seite 89

einem deterministischen Verfahren erzeugt wurde, welches die Prioritätsregel EST (früherster Anfangszeitpunkt) verwendet.

Der Testsatz `j120.sm` enthält 600 Probleminstanzen. Jede Probleminstanz hat 120 Aktivitäten und 4 unterschiedliche Ressourcen. Der Testsatz enthält jeweils 10 Probleminstanzen für jede der möglichen Kombinationen der folgenden Parameter: $NC \in \{1.5, 1.8, 2.1\}$, $RF \in \{0.25, 0.5, 0.75, 1\}$, and $RS \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

Falls wir nicht explizit auf eine Änderung hinweisen, dann werden die Parameter des ACO-Algorithmus für alle der in diesem Kapitel vorgestellten Ergebnisse wie folgt gewählt. Die Anzahl der Ameisen pro Generation ist $m = 5$. Die maximale Anzahl der Ameisengenerationen ist 850, wobei während der ersten 100 Generationen (Phase I) zwei Kolonien mit jeweils 5 Ameisen verwendet werden. Eine Kolonie erzeugt Pläne mit Vorwärts-Ameisen, die andere erzeugt Pläne mit Rückwärts-Ameisen (vgl. Abschnitt 5.3.2). Nach 100 Generationen (Ende Phase I) wird die Kolonie fortgesetzt, deren beste Lösungen im Mittel über die letzten 25 Generationen besser waren. Entweder nach 850 Generationen (Ende Phase IV) oder falls sich die durchschnittliche Lösungsqualität einer Generation über 10 Generationen hinweg nicht geändert hat, wird eine lokale Suche angestoßen (Phase V). Die lokale Suche (vgl. 5.3.2) wird auf die bisher beste aller gefundenen Lösungen angewendet und solange durchgeführt, bis maximal 5000 Pläne erzeugt wurden (einschließlich aller von den Ameisen erzeugten Pläne). Daher werden genau 250 Schritte der lokalen Suche durchgeführt, falls von den Ameisen in 850 Generationen Lösungen erzeugt wurden, und die lokale Suche nicht aufgrund frühzeitiger Konvergenz gestartet wurde.

Solange nicht explizit anders spezifiziert, verwenden wir $\alpha = 1, c = 0.5$ und $\gamma = 1.0$. Mit $c = 0.5$ und $\gamma = 1.0$ haben die lokale und die Summenpheromonauswertung den gleichen Einfluss. Die Pheromonwerte, die früheren Entscheidungen entsprechen, haben alle das gleiche Gewicht bei der Berechnung der Auswahlwahrscheinlichkeiten. Der Parameter β wird linear von 2 in der ersten Generation bis auf 0 nach 50% der maximalen Anzahl an Generationen abgesenkt (Phase I und Phase II). Bei maximal 5000 zu erzeugenden Plänen bedeutet das, dass nach 425 Generationen $\beta = 0$ erreicht ist und Phase III gestartet wird. Die Verdunstungsrate ist $\rho = 0.025$ und wird in den letzten 200 Generationen auf $\rho = 0.075$ erhöht (Phase IV). Die Elitelösung wird nach spätestens $g_{\max} = 10$ Generationen verworfen. Als Standardheuristik verwenden wir die nLST Heuristik.

Mit ACO-RCPSP bezeichnen wir den ACO-Algorithmus für RCPSP mit

den Parametereinstellungen, wie sie oben erläutert wurden. Mit s-ACO-RCPSP bezeichnen wir eine einfachere Variante des ACO-Algorithmus, der bis auf die Kombination der lokalen und der Summen-Pheromonauswertung die Erweiterungen aus Kapitel 5.3.2 nicht verwendet. s-ACO-RCPSP erzeugt Pläne nur mit Vorwärts-Ameisen, es wird ein konstanter Einfluss der Heuristik von $\beta = 1$ verwendet, und die Verdunstungsrate beträgt konstant $\rho = 0.025$. Eine lokale Suche wird nicht eingesetzt, d.h. der Algorithmus läuft 1000 Generationen mit jeweils $m = 5$ Ameisen.

Alle Tests wurden auf einem Rechner mit einem Pentium III 500MHz Prozessor durchgeführt. Ein Lauf von ACO-RCPSP benötigt etwa 25 Sekunden Rechenzeit für eine Problem Instanz.

Die Ergebnisse der Messungen werden mit einer unteren Schranke (LB) verglichen, die für jede Problem Instanz durch eine Heuristik berechnet wurde, die auf der Berechnung des kritischen Pfades des Präzedenzgraphen beruht (Stinson et al., 1978). Falls nicht explizit darauf hingewiesen wird, dann ist jedes Ergebnis im folgenden Abschnitt der Mittelwert der Ergebnisse über alle 600 Problem Instanzen mit je 4 Wiederholungen für jede der Instanzen.

5.5 Ergebnisse

5.5.1 s-ACO-RCPSP

Die einfache Variante des ACO-Algorithmus für RCPSP wurde für verschiedene Kombinationen von c und γ sowohl für das serielle als auch für das parallele Prioritätsregelverfahren untersucht. Die Kombination der lokalen und der Summen-Pheromonauswertung wurde für Werte $c \in \{0, 0.2, 0.4, 0.5, 0.6, 0.8, 1.0\}$ und $\gamma \in \{0, 0.2, 0.4, 0.6, 0.8, 0.9, 1.0, 1.1, 1.2, 1.4\}$ getestet. Die Ergebnisse sind in den Abbildungen 5.3 und 5.4 dargestellt. Die Resultate zeigen, dass eine Kombination der lokalen Pheromonauswertung mit der Summen-Pheromonauswertung sowohl für SSGS als auch für PSGS auch hier zu den besten Ergebnissen führt (vgl. Abschnitt 3.1). Die geringste Abweichung zur unteren Schranke (36.7%) wurde mit Hilfe des seriellen Prioritätsregelverfahren und Parameterwerten $c = 0.6$ und $\gamma = 1$ erreicht. Für das parallele Prioritätsregelverfahren ist das beste Ergebnis eine Abweichung von 37.6% von der unteren Schranke bei Parametereinstellungen $c = 0.8$ und $\gamma = 1.1$. Sowohl bei SSGS als auch bei PSGS wurden die besten Ergebnisse erzielt, wenn der relative Einfluss der Summen-Pheromonauswertung leicht stärker und die Gewichtung der Pheromonwerte früherer Entscheidungen etwa gleich war. Nahezu immer ergaben

sich die schlechtesten Ergebnisse bei der Verwendung der reinen lokalen Pheromonauswertung ($c = 1.0$ oder $\gamma = 0$). Die Abweichungen betragen 40.3% für SSGS und 38.5% für PSGS. Für das parallele Prioritätsregelverfahren ergaben sich die schlechtesten Ergebnisse (38.6% Abweichung von der unteren Schranke) für $c = 0$ und $\gamma > 1.0$. Im Vergleich hierzu erzielte die Summen-Pheromonauswertung viel bessere Ergebnisse, wenn die γ -Werte weder zu groß noch zu klein waren — bei SSGS und $\gamma = 0.9$ war die mittlere Abweichung nur 37.1% und bei PSGS und $\gamma \in \{0.4, 0.6, 0.8\}$ war die mittlere Abweichung nur 38.0%. ACO-RCPSP mit seriellem Prioritätsregelverfahren erzielte die besten Ergebnisse. In Verbindung mit dem parallelen Prioritätsregelverfahren scheint ACO-RCPSP jedoch robuster gegenüber Veränderungen der Parameter c und γ zu sein. Dass die besten Ergebnisse mit dem seriellen Prioritätsregelverfahren erzielt wurden, steht im Gegensatz zu den Ergebnissen, die in (Klein, 2000) und (Hartmann, 1999) publiziert wurden. Dort wurde gezeigt, dass auf den größten Probleminstanzen der PSPLIB (`j120.sm`) das parallele Prioritätsregelverfahren in Kombination mit einem genetischen Algorithmus oder auch in Kombination mit mehreren unterschiedlichen deterministischen Heuristiken im Mittel bessere Ergebnisse liefert. Da die besten Ergebnisse beim ACO-Algorithmus mit SSGS gefunden wurden, werden die weiteren Untersuchungen mit dem seriellen Prioritätsregelverfahren durchgeführt.

Bei der Untersuchung der Summen-Pheromonauswertung und der Einplanungsreihenfolge auf einfachen Permutationsproblemen (vgl. Kapitel 2) haben wir durchschnittliche Pheromonmatrizen visualisiert. Wir haben dadurch direkt sehen können, wie sich die Verfahrensänderungen auf die Pheromonmatrix auswirken. Eine wichtige Beobachtung war, dass die lokale Pheromonauswertung eine ungleichmäßige Konvergenz in der Pheromonmatrix verursacht — tendenziell konvergieren erst Zeilen mit niedrigem Index. Die Summen-Pheromonauswertung hat das Konvergenzverhalten deutlich verbessert, was sich auch in der Lösungsqualität zeigte. Die Visualisierung der Pheromonmatrizen wurde für einfache Probleminstanzen durchgeführt. Für RCPSP wird der Einfluss der lokalen Pheromonauswertung und der Summen-Pheromonauswertung nun auch genauer untersucht, indem wir die Entropie (Shannon, 1948) der von Ameisen verwendeten Wahrscheinlichkeitsverteilungen berechnen. Die Entropie ist unter anderem ein Maß für die Konzentration der Verteilung. Wir berechnen für jede i -te Entscheidung einer Ameise ($i \in [1, 120]$) während der Konstruktion einer Lösung die Entropie

$$e_i = - \sum_{j \in \mathcal{E}} p_{ij} \log p_{ij}$$

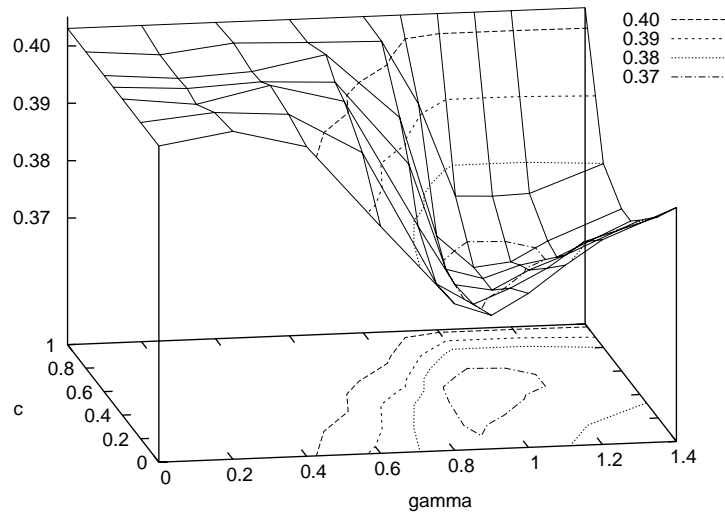


Abbildung 5.3: Resultate für s-ACO-RCPSP in Kombination mit SSGS für unterschiedliche Werte c und γ ; dargestellt ist die prozentuale Abweichung von LB (untere Schranke, die auf der Berechnung des kritischen Pfades basiert)

Die Entropiewerte geben an, wie weit der Algorithmus hinsichtlich der einzelnen Entscheidungen der Ameisen schon konvergiert ist. Eine Verteilung hat den maximalen Entropiewert, wenn jede Entscheidung gleich wahrscheinlich ist.

Im Gegensatz zu den meisten anderen Ergebnissen dieses Kapitels, bei denen jeweils maximal 5000 Pläne erzeugt wurden, basieren die Ergebnisse der Entropiebetrachtung auf einem ACO-Algorithmus, der pro Instanz 2000 Generationen lang lief. Dabei wurden jeweils 10000 Pläne erzeugt. Durch die höhere Anzahl an Generationen wurde eine Konvergenz sicher gestellt.

Abbildung 5.5 und 5.6 zeigen die Entropiekurven für unterschiedliche Generationen während eines Laufs. Jeder Punkt in der Abbildung wurde durch einen Mittelwert aus 300 Entropiewerten berechnet, die man bei der Berechnung der Entropiewerte von 60 Probleminstanzen mit jeweils 5 Ameisen pro Generation erhält. Die gewählten Instanzen waren alle jeweils die erste Instanz der 60 unterschiedlichen Typen von Instanzen aus dem Testsatz j120.sm. Ei-

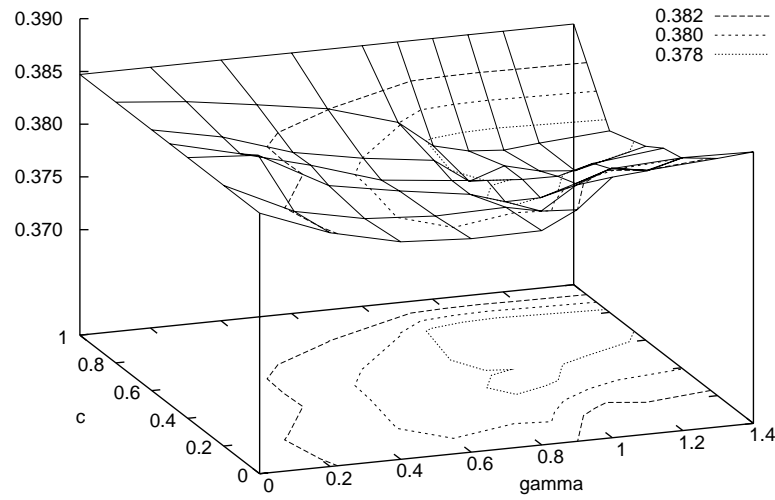


Abbildung 5.4: Resultate für s-ACO-RCPSP in Kombination mit PSGS für unterschiedliche Werte c und γ ; dargestellt ist die prozentuale Abweichung von LB

ne Beobachtung ist, dass die Entropiewerte der Entscheidungen in der Mitte der Lösungspermutation höher sind, als jene, die Entscheidungen am Anfang oder am Ende einer Lösungspermutation entsprechen. Dies liegt daran, dass die Menge der wählbaren Aktivitäten für die entsprechenden Entscheidungen größer ist. Des Weiteren sind viel kleinere Entropiewerte für die lokale Pheromonauswertung, als für die Kombination von lokaler und Summen-Pheromonauswertung zu beobachten.

Das ungleichmäßige Konvergenzverhalten der lokalen Pheromonauswertung und die Verbesserung des Konvergenzverhaltens der Summen-Pheromonauswertung lässt sich mit Hilfe der Entropieanalyse auch auf dem Problem der Projektplanung unter beschränkten Ressourcen zeigen. Bei der lokalen Pheromonauswertung ($c = 1$) sinkt die Entropie bei den ersten Entscheidungen sehr schnell. Alle Entropiewerte der ersten 60 Entscheidungen sind nach Generation 750 kleiner als 0.2. Im Gegensatz dazu zeigt die Kombination aus lokaler und Summen-Pheromonauswertung ($c = 0.5$) nahezu symmetrische Entropiekurven, d.h. die Wahrscheinlichkeitsverteilungen der Entscheidungen

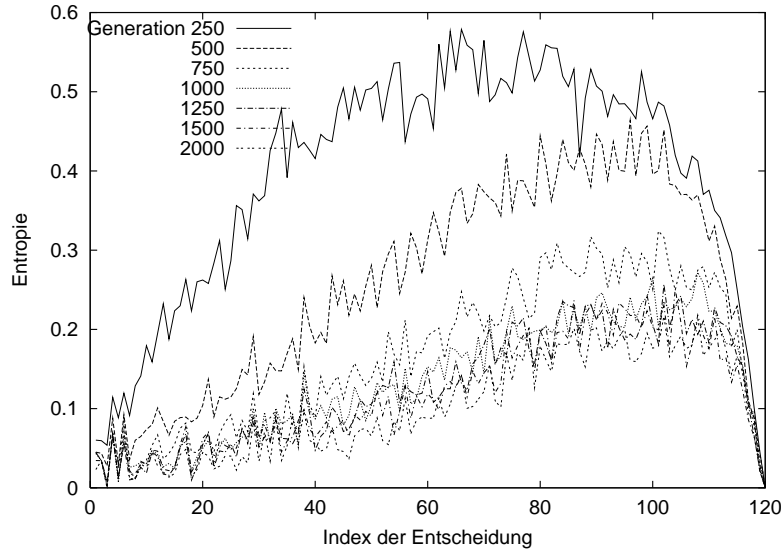


Abbildung 5.5: Entropie der Wahrscheinlichkeitsverteilungen für s-ACO-RCPSP mit $c = 1$ (lokale Pheromonauswertung): die Ergebnisse sind Mittelwerte über 5 Ameisen pro Generation und 60 Probleminstanzen

am Anfang und am Ende haben etwa die gleiche Entropie. Die Entropiewerte sind im Vergleich zur lokalen Pheromonauswertung auch höher als bei der Summen-Pheromonauswertung. Dies deutet darauf hin, dass die Kombination der Auswertungen den Algorithmus hier, wie bei einfachen Permutationsproblemen auch, von einer ungleichmäßigen und zu schnellen Konvergenz abhält.

5.5.2 ACO-RCPSP

Die Ergebnisse des verbesserten ACO-Algorithmus mit seriellem Prioritätsregelverfahren sind in Abbildung 5.7 und Tabelle 5.2 dargestellt. Die Ergebnisse sind denen von s-ACO-RCPSP sehr ähnlich: die besten Ergebnisse wurden für $c = 0.5$ und $\gamma = 1$ erzielt. Die Abweichung zur unteren Schranke beträgt im Mittel jedoch nur 35.43% (bester Wert: 35.35% bei 10 Wiederholungen). Die schlechtesten Ergebnisse liefert wiederum die reine lokale Pheromonauswertung ($c = 1.0$ oder $\gamma = 0$) mit einer Abweichung von 38.24%.

ACO-RCPSP wurde mit vielen anderen Heuristiken für RCPSP verglichen. Mehrere dieser Heuristiken sind in einer umfassenden Studie von (Hartmann und Kolisch, 2000) erläutert. Die Studie betrachtet folgende Heuristiken:

1. Drei Heuristiken mit normierten Prioritätsregeln; (Kolisch, 1996a,b) (heu-

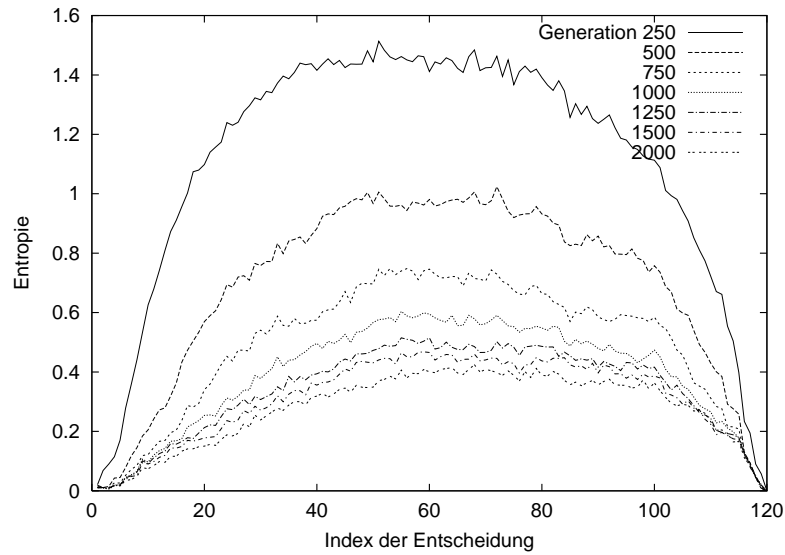


Abbildung 5.6: Entropie der Wahrscheinlichkeitsverteilungen für s-ACO-RCPSP mit $c = 0.5$ (Kombination aus lokaler und Summen-Pheromonauswertung): die Ergebnisse sind Mittelwerte über 5 Ameisen pro Generation und 60 Probleminstanzen

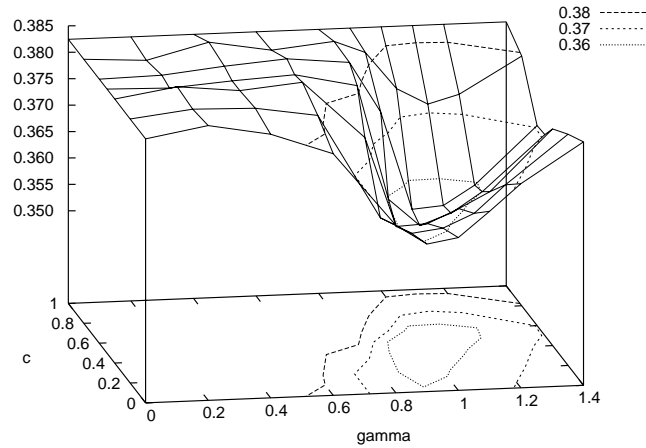


Abbildung 5.7: Resultat für unterschiedliche Werte c und γ für ACO-RCPSP; dargestellt ist die prozentuale Abweichung von LB

Tabelle 5.2: Resultate für unterschiedliche Werte c und γ für ACO-RCPS: die besten Ergebnisse jeder Spalte sind fett gedruckt; das insgesamt beste Ergebnis ist unterstrichen

γ	c						
	0.0	0.2	0.4	0.5	0.6	0.8	1.0
0.0	38.2	38.2	38.2	38.2	38.2	38.2	38.2
0.2	38.4	38.4	38.4	38.2	38.3	38.2	38.2
0.4	38.2	38.4	38.3	38.3	38.4	38.8	38.2
0.6	37.8	38.2	38.4	38.4	38.5	38.8	38.2
0.8	36.4	36.2	37.3	38.1	38.4	38.8	38.2
0.9	36.0	35.9	35.6	35.9	37.4	38.6	38.2
1.0	36.1	35.9	35.6	<u>35.4</u>	35.5	38.0	38.2
1.1	36.6	36.2	35.9	35.6	35.6	37.6	38.2
1.2	37.1	36.7	36.2	36.0	36.0	37.9	38.2
1.4	37.9	37.6	37.3	37.0	37.0	38.4	38.2

istic with regret based random sampling): Ähnlich wie bei der Berechnung der normierten Heuristikwerte werden die Werte der verwendeten Prioritätsregeln normiert. Die Wahl der nächsten Aktivität beim seriellen oder parallelen Prioritätsregelverfahren erfolgt entsprechend einer Wahrscheinlichkeitsverteilung, die auf normierten Werten einer Prioritätsregel basiert.

2. Zwei Heuristiken ähnlich wie in 1.; (Kolisch und Drexl, 1996) und (Schirmer, 2000) (heuristics with adaptive regret-based random sampling): Bei dieser Erweiterung wird in einem Vorverarbeitungsschritt die betrachtete Instanz analysiert. Dann wird die verwendete Prioritätsregel beispielsweise entsprechend dem Parameter Resource Strength gewählt.
3. Vier unterschiedliche genetische Algorithmen; (Hartmann, 1998) und (Leon und Ramamoorthy, 1995)
4. Ein auf Simulated Annealing (SA) basierendes Verfahren; (Bouleimen und Lecocq, 1998)
5. Beim in (Hartmann, 1999) vorgestellten selbst-adaptiven genetischen Algorithmus ändert sich während eines Laufs der Anteil der Pläne, die mit dem seriellen und mit dem parallelen Prioritätsregelverfahren erzeugt

werden. Eine lokale Suche wird gestartet, wenn sich die beste aller gefundenen Lösung über mehrere Generationen nicht verbessert hat (self-adapting GA).

In den vergleichenden Studien (Hartmann und Kolisch, 2000) und (Hartmann, 1999) erzeugte jede der Heuristiken maximal 5000 Pläne für jede der Probleminstanzen. In Tabelle 5.3 werden die Ergebnisse der Arbeiten mit ACO-RCPSP verglichen. Zusätzlich zu den Resultaten der schon erwähnten Heuristiken werden auch Ergebnisse der folgenden Arbeiten aufgeführt. Man beachte, dass die in 6. und 7. verwendeten Methoden nicht die maximale Obergrenze von 5000 erzeugten Plänen einhalten, jedoch bezüglich der Rechenzeit grob vergleichbar mit den weiteren Verfahren sind.

6. Der in (Möhring et al., 2000) vorgestellte Ansatz beruht auf der Methodik der Lagrange Relaxierung und der ganzzahligen linearen Programmierung. Die Ressourcenbeschränkungen werden relaxiert und die Verletzung wird durch eine entsprechende Veränderung des Zielfunktionswerts bestraft. Mit Hilfe dieses Ansatzes wird eine Prioritätsliste erzeugt, die dann durch ein modifiziertes serielles Prioritätsregelverfahren einen Plan erzeugt (LS mit LR-IP).
7. In (Dorndorf et al., 2000) wird ein auf Branch-and-Bound basierendes Verfahren vorgestellt und untersucht (B&B).

Ein „*“ in Tabelle 5.3 bedeutet, dass das entsprechende Ergebnis statistisch signifikant besser als das nächste mit „*“ versehene Ergebnis der Tabelle ist. Das Signifikanzniveau beträgt 5% bei Verwendung des Wilcoxon-Vorzeichen-Rang Tests. Alle Signifikanztests, bis auf den Signifikanztest für ACO-RCPSP, wurden in (Hartmann, 1999) durchgeführt. Man beachte, dass ein fehlendes „*“ nur bedeutet, dass kein Test durchgeführt wurde. Es bedeutet nicht notwendigerweise, dass keine statistische Signifikanz vorhanden ist.

ACO-RCPSP hat mit einer durchschnittlichen prozentualen Abweichung von 35.43% das beste Ergebnis erzielt. Die Abweichung ist hierbei der Mittelwert aus 10 Wiederholungen auf jeweils allen 600 Probleminstanzen. Die Standardabweichung beträgt 0.046, die minimal erreichte prozentuale Abweichung ist 35.35%, die maximale prozentuale Abweichung beträgt 35.50%. Diese maximale prozentuale Abweichung ist besser als die des zweitbesten Algorithmus, der in Tabelle 5.3 aufgeführt ist. Mit dem genetischen Algorithmus (Hartmann, 1999) wurde eine prozentuale Abweichung von 35.60% erreicht. Das drittbeste Ergebnis stammt mit 36.2% von (Möhring et al., 2000). Beim

Tabelle 5.3: Vergleich von ACO-RCPSP mit unterschiedlichen zufallsbasierten Verfahren: mit jedem Verfahren wurden pro Problem Instanz maximal 5000 Pläne erzeugt und ausgewertet (bis auf die Verfahren mit geklammerten Abweichungen); „*“ bedeutet, dass der Unterschied zweier aufeinander folgender Ergebnisse statistisch signifikant ist; ACO-RCPSP wurde 10 mal wiederholt, angegeben ist der Durchschnittswert, bester Wert der 10 Wiederholungen: 35.35%, schlechtester Wert: 35.50%, Standardabweichung: 0.046

Algorithmus	Referenz	Abweichung von LB in %
ACO-RCPSP		35.43*
Selbst-adaptiver GA	(Hartmann, 1999)	35.60*
LS mit LR-IP	(Möhring et al., 2000)	(36.2)
GA 1	(Hartmann, 1998)	36.74*
B&B	(Dorndorf et al., 2000)	(37.1)
SA	(Bouleimen und Lecocq, 1998)	37.68*
GA 2	(Hartmann, 1998)	38.49
adaptive sampling 1	(Schirmer, 2000)	38.70
single pass/sampling 1	(Kolisch, 1996b)	38.75
single pass/sampling 2	(Kolisch, 1996a, b)	38.77*
adaptive sampling 2	(Kolisch und Drexler, 1996)	40.45*
GA 3	(Leon und Ramamoorthy, 1995)	40.69*
single pass/sampling 3	(Kolisch, 1996b)	41.84*
GA 4	(Hartmann, 1998)	42.25
random sampling 1	(Kolisch, 1995)	43.05*
random sampling 2	(Kolisch, 1995)	47.61

Ansatz von Möhring et al. wurde die maximale Obergrenze von 5000 Plänen nicht eingehalten. Im Durchschnitt wurden pro Instanz 3675 in 65 Sekunden erzeugt. Der verwendete Rechner war eine SUN Ultra 2 mit einem 200MHz Prozessor. Der auf Branch-and-Bound basierende Ansatz von Dorndorf et al. erreicht eine prozentuale Abweichung von 37.5% (37.1%) bei 60 Sekunden (300 Sekunden) Rechenzeit auf einem Pentium Pro/200 PC.

Zwei weitere Heuristiken für RCPSP sind nicht in der Tabelle 5.3 aufgeführt, da die Ergebnisse bezüglich des Rechenaufwands nicht direkt vergleichbar sind. Im Ansatz von (Valls et al., 2000) wird versucht die Länge der Folge von kritischen Aktivitäten eines zulässigen Plans sukzessive zu verkürzen. Die

maximale Rechenzeit für eine Instanz im Datensatz `j120.sm` beträgt 44 Sekunden auf einem PC mit AMD 400MHz Prozessor. Ohne die Beschränkung der Anzahl der maximal auszuwertenden Pläne, wurde eine prozentuale Abweichung von 34.53% erreicht. Ein Ansatz, der das Prinzip der Tabusuche verwendet, wurde in (Nonobe und Ibaraki, 1999) vorgestellt. Mit maximal 30000 Iterationen pro Instanz des Datensatzes `j120.sm` wurde eine prozentuale Abweichung von 34.99% erreicht. In (Nonobe und Ibaraki, 1999) wurde für 219 der 600 Probleminstanzen des Datensatzes `j120.sm` eine Lösung gefunden, die mindestens genauso gut war, wie die zum Zeitpunkt der Untersuchung (Juni 1999) besten bekannten Lösungen. Für 50 Instanzen wurde die obere Schranke verbessert. Ein Lauf benötigte ca. 10 Minuten auf einem Sun Ultra 2 Rechner mit einem 300MHz Prozessor und 1GB Hauptspeicher. Im Vergleich hierzu wurden mit dem Ameisenalgorithmus auf den deutlich verbesserten Ergebnissen im Januar 2001 mit ca. 25 Sekunden Rechenzeit auf einem Pentium III Rechner mit einem 500MHz Prozessor für 186 Instanzen eine Lösung gefunden, die mindestens genauso gut wie die bisher beste bekannte Lösung war. Man beachte, dass die in diesem Kapitel vorgestellten Ameisenalgorithmen nicht speziell so entworfen wurden, dass ein einzelner Plan schnell erzeugt werden kann. Vielmehr war das Ziel, ein sehr gutes Optimierungsverhalten mit einer bestimmten Anzahl zu erzeugender Pläne zu erhalten.

Im Folgenden untersuchen wir die von ACO-RCPSPP eingesetzten Methoden genauer.

Bidirektionales Planen

Am Anfang eines Messlaufs laufen 100 Generationen lang zwei Kolonien parallel. Die eine erzeugt Pläne beginnend mit der ersten Aktivität der Lösungspermutation, die andere beginnend mit der letzten Aktivität der Lösungspermutation. Falls nur Pläne mit Vorwärts-Ameisen erzeugt werden, dann erreicht man eine prozentuale Abweichung von 35.94%. Erzeugt man Pläne nur mit Rückwärts-Ameisen, dann erreicht man eine prozentuale Abweichung von 35.77%. Da beide Ergebnisse deutlich schlechter sind, als die Abweichung von 35.43%, die ACO-RCPSPP erreicht, können die Ameisen erfolgreich nach 100 Generationen entscheiden, welche Einplanungsrichtung auf der entsprechenden Instanz vorteilhafter erscheint. Im Durchschnitt fanden in 38.6% aller Fälle die Vorwärts-Ameisen mit dem in Abschnitt 5.4 erläuterten Entscheidungskriterium nach 100 Generationen das bessere Ergebnis. In 51.6% aller Fälle fanden die Rückwärts-Ameisen das bessere Ergebnis. In 9.9% aller Fälle

waren beide Kolonien gleich gut. Im Allgemeinen scheint es auf dem Datensatz `j120.sm` so zu sein, dass es dem Ameisenalgorithmus etwas einfacher fällt, mit Rückwärts-Ameisen (oder auf invertierten Instanzen) gute Ergebnisse hinsichtlich der prozentualen Abweichung zur unteren Schranke zu erreichen. Dies wird auch in den Abbildungen der nächsten Abschnitte zu sehen sein. In Abschnitt 5.3.2 wurde ein Verfahren vorgeschlagen, das, basierend auf Durchschnittswerten der besten Ameisen der letzten Generationen, entscheidet, welche der beiden Kolonien fortgesetzt wird. Dieses Verfahren lieferte immer bessere Ergebnisse, als die Methode, bei der einfach die Kolonie fortgesetzt wird, welche nach 100 Generationen die bisher global beste Lösung gefunden hat.

Veränderung des Heuristikeinflusses

In diesem Abschnitt vergleichen wir die Ergebnisse, die sich bei unterschiedlich starken (aber konstanten) Einflüssen der Heuristik bei der Berechnung der Auswahlwahrscheinlichkeiten ergeben, mit dem Resultat, welches sich durch eine Verringerung des Heuristikeinflusses zur Laufzeit ergibt. Bei ACO-RCPSP wird β linear von 2 in der ersten Generation auf 0 nach der Hälfte aller Generationen gesenkt. Abbildung 5.8 zeigt das Verhalten von ACO-RCPSP für unterschiedliche konstante Werte von β . Das beste Ergebnis ($\beta = 1.25$) ist eine durchschnittliche prozentuale Abweichung von 35.71% von der unteren Schranke. Dies ist deutlich schlechter als die durchschnittliche Abweichung von 35.43%, die erreicht wird, wenn man β wie eben erläutert zur Laufzeit verändert.

Veränderung der Verdunstungsrate

Ein weiterer Parameter, der bei ACO-RCPSP während der Laufzeit verändert wird, ist die Verdunstungsrate ρ . Diese ist zu Beginn $\rho = 0.025$. In den letzten 200 Generationen eines Laufs, wird die Verdunstungsrate auf $\rho = 0.075$ erhöht. Abbildung 5.9 zeigt die Ergebnisse für ACO-RCPSP, falls die Verdunstungsrate nicht verändert wird, sondern konstant bleibt. Das beste Ergebnis, welches dann erzielt wird, ist eine prozentuale Abweichung von 35.58% bei einer Verdunstungsrate von $\rho = 0.04$. Wie beim Vergleich des konstanten bzw. zur Laufzeit angepassten Heuristikeinflusses, ist das Ergebnis für das beste konstante ρ deutlich schlechter als die durchschnittliche Abweichung von 35.43%, die man durch eine Erhöhung von ρ am Ende eines Laufs erhält.

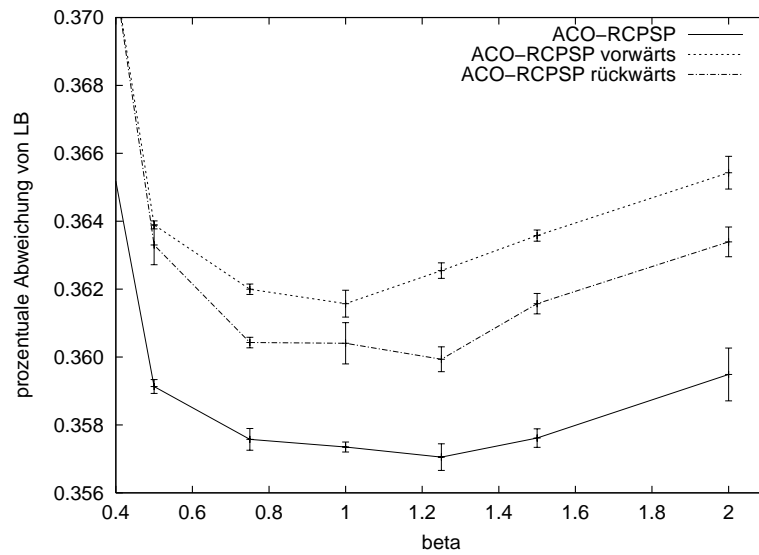


Abbildung 5.8: ACO-RCPSP mit unterschiedlichen konstanten Werten für β (die Balken geben die Standardabweichung an)

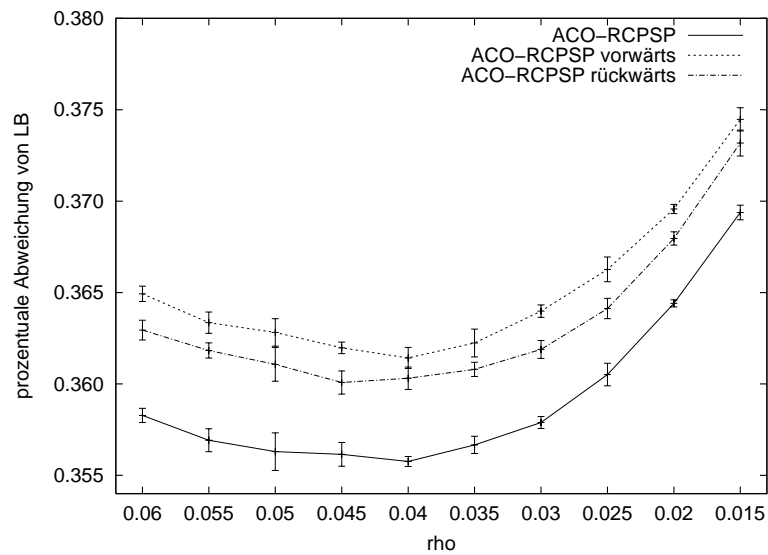


Abbildung 5.9: ACO-RCPSP mit unterschiedlichen konstanten Werten für ρ (die Balken geben die Standardabweichung an)

Verwerfen der Elitelösung

Bei ACO-RCPSP wird der Einfluss einer Elitelösung dadurch eingeschränkt, dass diese nach maximal $g_{\max} = 10$ Generationen verworfen wird. Abbildung

5.10 zeigt das Verhalten von ACO-RCPSP bei der Verwendung unterschiedlicher Werte für g_{\max} . Es ist deutlich zu erkennen, dass g_{\max} einen starken Einfluss auf das Optimierungsverhalten hat. Mit einem hohen Wert ($g_{\max} = 200$) wurde eine prozentuale Abweichung von 36.0% erreicht. Bei der Verwendung von einem kleinen Wert ($g_{\max} = 3$) wurde eine prozentuale Abweichung von 35.77% erreicht. Beides ist deutlich schlechter als die durchschnittliche Abweichung von 35.43%, die für $g_{\max} = 10$ erreicht wird.

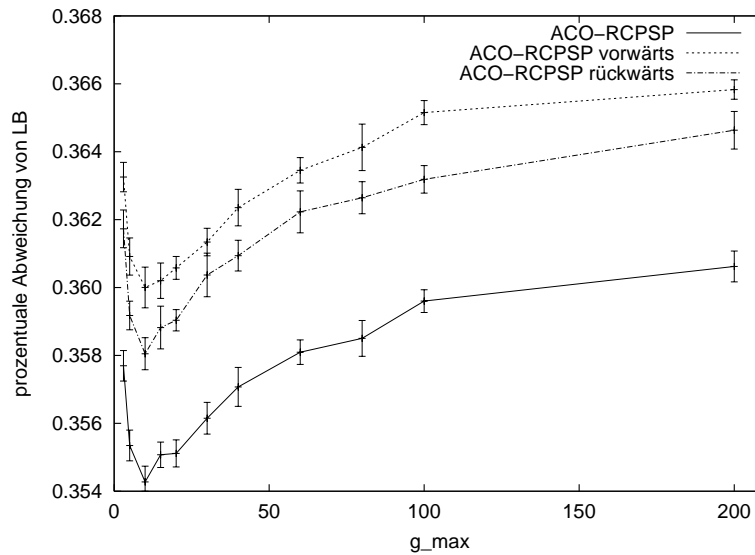


Abbildung 5.10: ACO-RCPSP mit unterschiedlichen Werten für g_{\max} (die Balken geben die Standardabweichung an)

Lokale Optimierung

Der Einfluss der lokalen Optimierung am Ende eines Laufs ist in Abbildung 5.11 dargestellt. Die Abbildung zeigt, dass die Anzahl der Schritte für eine lokale Optimierung am Ende eines Laufs nur einen relativ schwachen Einfluss hat. Ohne lokale Optimierung erreicht ACO-RCPSP im Mittel eine prozentuale Abweichung von 35.51% von der unteren Schranke. Bei 250 lokalen Optimierungsschritten wird das beste Ergebnis erzielt (35.43%). Da jeder Lauf von ACO-RCPSP genau 5000 Pläne erzeugt und auswertet, reduziert die Erhöhung der Anzahl der lokalen Optimierungsschritte die Anzahl der Generationen des ACO-Algorithmus im Verhältnis 5:1. Daher führt eine zu hohe Anzahl an lokalen Optimierungsschritten zu schlechteren Ergebnissen. Bei 1000 lokalen Optimierungsschritten beträgt die prozentuale Abweichung 35.56%.

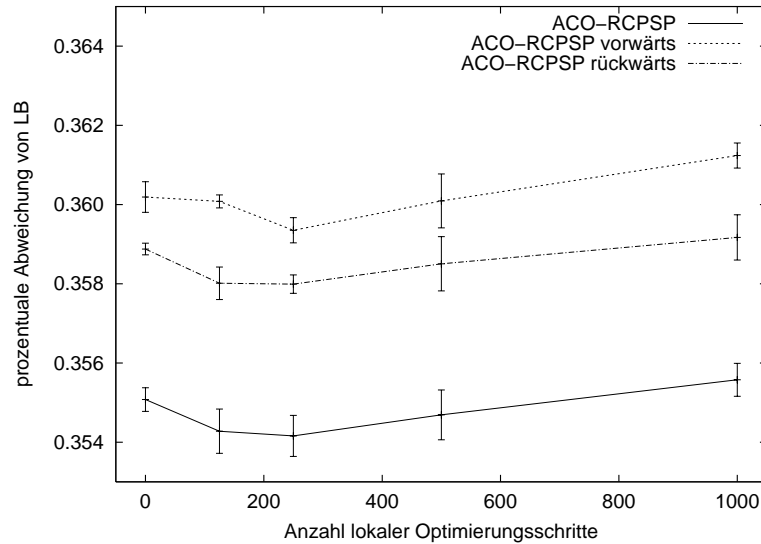


Abbildung 5.11: ACO-RCPSP mit unterschiedlichen Werten lokaler Optimierungsschritte am Ende eines Laufs (die Balken geben die Standardabweichung an)

Anstatt die 2-opt Strategie bei der lokalen Optimierung einzusetzen, wurde auch die in (Hartmann, 1999) vorgeschlagene Methode zur lokalen Optimierung getestet. Mit 250 lokalen Optimierungsschritten wie in (Hartmann, 1999) vorgeschlagen, wird eine prozentuale Abweichung von der unteren Schranke von 35.46% bei einer Standardabweichung von 0.088 erreicht. Die 2-opt Strategie ist somit geringfügig besser.

Heuristiken

Die Ergebnisse, die mit Hilfe der auf Prioritätsregeln basierenden Heuristiken, wie sie in Abschnitt 5.3.1 vorgestellt wurden, erreicht wurden, sind in Tabelle 5.4 dargestellt. Mit einer prozentualen Abweichung von der unteren Schranke von 35.64%, ist die nLFT Heuristik nur leicht schlechter, als die nLST Heuristik, die beim Standard ACO-RCPSP Algorithmus verwendet wurde. Die nGRPWA-Heuristik liefert auch noch gute Ergebnisse mit einer prozentualen Abweichung von 35.79%. nMTS liefert nicht ganz so gute Ergebnisse. nWRUP liefert für alle drei getesteten Parametereinstellungen $\omega \in \{0, 0.5, 1\}$ die schlechtesten Ergebnisse. Diese Beobachtungen stimmen mit denen von (Klein, 2000) überein, bei denen die gleiche Rangfolge der Prioritätsregeln bei der Verwendung von SSGS beobachtet wurde.

Tabelle 5.4: ACO-RCPSP mit unterschiedlichen auf Prioritätsregeln basierenden Heuristiken: angegeben ist die prozentuale Abweichung von der unteren Schranke

Heuristik	Abweichung von LB in %	Standard- abweichung
nLST	35.43	0.046
nLFT	35.64	0.043
nGRPWA	35.79	0.049
nMTS	36.38	0.004
nWRUP ($\omega = 0$)	39.22	0.080
nWRUP ($\omega = 0.5$)	38.32	0.082
nWRUP ($\omega = 1$)	38.70	0.037

Weitere Ergebnisse

Um das Potential des ACO-Algorithmus für RCPSP weiter zu untersuchen, wurde s-ACO-RCPSP mit zusätzlichen lokalen Optimierungsschritten zwischen den einzelnen Generationen getestet. Die verwendeten Parameter waren $m = 20$ Ameisen, $\rho = 0.005$, maximal 20000 Generationen und die Anwendung einer lokalen Suche auf die jeweils beste Ameise jeder Generation. Der so veränderte Algorithmus erreichte eine prozentuale Abweichung von 32.97% von der unteren Schranke. Für 278 der 600 Instanzen wurde eine Lösung gefunden, die mindestens so gut wie die beste bekannte Lösung ist (Stand: Januar 2000).

Ein weiterer Beweis für die Stärke des Algorithmus ist, dass für viele der 600 Probleminstanzen neue beste Lösungen gefunden wurden. Bei über 150 der 600 Probleminstanzen wurde die bisher beste bekannte Lösung mit Hilfe des hier vorgestellten Verfahrens verbessert (Stand: August 2000). Dies sind über 1/3 aller Probleminstanzen, für die eine optimale Lösung noch nicht bekannt ist.

Manche der Parameter von ACO-RCPSP sind nur wichtig, wenn die Anzahl der zu erzeugenden Pläne beschränkt ist. Falls die eingeschränkte Anzahl an maximal zu erzeugenden Projektplänen keine Einschränkung darstellt, dann kann man, anstatt wie bei ACO-RCPSP in der Anfangsphase zwei Kolonien zu verwenden, den Algorithmus auch mit jeweils unterschiedlicher Einplanungsrichtung zweimal laufen lassen. Die Verdunstungsrate ρ kann konstant auf einem kleineren Wert bleiben, anstatt am Ende eines Laufs erhöht zu werden. Eine geringere Verdunstungsrate bedeutet jedoch im Allgemeinen,

dass der Algorithmus erst später konvergiert, und somit eine größere Rechenzeit benötigt. Die Anzahl der lokalen Optimierungsschritte am Ende eines Laufs sollte bei hinreichender Rechenzeit nicht zu klein sein, denn nach einigen Hundert lokalen Optimierungsschritten wird die Wahrscheinlichkeit einer weiteren Verbesserung durch einen lokalen Optimierungsschritt sehr klein. Je mehr Ameisen pro Generation verwendet werden, desto besser werden im Allgemeinen die Ergebnisse sein. Oft genügen jedoch schon wenige Ameisen pro Generation zur Erzeugung sehr guter Lösungen. Die Summen-Pheromonauswertung sollte einen wichtigen Einfluss bei der Berechnung der Auswahlwahrscheinlichkeiten der Ameisen haben ($c = 0.5$), und die Eliteameise sollte nach einigen Generationen verworfen werden. Während den ersten Generationen sollte der Einfluss der Heuristik mindestens so groß sein wie der Einfluss der Pheromone ($\alpha = 1$, $\beta \geq 1$), jedoch sollte der Einfluss der Heuristik nach einer gewissen Zeit deutlich unter den Einfluss der Pheromonwerte sinken.

5.6 Zusammenfassung

Es wurde ein ACO-Algorithmus für das Problem der Projektplanung unter beschränkten Ressourcen (RCPSp) vorgestellt. Die in Kapitel 3 eingeführte Summen-Pheromonauswertung und insbesondere die Kombination der lokalen Pheromonauswertung mit der gewichteten Summen-Pheromonauswertung wurden erfolgreich angewendet. Neben der Verwendung der Methoden zur globalen Nutzung der Pheromoninformation wurde ein Phasenmodell für ACO-Algorithmen vorgestellt. Die phasenabhängigen unterschiedlichen Ziele (z.B. Auswahl von Algorithmus-Varianten oder Konvergenz des Algorithmus) erfordern entsprechende Anpassungen. Eine auf die jeweilige Problem Instanz angepasste Entscheidungsreihenfolge (vgl. Kapitel 4), die in einer ersten Phase des ACO-Algorithmus festgelegt wird, führte zu deutlich besseren Resultaten. Weitere Verbesserungen konnten durch einen sinkenden Einfluss der Heuristik in den ersten Phasen und eine Erhöhung der Verdunstungsrate am Ende eines Laufs erzielt werden. Die einzelnen Entwurfsentscheidungen wurden anhand zahlreicher Experimente untersucht und gerechtfertigt. Eine Entropieanalyse zeigte, ähnlich wie die Visualisierung der Pheromonmatrizen bei einfachen Problemen in den vorherigen Kapiteln, dass eine gleichmäßige Konvergenz der Pheromonmatrix entscheidend für ein gutes Optimierungsverhalten ist. ACO-RCPSp wurde mit zahlreichen anderen zufallsbasierten Verfahren für RCPSp verglichen. Hierfür wurde der Datensatz mit den größten Problemen der Benchmark-Bibliothek PSPLIB verwendet. Unter der Voraussetzung, dass

alle Algorithmen die gleiche Anzahl an Projektplänen erzeugen und auswerten dürfen, wurde mit ACO-RCPSPP das beste Ergebnis erzielt. Gibt man diese Beschränkung auf, konnten mit ACO-RCPSPP über 150 neue beste Lösungen in der Bibliothek gefunden werden. Die Tatsache, dass der Algorithmus sowohl mit als auch ohne Einschränkung der Rechenzeit sehr gute Ergebnisse liefert, zeigt die Flexibilität und das hohe Potenzial des Ansatzes.

Kapitel 6

Bikriterielle Optimierung

Häufig müssen in der Praxis vorkommende Optimierungsprobleme bezüglich mehrerer schwer vereinbarer Zielkriterien optimiert werden. Bisher haben wir in dieser Arbeit nur einkriterielle Probleme betrachtet, d.h. einer Lösungspermutation wurde genau ein Zielfunktionswert zugeordnet. In diesem Kapitel beschäftigen wir uns mit Optimierungsproblemen, die bezüglich mehrerer konkurrierender Kriterien optimiert werden, d.h. die Verbesserung der Qualität des Zielfunktionswerts eines Kriteriums wird im Allgemeinen eine Verschlechterung der Qualität eines anderen Kriteriums zur Folge haben. Die in Kapitel 3 vorgestellten Methoden zur globalen Pheromonauswertung werden in diesem Kapitel ein Baustein für Methoden zur bikriteriellen Optimierung mit ACO-Algorithmen sein. In Abschnitt 6.1 werden wir zunächst einen Überblick über verwandte Arbeiten geben und notwendige Definitionen einführen. In Abschnitt 6.2 werden wir die ungewichtete Variante des im Abschnitt 3.1.3 untersuchten Planungsproblems um ein weiteres Optimierungskriterium erweitern und in Abschnitt 6.3 Methoden für die bikriterielle Optimierung mit ACO-Algorithmus vorstellen. Dabei werden mehrere Ameisenkolonien mit mehreren Pheromonmatrizen unterschiedlicher Kodierung Lösungen erzeugen. Da teilweise bis zu 20 Pheromonmatrizen während eines Optimierungslaufs verwendet werden, treffen wir die qualitativen Aussagen der Methoden nicht aufgrund einfacher Probleminstanzen im Sinne von Kapitel 2, sondern auf dem in Abschnitt 6.2 eingeführten bikriteriellen Optimierungsproblem. Teile der Ergebnisse dieses Kapitels sind in (Merkle und Middendorf, 2001a) und (Iredi et al., 2001) erschienen.

6.1 Motivation und Definitionen

Im Bereich der multikriteriellen Optimierung mit evolutionären Algorithmen wurden inzwischen zahlreiche Methoden entwickelt. Ein Überblick ist z.B. in (Deb, 2001), (Branke, 2002) oder (Veldhuizen und Lamont, 2000) gegeben. Arbeiten über die multikriterielle Optimierung mit Ameisenalgorithmen erschienen bisher nur sehr wenige. In (Mariano und Morales, 1999) wurde ein Ameisenalgorithmus zur multikriteriellen Optimierung vorgeschlagen, bei dem mehreren Kolonien je ein Optimierungskriterium zugeordnet ist, wobei die Kriterien der Wichtigkeit nach geordnet sind. Eine Ameise k der Kolonie i erhält jeweils die partielle Lösung der Ameise k der Kolonie $i - 1$ und versucht, diese Lösung hinsichtlich der Kriterien der Kolonie i zu verbessern. Nach dem sequentiellen Durchlaufen aller Kolonien aktualisieren die nicht-dominierten Lösungen (siehe Definition auf Seite 113) die Pheromoninformationen. In (Gambardella et al., 1999a) wurde ein Ameisenalgorithmus für ein Vehicle Routing Problem mit zwei Kriterien vorgestellt, bei dem zwei Ameisenkolonien jeweils ein Kriterium zugeordnet ist. Dabei wurde angenommen, dass das erste Optimierungskriterium (Anzahl der Fahrzeuge) wichtiger als das zweite Kriterium (Gesamtfahrzeit) ist. Die erste Kolonie versucht eine möglichst gute zulässige Lösung zu finden, die genau ein Fahrzeug weniger als die bisher beste Lösung benötigt. Die zweite Kolonie versucht, die bisher beste Lösung bezüglich des zweiten Kriteriums zu verbessern. Sobald die erste Kolonie eine neue Lösung findet, beginnen beide Kolonien mit reinitialisierten Pheromonwerten erneut mit der neuen besten Lösung. In (Gagné et al., 2000) wurde ein Ansatz vorgestellt, bei dem ein SMTTP (single machine total tardiness problem) mit Umrüstkosten und zwei weiteren Kriterien betrachtet wurde. Der Ansatz geht davon aus, dass die Minimierung der Umrüstkosten das wichtigste Optimierungskriterium ist. Die grundsätzliche Idee ist, bei der Berechnung der Auswahlwahrscheinlichkeiten beim Heuristikanteil alle Kriterien in Betracht zu ziehen. Alle bisherigen Ansätze gehen davon aus, dass die Kriterien bezüglich ihrer Wichtigkeit geordnet werden können. Bei Ansätzen mit mehreren Kolonien wurde jeweils eine Kolonie für ein Kriterium verwendet.

Bevor wir ACO-Algorithmen für bikriterielle Optimierungsprobleme vorstellen, benötigen wir einige Definitionen. Gegeben sei ein Permutationsproblem mit k Optimierungskriterien. Es sei $c_i(\pi)$, $i \in [1, k]$ die Funktion, die für das i -te Optimierungskriterium die Kosten der Lösungspermutation π angibt. Wie in den Kapiteln zuvor beschränken wir uns o.B.d.A. auf Minimierungsprobleme. Wir definieren:

- Eine Lösung π_1 *dominiert* eine Lösung π_2 (*schwach*), falls beide der folgenden Kriterien erfüllt sind:
 - $\forall i \in [1, k] : c_i(\pi_1) \leq c_i(\pi_2)$, d.h. für jedes Kriterium gilt, dass die Kosten der Lösung π_1 nicht höher als die Kosten der Lösung π_2 sind.
 - $\exists i \in [1, k] : c_i(\pi_1) < c_i(\pi_2)$, d.h. für mindestens ein Kriterium gilt, dass die Kosten bezüglich dieses Kriteriums für die Lösung π_1 echt kleiner sind als die entsprechenden Kosten der Lösung π_2 .
- Eine Lösung π_1 *dominiert* eine Lösung π_2 *stark*, falls gilt:
 - $\forall i \in [1, k] : c_i(\pi_1) < c_i(\pi_2)$, d.h. für jedes Kriterium sind die Kosten der Lösung π_1 echt kleiner als die Kosten der Lösung π_2 .
- Neben der Dominanz von Lösungen im Raum der Lösungspermutationen verwenden wir auf offensichtliche Weise auch den Begriff der Dominanz von Lösungen im Raum der Zielfunktionswerte. Gegeben sei eine Menge von nicht-dominierten Lösungen $\{\pi_1, \dots, \pi_m\}$. Unter der *nicht-dominierten Front* dieser Menge verstehen wir die Menge aller Punkte (x_1, \dots, x_k) im k -dimensionalen Raum der Zielfunktionswerte, die schwach, aber nicht stark, dominiert werden. Da ein Punkt $(c_1(\pi_i), \dots, c_k(\pi_i))$, $i \in [1, m]$ von sich selbst jedoch nicht schwach dominiert wird, seien diese Punkte auch in der nicht-dominierten Front enthalten.
- Eine Lösung π heißt *Pareto-optimal*, wenn es keine andere Lösung im Suchraum gibt, die π dominiert. Die Pareto-optimale Menge im Raum der Zielfunktionswerte ist die Menge aller Zielfunktionswerte von Pareto-optimalen Lösungen. Die *Pareto-optimale Front* ist die nicht-dominierte Front der Menge aller Pareto-optimalen Lösungen.
- Es seien n_f viele nicht-dominierte Fronten gegeben, wobei n_f ungerade sei. Des Weiteren sei für alle Lösungen der n_f nicht-dominierten Fronten o.B.d.A. jeder Zielfunktionswert jedes Kriteriums echt größer als 0. Die *Medianfront* (vgl. Fonseca und Fleming, 1996) dieser Fronten ergibt sich dann wie folgt. Zieht man eine Linie ausgehend vom Ursprung $(0, \dots, 0)$ durch alle nicht-dominierten Fronten, dann ergeben sich n_f Schnittpunkte. Sortiert man die Schnittpunkte entsprechend ihres Abstands zum Ursprung, dann ist der $\frac{n_f+1}{2}$ -te Schnittpunkt Element der Medianfront. Ein Beispiel für $n_f = 5$ und zwei Optimierungskriterien ist in Abbildung 6.1 gegeben.

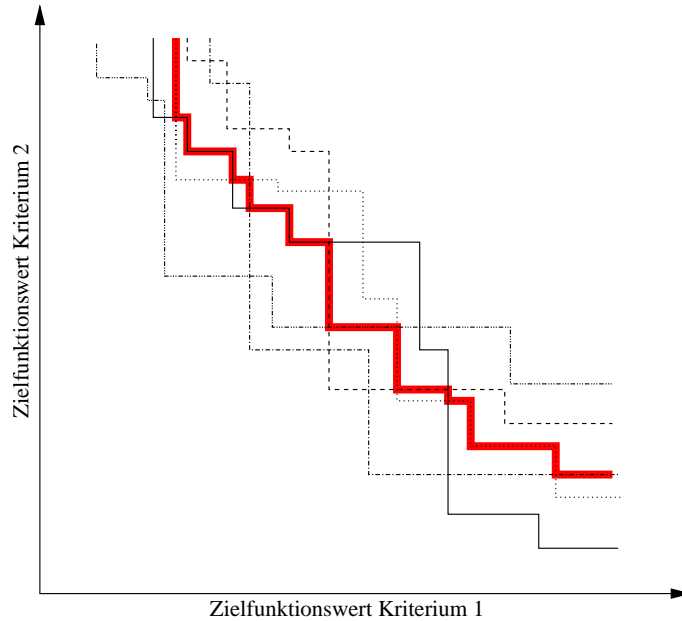


Abbildung 6.1: Medianfront für 5 nicht-dominierte Fronten und zwei Optimierungskriterien; die Medianfront ist fett eingezeichnet

Wir werden ACO-Algorithmen für bikriterielle Optimierungsprobleme vorstellen, bei denen, im Gegensatz zu den in der Literatur vorgestellten Verfahren, die einzelnen Kriterien nicht hinsichtlich ihrer Wichtigkeit sortiert werden müssen. Prinzipiell sind die in diesem Kapitel vorgestellten Verfahren auch für Optimierungsprobleme mit mehr als zwei Kriterien geeignet. Wir beschränken uns hier jedoch auf Probleme mit zwei Optimierungskriterien. Das Ziel ist es, unterschiedliche Lösungen der Pareto-optimalen Front zu finden. Hierfür wird ein Ansatz mit mehreren Kolonien verfolgt, bei dem die einzelnen Kolonien in unterschiedlichen Gebieten der nicht-dominierten Front nach neuen Lösungen suchen. Ansätze mit mehreren Kolonien wurden im Wesentlichen bei Parallelisierungen von ACO-Algorithmen (z.B. in Middendorf et al. 2002) betrachtet. Wir verwenden heterogene Kolonien, bei denen die einzelnen Ameisen einer Kolonie die zwei Kriterien hinsichtlich ihrer Wichtigkeit unterschiedlich gewichten, damit unterschiedliche Lösungen in der Pareto-optimalen Front gefunden werden können. Die Kooperation der einzelnen Kolonien untereinander wird realisiert, indem Ameisen die Aktualisierung der Pheromonwerte evtl. auf den Pheromonmatrizen anderer Kolonien durchführen, d.h. auf Pheromonmatrizen, die nicht zur Konstruktion der Lösungspermutation der Ameise verwendet wurden.

6.2 SMTTP mit Umrüstkosten

Das hier untersuchte Optimierungsproblem ist eine Erweiterung des in Abschnitt 3.1.3 vorgestellten Ein-Maschinen-Planungsproblems. Wir betrachten das SMTTP (single machine total tardiness problem) mit Umrüstkosten. Das Problem ist wie folgt definiert.

- Gegeben seien n Jobs, wobei Job j , $j \in [1, n]$ eine Laufzeit p_j und einen Fälligkeitstermin d_j hat. Für jedes Paar von Jobs i, j mit $i \neq j$ seien Umrüstkosten $c(i, j) > 0$ gegeben, die anfallen, wenn Job j direkter Nachfolger des Jobs i im erzeugten Plan ist.
- Gesucht ist ein unterbrechungsfreier Ein-Maschinen-Belegungsplan, der sowohl die Summe der Verspätungen $T = \sum_{j=1}^n \max\{0, C_j - d_j\}$ (C_j ist der Fertigstellungszeitpunkt des Jobs j) als auch die Umrüstkosten $C = \sum_{i=1}^{n-1} c(j_i, j_{i+1})$ minimiert, wobei die Permutation (j_1, j_2, \dots, j_n) den Belegungsplan der Maschine angibt.

SMTTP ist NP-hart im schwachen Sinne (vgl. Kapitel 3.1.3 und (Du und Leung, 1990)). Ein ACO-Algorithmus für die gewichtete Variante von SMTTP wurde in Abschnitt 3.1.3 vorgestellt. Dabei wurden mit einer Kombination aus lokaler Pheromonauswertung und Summen-Pheromonauswertung die besten Ergebnisse erzielt.

Das Problem, einen Belegungsplan mit minimalen Umrüstkosten zu finden, ist äquivalent zur asymmetrischen Variante des Problems des kürzesten Hamiltonschen Pfades, welches NP-vollständig im strengen Sinne ist. Ein ACO-Algorithmus, bei dem nur die Umrüstkosten minimiert werden sollen, verwendet im Gegensatz zu einem ACO-Algorithmus für SMTTP eine unterschiedliche Kodierung der Pheromonmatrix. Während für SMTTP die Platzierung eines Jobs im Belegungsplan wichtig ist, muss für die Minimierung der Umrüstkosten betrachtet werden, welcher Job direkt nach welchem anderen Job geplant wird. Daher wird eine Kodierung der Pheromonmatrix verwendet, bei der sowohl die Zeilen als auch die Spalten einzelnen Jobs entsprechen. Wir wählen für die Minimierung der Summe der Verspätungen wie üblich eine Pheromonmatrix mit Platz-Job Kodierung (d.h., die Zeilen entsprechen Plätzen der Lösungspermutation und die Spalten entsprechen unterschiedlichen Jobs) und für die Minimierung der Umrüstkosten eine Pheromonmatrix mit Job-Job Kodierung. Der Pheromonwert τ_{ij} gibt bei einer Matrix mit Job-Job Kodierung an, wie wünschenswert es ist, dass Job j im Belegungsplan (also

in der Lösungspermutation) nach Job i kommt. Zusätzlich führen wir einen Pseudo-Job mit Index 0 ein und definieren die Umrüstkosten $c(0, j) = 0$ für alle Jobs $j \in [0, n]$. Der Pheromonwert τ_{0j} gibt an, wie wünschenswert es ist, Job j als ersten Job im Belegungsplan zu platzieren. Die Heuristik, die für die Minimierung der Umrüstkosten verwendet wird, ist $\eta_{ij} = 1/c(i, j)$, falls $i \in [1, n]$, und $\eta_{0j} = 1$ sonst.

6.3 Ameisenalgorithmen für bikriterielle Optimierungsprobleme

In diesem Abschnitt führen wir ACO-Algorithmen für bikriterielle Optimierungsprobleme ein. In Abschnitt 6.3.1 werden wir einen Ansatz mit nur einer Kolonie vorstellen. Diesen Ansatz werden wir im Abschnitt 6.3.2 erweitern und mehrere Kolonien zur Optimierung verwenden.

6.3.1 Eine Kolonie

Zwei Pheromonmatrizen

Unterschiedliche Optimierungskriterien benötigen evtl. Pheromonmatrizen mit unterschiedlicher Kodierung (vgl. Abschnitt 6.2). Daher werden wir für jedes Kriterium eine eigene Pheromonmatrix verwenden. Eine Kolonie von Ameisen verwendet zur Berechnung der Auswahlwahrscheinlichkeiten des nächsten Jobs immer diese zwei Pheromonmatrizen.

Für das SMTTP mit Umrüstkosten verwenden wir eine Pheromonmatrix $[\tau_{ij}]$ für die Minimierung der Verspätungen und $[\tau'_{ij}]$ für die Minimierung der Umrüstkosten. Dabei ist die Kodierung der beiden Matrizen unterschiedlich und auf das Optimierungskriterium angepasst (vgl. Abschnitt 6.2).

Heterogene Kolonien

Um die Ameisen in unterschiedlichen Gebieten der nicht-dominierten Front nach guten Lösungen suchen zu lassen, gewichtet jede der m Ameisen einer Kolonie die einzelnen Optimierungskriterien unterschiedlich. Bei unserem bikriteriellen Optimierungsproblem verwendet Ameise k den Parameter $\lambda_k = \frac{k-1}{m-1}$ und berechnet die Auswahlwahrscheinlichkeiten wie folgt (der Übersicht wegen sei $\lambda = \lambda_k$).

$$p_{ij} = \frac{(\tau_{ij}^*)^{\lambda\alpha} \cdot (\tau'_{ij})^{(1-\lambda)\alpha} \cdot \eta_{ij}^{\lambda\beta} \cdot (\eta'_{ij})^{(1-\lambda)\beta}}{\sum_{h \in S} (\tau_{ih}^*)^{\lambda\alpha} \cdot (\tau'_{ih})^{(1-\lambda)\alpha} \cdot \eta_{ih}^{\lambda\beta} \cdot (\eta'_{ih})^{(1-\lambda)\beta}} \quad (6.1)$$

Hierbei ist η_{ij} die Heuristikinformation für das Kriterium Minimierung der Verspätungen (vgl. Gleichung (3.5) in Abschnitt 3.1.3 auf Seite 35) und η'_{ij} die Heuristikinformation für das Kriterium Minimierung der Umrüstkosten (vgl. Abschnitt 6.2). In Abschnitt 3.1 haben wir gesehen, dass die gewichtete Summen-Pheromonauswertung in Kombination mit der lokalen Pheromonauswertung für SMMTP die besten Ergebnisse erzielt hat. Die Parameter c und γ (Gewichtung der beiden Pheromonauswertungsmethoden und Gewichtung der Pheromonwerte früherer Entscheidungen), mit denen die besten Ergebnisse erzielt wurden, werden auch in diesem Kapitel für die Messungen verwendet. Die Pheromonwerte τ_{ij}^* in Gleichung (6.1) werden somit entsprechend Gleichung (3.3) auf Seite 27 berechnet.

Man beachte, dass durch die Verwendung unterschiedlicher λ_k die Ameise m mit $\lambda_m = 1$ nur hinsichtlich des ersten Optimierungskriteriums, und Ameise 1 mit $\lambda_1 = 0$ nur hinsichtlich des zweiten Kriteriums Lösungen erzeugt.

Aktualisierung

Wenn alle m Ameisen einer Generation eine Lösung gefunden haben, dann muss entschieden werden, welche der Ameisen die Pheromonmatrizen aktualisieren dürfen. Bei dem hier vorgeschlagenen Ansatz darf jede Ameise in der nicht-dominierten Front der aktuellen Generation eine Aktualisierung durchführen. Dabei verändert die Ameise beide Pheromonmatrizen $[\tau_{ij}]$ und $[\tau'_{ij}]$. Man beachte, dass diese Regel voraussetzt, dass nicht zu wenige Ameisen in einer Kolonie sind. Wenn nur wenige Ameisen in einer Kolonie sind, dann unterscheiden sich die λ -Werte der einzelnen Ameisen stark, und es findet keine Konkurrenz unter den Ameisen statt, da jede Ameise in einem unterschiedlichen Gebiet der nicht-dominierten Front nach Verbesserungen sucht. Wenn das der Fall wäre, dann würden auch Ameisen mit schlechter Qualität die Pheromonmatrizen aktualisieren.

Um jeder Generation von Ameisen den gleichen Einfluss zu geben, ist die Pheromonmenge, die eine Generation von Ameisen auf die Pheromonmatrix addiert, in jeder Generation gleich. Eine Ameise, welche die Pheromonmatrix aktualisieren darf, rechnet $\tau_{ij} = \tau_{ij} + 1/l$ für jedes Paar (i, j) der entsprechenden Lösung. Hierbei ist l die Anzahl der Ameisen, die in der aktuellen Generation eine Aktualisierung durchführen dürfen.

6.3.2 Mehrere Kolonien

Hier erweitern wir den bisher beschriebenen ACO-Algorithmus für das bikriterielle Optimierungsproblem. Anstelle einer Kolonie werden mehrere Kolonien miteinander kooperieren und versuchen, sich auf unterschiedliche Gebiete der Pareto-optimalen Front zu konzentrieren. Jede der p Kolonien wird m/p Ameisen haben.

Aktualisierung

Bei dem in Abschnitt 6.3.1 vorgestellten Ansatz mit nur einer Kolonie durften nur die Ameisen in der nicht-dominierten Front die Pheromonmatrix aktualisieren. Bei einem ACO-Algorithmus mit mehreren Kolonien muss man bei der Aktualisierung der Pheromonmatrizen anders vorgehen. Die einfachste Methode ist, dass jeweils diejenigen Ameisen die zwei Pheromonmatrizen ihrer Kolonie aktualisieren dürfen, welche in der nicht-dominierten Front der jeweiligen Kolonie sind. Bei dieser Vorgehensweise findet keine Kooperation zwischen den Kolonien statt. Jede Kolonie sucht somit unabhängig von den jeweils anderen Kolonien nach der Pareto-optimalen Front. Daher sind die Ergebnisse eines Messlaufs, bei dem diese Methode angewendet wird, identisch zu den Ergebnissen, die man erhält, wenn ein ACO-Algorithmus mit nur einer Kolonie mehrfach neu gestartet wird, und am Ende die nicht-dominierte Front aller zugehörigen Lösungen berechnet wird (Multi-Start-Strategie). Im Folgenden stellen wir Methoden vor, bei denen die Ameisenkolonien kooperieren. Hierfür werden alle Lösungspermutationen aller Kolonien in der Menge der aktuellen Lösungen gespeichert. Diese wird verwendet, um die nicht-dominierte Front der aktuellen Lösungen aller Kolonien einer Generation zu berechnen. Wir bezeichnen diese Front mit *globale nicht-dominierte Front* (die *lokale nicht-dominierte Front* ist entsprechend die nicht-dominierte Front der Lösungen einer einzelnen Kolonie). Nur solche Ameisen, die eine Lösung in dieser globalen nicht-dominierten Front gefunden haben, dürfen die Pheromonmatrizen aktualisieren. Es stellt sich die Frage, in welcher Kolonie die jeweilige Ameise die Aktualisierung durchführt. Wir werden zwei unterschiedliche Methoden hierfür untersuchen.

1. *Pheromonaktualisierung entsprechend dem Ursprung*: Eine Ameise aktualisiert nur genau die Pheromonmatrizen, welche von der zugehörigen Kolonie zur Lösungskonstruktion verwendet wurden (vgl. Abbildung 6.2).

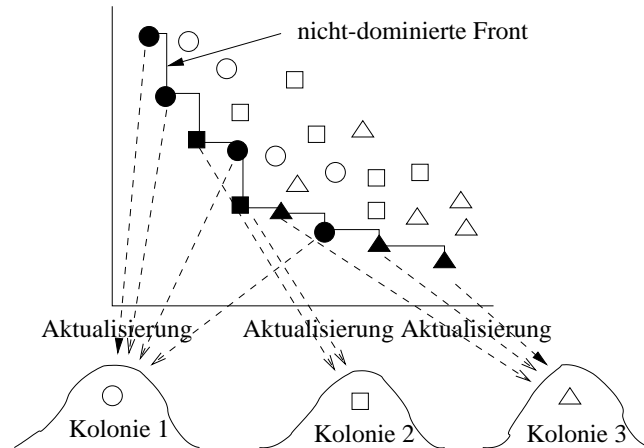


Abbildung 6.2: Aktualisierung entsprechend dem Ursprung: Jede Ameise mit einer Lösung in der globalen nicht-dominierten Front aktualisiert genau die Pheromonmatrizen, welche von der entsprechenden Kolonie zur Lösungskonstruktion verwendet wurde.

2. *Pheromonaktualisierung entsprechend dem Gebiet in der nicht-dominierten Front:* Die Folge der Lösungen entlang der nicht-dominierten Front wird in p Teile gleicher Größe eingeteilt. Ameisen, die im i -ten Teil der nicht-dominierten Front eine Lösung gefunden haben, aktualisieren die Pheromonmatrizen der i -ten Kolonie (vgl. Abbildung 6.3). Formal gehen wir wie folgt vor. Die Lösungen der nicht-dominierten Front werden entsprechend dem ersten Optimierungskriterium sortiert (es ist nicht relevant, hinsichtlich welchen Kriteriums die Liste sortiert wird). Es sei L diese sortierte Liste. Die Liste wird so in Teile L_1, L_2, \dots, L_p eingeteilt, dass sich deren Größe paarweise um maximal 1 unterscheidet. Alle Ameisen, die Lösungen der Liste L_i , $i \in [1, p]$ gefunden haben, aktualisieren die Pheromonmatrizen der Kolonie i .

Durch die beiden Methoden unterliegen die Ameisen einem sehr viel stärkeren Selektionsdruck bezogen auf die Menge der Ameisen, die eine Aktualisierung durchführen dürfen. Für eine Ameise ist es nicht hinreichend, eine Lösung in der lokalen nicht-dominierten Front zu finden. Die Lösung muss in der globalen nicht-dominierten Front liegen. Dies könnte von Vorteil sein, weil im Zusammenspiel mit anderen Kolonien festgestellt werden kann, welche der Lösungen der lokalen nicht-dominierten Front schlecht sind. Eine interessante Beobachtung ist, dass bei der Aktualisierung entsprechend dem Ursprung

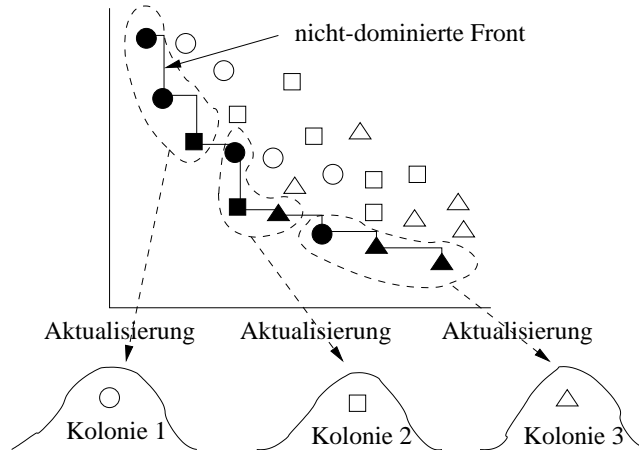


Abbildung 6.3: Aktualisierung entsprechend dem Gebiet in der nicht-dominierten Front: Ameisen mit einer Lösung in der globalen nicht-dominierten Front aktualisieren die Pheromonmatrizen der Kolonie, zu der das entsprechende Gebiet der globalen nicht-dominierten Front gehört.

die Kolonien automatisch in unterschiedlichen Gebieten der nicht-dominierten Front nach Verbesserungen suchen. Denn es ist wahrscheinlicher, dass eine Lösung einer lokalen nicht-dominierten Front auch Lösung der globalen nicht-dominierten Front ist, wenn sich nur wenige Lösungen anderer Kolonien im gleichen Gebiet der globalen nicht-dominierten Front befinden. Daher ist es wahrscheinlicher, dass die Ameisen die Pheromonmatrizen aktualisieren (und somit den darauf folgenden Suchprozess beeinflussen), die Lösungen erzeugt haben, die in ein Gebiet der nicht-dominierten Front fallen, in der ansonsten wenig andere Lösungen gefunden wurden.

Mit der Aktualisierung entsprechend dem Gebiet der nicht-dominierten Front werden wir die Suche in unterschiedlichen Gebieten der nicht-dominierten Front der einzelnen Kolonien explizit verstärken.

Heterogene Kolonien

Analog zum ACO-Algorithmus mit nur einer Kolonie verwenden alle Ameisen einer Kolonie unterschiedliche λ -Werte, um den relativen Einfluss der Optimierungskriterien zu steuern. Wir verwenden unterschiedliche Regeln, mit denen wir die λ -Werte der einzelnen Kolonien berechnen. Bei der ersten Regel (Regel 1) werden wir, wie beim Ansatz mit nur einer Kolonie, unabhängig von der Nummer der Kolonie möglichst unterschiedliche Gewichte $\lambda \in [0, 1]$ wählen.

Bei den Regeln 2 und 3 ordnen wir den einzelnen Kolonien Teilintervalle aus $[0, 1]$ zu, damit die einzelnen Kolonien die Optimierungskriterien unterschiedlich stark gewichten. Die Regel 2 und 3 unterscheiden sich hinsichtlich der Tatsache, dass die gewählten Teilintervalle der einzelnen Kolonien disjunkt sind (Regel 2), oder ob sie sich überlappen (Regel 3). Formal schreiben wir

Regel 1: Ameise k , $k \in [1, m/p]$, der Kolonie i , $i \in [1, p]$, verwendet $\lambda_k = \frac{k-1}{m/p-1}$.

Regel 2: (disjunkte λ -Intervalle): Ameise k , $k \in [1, m/p]$ der Kolonie i , $i \in [1, p]$, verwendet $\lambda_k = (i-1) \cdot \frac{m}{p} + k$.

Regel 3: (überlappende λ -Intervalle): Das λ -Intervall der Kolonie i , $i \in [1, p]$ überlappt zu 50% mit den λ -Intervallen der Kolonien $i-1$ und $i+1$. Eine Ameise der Kolonie i hat λ -Werte, die im Intervall $[\frac{i-1}{p+1}, \frac{i+1}{p+1}]$ äquidistant angeordnet sind (vgl. Abbildung 6.4).

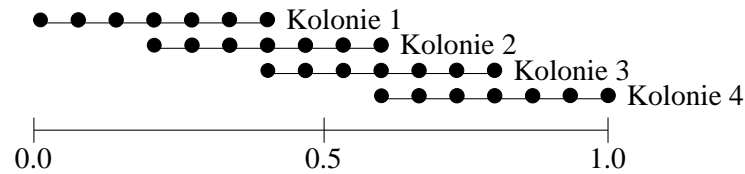


Abbildung 6.4: λ -Werte bei Verwendung der Regel 3 bei 4 Kolonien und 7 Ameisen pro Kolonie

6.4 Probleminstanzen und Parameter

Die ACO-Algorithmen für bikriterielle Optimierung wurden auf Instanzen getestet, welche bis auf die Umrüstkosten auf die gleiche Weise wie die Instanzen für die Untersuchungen der Summen-Pheromonauswertung bei dem NP-vollständigen Problem SMTWTP erzeugt wurden (vgl. Abschnitt 3.1.3 ab Seite 35). Jeder Job hat eine ganzzahlige Laufzeit, die zufällig gleichverteilt aus dem Intervall $[1, 100]$ gewählt wurde. Die ganzzahligen Fälligkeitstermine wurden gleichverteilt aus dem Intervall gewählt, welches in Gleichung (3.6) auf Seite 36 angegeben ist. Dabei wurde $RDD = 0.6$ gesetzt, und TF aus der Menge $\{0.4, 0.5, 0.6\}$ gewählt. Die Umrüstkosten wurden zufällig gleichverteilt entweder aus dem Intervall $[1, 100]$ oder $[50, 100]$ gewählt.

Die Parameter für alle Testläufe waren $\alpha = 1$, $\beta = 1$ und $\rho = 0.02$. Für das Kriterium der Minimierung der Verspätungen wurde die gewichtete Summen-Pheromonauswertung in Kombination mit der lokalen Pheromonauswertung

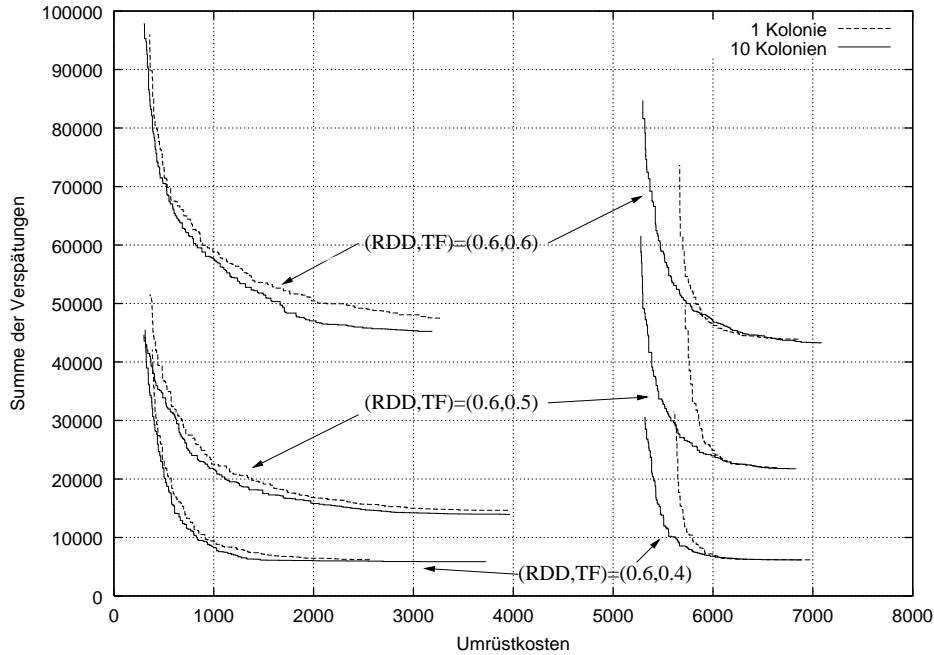


Abbildung 6.5: Medianfront der globalen nicht-dominierten Fronten mit einer und 10 Kolonien; links: 3 Instanzen mit Umrüstkosten $\in [1, 100]$; rechts: 3 Instanzen mit Umrüstkosten $\in [50, 100]$

verwendet. Die gewählten Parameter waren $c = 0.6$ und $\gamma = 0.9$. Die Anzahl der Ameisen pro Generation war $m = 100$. Wenn mehrere Kolonien verwendet wurden, dann wurden die Ameisen gleichmäßig auf alle Kolonien verteilt. Jeder Testlauf wurde 11 mal wiederholt, die in Abschnitt 6.5 angegebenen Medianfronten wurden jeweils mit den nicht-dominierten Fronten dieser 11 Läufe berechnet. Jeder Lauf wurde nach 300 Generationen gestoppt.

6.5 Ergebnisse

Das Verhalten des ACO-Algorithmus mit mehreren Kolonien wurde auf 6 Probleminstanzen untersucht: 3 mit Umrüstkosten aus dem Intervall $[1, 100]$ und 3 mit Umrüstkosten aus dem Intervall $[50, 100]$ (und jeweils $TF \in \{0.4, 0.5, 0.6\}$). Das Ergebnis eines Laufs ist die Teilmenge aller nicht-dominierten Lösungen in der Menge aller Lösungen, die während eines Laufs erzeugt wurden. Die Medianfront der globalen nicht-dominierten Fronten für Läufe mit einer oder 10 Kolonien sind in Abbildung 6.5 dargestellt. Für die Messungen mit

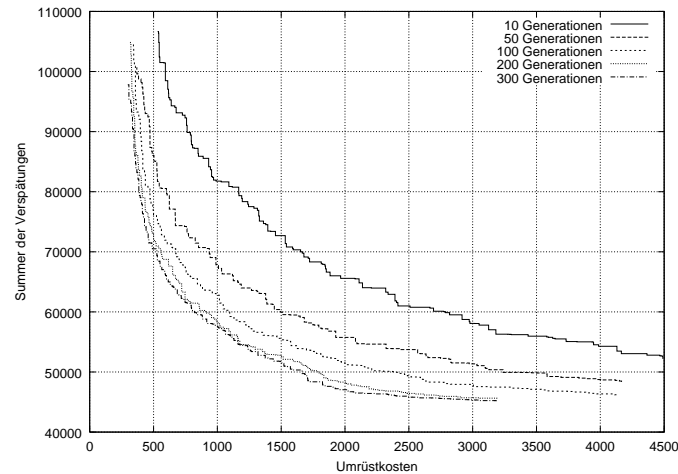


Abbildung 6.6: Konvergenzverhalten (10 Kolonien); dargestellt sind die Medianfronten nach 10, 50, 100, 200 und 300 Generationen

10 Kolonien wurde die Methode Aktualisierung entsprechend dem Ursprung verwendet. Für die Wahl der Parameter λ wurde die Regel 3 (überlappende λ -Intervalle) verwendet. Die Abbildung zeigt, dass die Medianfronten der Läufe mit 10 Kolonien nahezu immer besser sind als diejenigen, welche mit nur einer Kolonie erzielt werden. Nur für Instanzen mit Umrüstkosten aus dem Intervall $[50, 100]$ ist die Medianfront des Ansatzes mit nur einer Kolonie in einem kleinen Gebiet geringfügig besser. Bei diesen Instanzen wurden mit 10 Kolonien Lösungen mit viel kleineren Umrüstkosten gefunden.

Im Folgenden vergleichen wir die unterschiedlichen λ -Regeln. Für die Testläufe verwenden wir die Instanz mit $TF = 0.6$ und Umrüstkosten aus dem Intervall $[1, 100]$.

Abbildung 6.6 zeigt das Konvergenzverhalten, falls 10 Kolonien verwendet werden. Dargestellt sind die Medianfronten zu unterschiedlichen Zeitpunkten. Zwischen Generation 200 und 300 unterscheiden sich die Medianfronten nicht mehr stark. Die Ergebnisse mit nur einer Kolonie sind bezüglich des Konvergenzverhaltens sehr ähnlich und daher nicht abgebildet.

Die Medianfronten, die mit unterschiedlichen Methoden der Aktualisierung erreicht wurden, sind in Abbildung 6.7 abgebildet. Die Multi-Start-Strategie ohne Kooperation der Kolonien ist entlang der gesamten Front am schlechtesten. Bei kleinen Umrüstkosten sind die beiden anderen Aktualisierungsmethoden etwa gleich gut. Mit der Pheromonaktualisierung entsprechend dem Gebiet in der nicht-dominierten Front wurde bei mittelgroßen oder großen Umrüstkosten die bessere Medianfront erzielt.

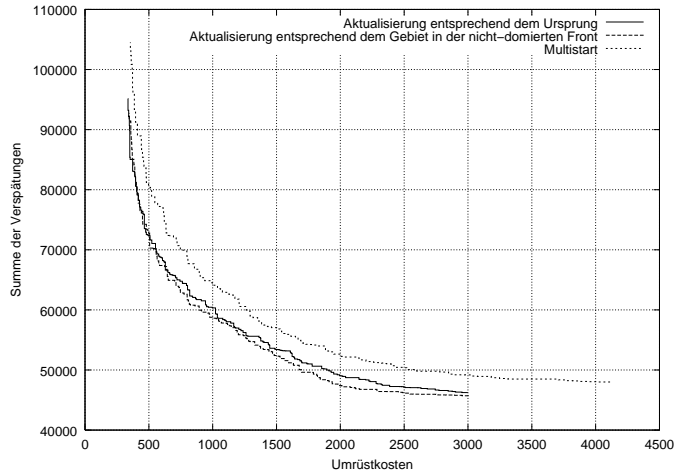


Abbildung 6.7: Einfluss der Aktualisierung (10 Kolonien); dargestellt sind die Medianfronten für: Aktualisierung entsprechend dem Ursprung, Aktualisierung entsprechend dem Gebiet in der nicht-dominierten Front und bei Verwendung der Multi-Start-Strategie

Abbildung 6.8 zeigt, von welchen Kolonien die Ameisen der nicht-dominierten Front nach 300 Generationen stammen (bzw. mit welcher Matrix oder streng genommen mit welchen beiden Matrizen die Ameise erzeugt wurde). Es ist deutlich zu erkennen, dass sich die Ameisen bei einer Aktualisierung entsprechend dem Gebiet in der nicht-dominierten Front mit steigendem Index der Kolonie immer stärker auf ein Gebiet in der nicht-dominierten Front spezialisieren, bei dem die Umrüstkosten wichtiger sind. Auch bei der Aktualisierung entsprechend dem Ursprung scheinen sich die Ameisen auf einzelne Gebiete in der nicht-dominierten Front zu spezialisieren. Hier kann, im Gegensatz zur Aktualisierung entsprechend dem Gebiet in der nicht-dominierten Front, natürlich nicht vorhergesagt werden, in welchem Gebiet der nicht-dominierten Front sich eine Kolonie spezialisieren wird.

Der Einfluss der λ -Regeln bei Verwendung der Aktualisierung entsprechend dem Ursprung ist in Abbildung 6.9 für 2 und 5 Kolonien dargestellt. Bei Verwendung der Regel 2 (disjunkte λ -Intervalle) zeigt der Messlauf mit nur 2 Kolonien deutlich, dass sich die Ameisen auf unterschiedlichen Gebieten der nicht-dominierten Front spezialisieren müssen. Die Ergebnisse im Zentrum dieses Gebiets, auf das sich die Kolonie spezialisiert hat, sind bei Regel 2 besser als bei Regel 3. In der Mitte der nicht-dominierten Front und in den Extremgebieten sind die Ergebnisse der Regel 2 schlecht. Regel 3 zeigt ein viel ausgeglicheneres Verhalten. Regel 1 (alle Kolonien haben Ameisen mit

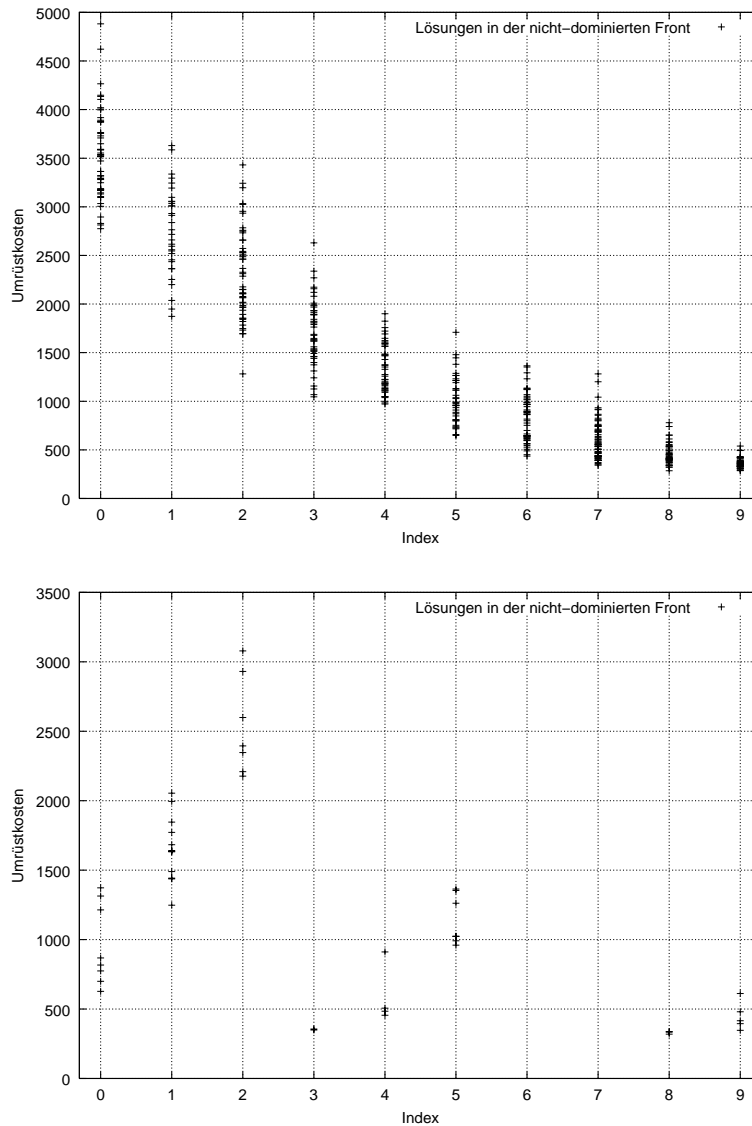


Abbildung 6.8: Lösungen in der nicht-dominierten Front nach 300 Generationen bei 10 Kolonien: abgebildet sind die Umrüstkosten und der Index der Kolonie, von der die Ameise stammt, die diese Umrüstkosten erreicht hat; oben: Aktualisierung entsprechend dem Gebiet in der nicht-dominierten Front (dargestellt sind die Lösungen aller 11 Läufe); unten: Aktualisierung entsprechend dem Ursprung (dargestellt sind die Lösungen eines Laufs, da sich die Kolonien mit gleichem Index je nach Messlauf in unterschiedlichen Gebieten der nicht-dominierten Front spezialisieren)

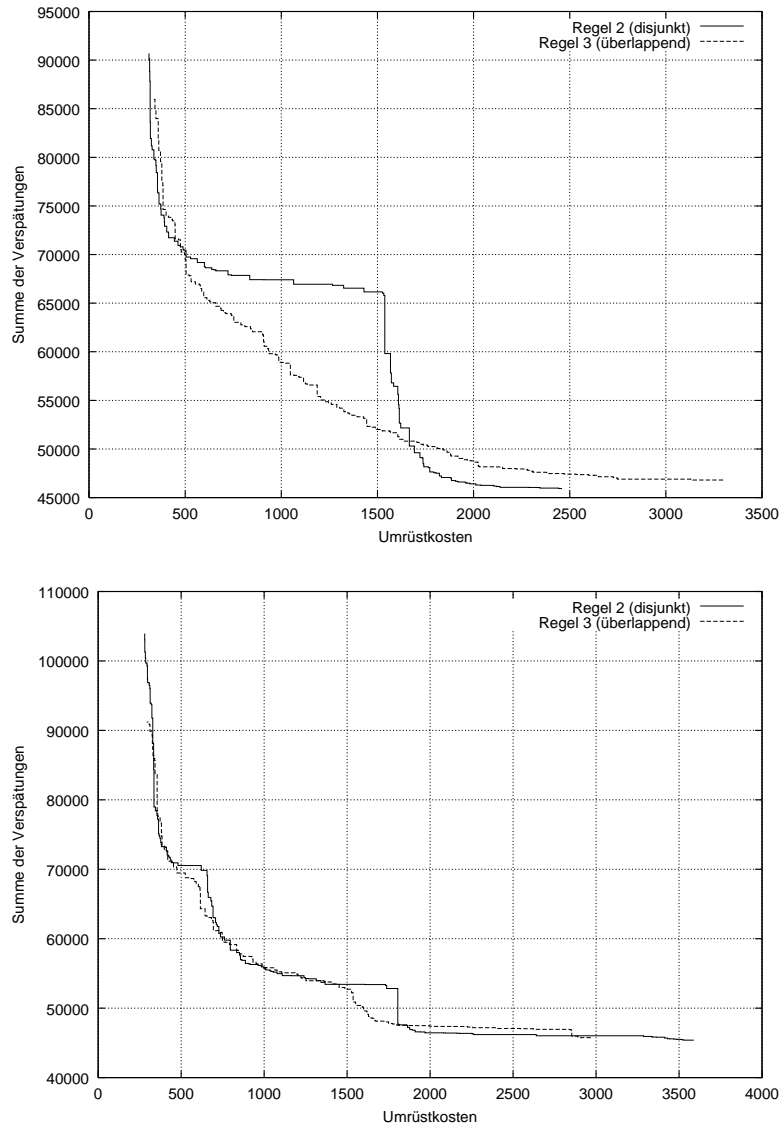


Abbildung 6.9: Einfluss der λ -Regel 2 (disjunkte Intervalle) und der λ -Regel 3 (überlappende Intervalle); oben: 2 Kolonien; unten: 5 Kolonien

λ -Werten im Intervall $[0, 1]$) lieferte entlang der gesamten Medianfront ein schlechteres Ergebnis (nicht dargestellt). Die Ergebnisse für 5 Kolonien sind ähnlich. Die Medianfronten der einzelnen Regeln unterscheiden sich nicht so sehr, da sich die Kolonien auf kleinere Gebiete spezialisieren. Bei 10 Kolonien waren nur noch sehr kleine Unterschiede zu erkennen (nicht dargestellt).

In Abschnitt 6.3.1 haben wir unterschiedliche Kodierungen für die beiden Pheromonmatrizen gewählt: eine Pheromonmatrix mit Platz-Job Kodierung

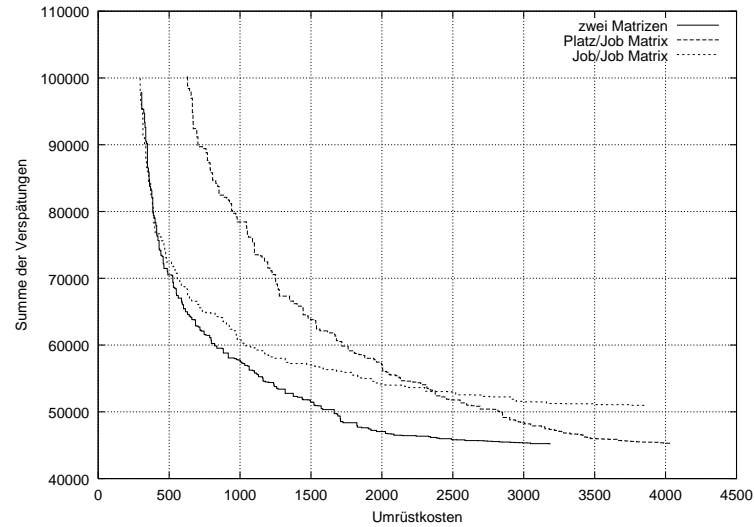


Abbildung 6.10: Einfluss der Matrixkodierung der Pheromonmatrizen einer Kolonie; dargestellt sind: Verwendung nur einer Pheromonmatrix in Platz-Job Kodierung, Verwendung nur einer Pheromonmatrix mit Job-Job Kodierung, Verwendung beider Matrizen. Die Testläufe wurden mit 10 Kolonien durchgeführt; Pheromonaktualisierung entsprechend dem Gebiet in der nicht-dominierten Front und überlappenden λ -Intervallen

für die Minimierung der Verspätungen und eine Pheromonmatrix mit Job-Job Kodierung für die Minimierung der Umrüstkosten. Anhand der folgenden Messläufe zeigen wir, dass die Verwendung zweier Matrizen mit unterschiedlichen Kodierungen zu besseren Ergebnissen führt. Wenn nur die Pheromonmatrix mit Platz-Job Kodierung verwendet wird, dann sind die Ergebnisse bezüglich der Verspätungskosten sehr gut, wenn die Umrüstkosten keine Rolle spielen. Jedoch konnten die Ameisen keine Lösungen mit geringen Umrüstkosten finden. Wenn wir nur die Pheromonmatrix mit Job-Job Kodierung verwenden, dann beobachten wir genau das Umgekehrte. Die Ameisen finden gute Lösungen hinsichtlich der Umrüstkosten, wenn die Verspätungen keine Rolle spielen, jedoch werden keine Lösungen mit geringen Verspätungskosten gefunden. Die Verwendung beider Matrizen pro Kolonie führt dazu, dass entlang der gesamten Medianfront die Ergebnisse besser sind als bei Verwendung nur einer Matrix.

6.6 Zusammenfassung

In diesem Kapitel haben wir Methoden vorgestellt, um bikriterielle Optimierungsprobleme mit ACO-Algorithmen zu lösen. Dabei hat sich gezeigt, dass eine geeignete Kooperation zwischen Ameisenkolonien, die in unterschiedlichen Gebieten der nicht-dominierten Front suchen, die besten Ergebnisse liefert. Die Kooperation der Kolonien wurde unter anderem durch unterschiedliche Methoden der Pheromonaktualisierung der Ameisen realisiert. Für die Optimierungskriterien wurden Pheromonmatrizen mit unterschiedlicher Kodierung verwendet, wobei für die Minimierung der Summe der Verspätungen des untersuchten bikriteriellen Optimierungsproblems die in Kapitel 3 eingeführte globale Pheromonauswertung verwendet wurde.

Kapitel 7

Modellierung

Ausgehend von Ergebnissen über die Verzerrung von Auswahlwahrscheinlichkeiten haben wir in dieser Arbeit unterschiedliche Methoden zur Verbesserung des Optimierungsverhaltens von ACO-Algorithmen eingeführt. Dazu wurden empirische Untersuchungen sowohl auf einfachen Problemen als auch auf NP-vollständigen Problemen durchgeführt. In den Kapiteln 5 und 6 wurden die entwickelten Verfahren dann erweitert, sodass sie auch auf großen Instanzen praxisrelevanter Probleme die besten bisherigen Verfahren im Optimierungsverhalten übertreffen. Umfangreiche Tests haben die Signifikanz der erzielten Verbesserung gezeigt. In diesem Kapitel werden wir ein tieferes Verständnis des Optimierungsverfahrens durch die Modellierung von ACO-Algorithmen erhalten. ACO-Algorithmen basieren auf Folgen von Zufallsentscheidungen. Da diese im Regelfall nicht unabhängig voneinander sind, ist eine theoretische Analyse des Verhaltens von ACO-Algorithmen schwierig. Außer den Konvergenzbeweisen für ACO-Algorithmen (Gutjahr, 2000, 2002; Stützle und Dorigo, 2000) (vgl. Kapitel 1) wurden bisher nahezu keine theoretischen Untersuchungen durchgeführt. In diesem Kapitel wird ein deterministisches Modell für ACO-Algorithmen vorgeschlagen, welches analytischen Methoden zugänglich ist. Für einfach strukturierte Optimierungsprobleme können wir mit Hilfe des Modells numerisch exakte und analytische Ergebnisse ableiten. Um zu zeigen, dass das Modellverhalten in vieler Hinsicht dem zufallsgesteuerten ACO-Algorithmus entspricht, werden die Resultate um empirische Studien ergänzt.

Modellbasierte Analysen werden oft in Bereichen eingesetzt, in denen das modellierte System nicht vollständig verstanden wird oder theoretische Analysen sehr schwierig sind. Zahlreiche Arbeiten sind hierzu für die Analyse genetischer Algorithmen (GAs) erschienen. Ein möglicher Ansatz dabei ist, anstatt einer endlichen eine unendlich große Population zu verwenden. Diese

sind im Modell meist einfacher zu analysieren, da sich viele Eigenschaften einer unendlich großen Population bei nur endlich vielen Zufallsentscheidungen nicht ändern (Vose und Liepins, 1991; Vose, 1999). Eine Population wird bei dieser Methode durch einen Vektor aus Wahrscheinlichkeiten von Individuen repräsentiert. Für jedes mögliche Individuum wird die Wahrscheinlichkeit angegeben, dass ein zufällig aus der Population gewähltes Individuum das entsprechende ist. Eine weitere Methode GAs zu modellieren charakterisiert eine Population durch wenige Parameter, die jedoch wichtige Eigenschaften einer Population abdecken. Beispiele für solche Parameter sind Durchschnitt und Varianz der Güteverteilung einer Population oder auch der prozentuale Anteil von Individuen einer bestimmten Güte. Diese Methode wird beispielsweise in den Arbeiten (Prügel-Bennett und Shapiro, 1997), (van Nimwegen et al., 1999) und (Prügel-Bennett und Rogers, 2000) verwendet, und ist durch Ideen der statistischen Mechanik motiviert. Größtenteils werden GAs auf Problemen modelliert, die sehr einfach strukturiert sind, aber dennoch charakteristische Eigenschaften komplexerer Problemen widerspiegeln. Ein Beispiel hierfür sind die „Royal Road“-Funktionen, bei denen die Anzahl der Einsen eines Genstrings auf gewisse Art und Weise maximiert werden muss (Mitchell und Forrest, 1997).

Das deterministische ACO-Modell, welches in diesem Kapitel vorgestellt wird, basiert auf (deterministisch berechneten) Wahrscheinlichkeiten einzelner Entscheidungen der Ameisen eines ACO-Algorithmus. Insbesondere werden wir die Dynamik der Pheromonwerte während eines Laufs eines ACO-Algorithmus untersuchen. Im vorgestellten Modell wird die Aktualisierung der Pheromoninformation durch den erwarteten Aktualisierungswert einer zufälligen Generation von Ameisen berechnet. Das bedeutet, dass nicht Zufallsentscheidungen einzelner Ameisen die Dynamik beeinflussen, sondern vielmehr das ermittelte Durchschnittsverhalten unendlich vieler Ameisen. Das Modellverhalten des Algorithmus entspricht einer Vorgehensweise, bei der unendlich viele Generationen eines ACO-Algorithmus unabhängig und basierend auf der gleichen Pheromoninformation durchgeführt werden. Die Pheromonaktualisierung wird mit dem jeweiligen Erwartungswert der Aktualisierung berechnet.

Das Kapitel gliedert sich wie folgt. Zunächst definieren wir in Abschnitt 7.1 die in den Untersuchungen verwendete Problemklasse. Das deterministische ACO-Modell wird in Abschnitt 7.2 definiert. In Abschnitt 7.3 und 7.4 wird erläutert, wie wir für das Modell numerisch exakte Berechnungen auch auf größeren Permutationsproblemen, die eine eingeschränkte Struktur besitzen, durchführen können. Eine Fixpunktanalyse basierend auf den Werten der

Pheromonmatrix des ACO-Modells, welche sowohl die Analyse des Verhaltens des ACO-Modells als auch des ACO-Algorithmus unterstützt, wird in Abschnitt 7.5 vorgestellt. In Abschnitt 7.6 wird das dynamische Verhalten des ACO-Modells analysiert. In Abschnitt 7.7 wird das Verhalten von ACO-Modell und ACO-Algorithmus verglichen. In Abschnitt 7.8 wenden wir das Modell auf so genannte heterogene Permutationsprobleme an. In Abschnitt 7.9 fassen wir die Ergebnisse dieses Kapitels zusammen. Teile dieses Kapitels sind in (Merkle und Middendorf, 2002d,e) und (Merkle und Middendorf, 2002g) erschienen.

7.1 Einfache Probleme

Das generelle Prinzip des hier vorgestellten ACO-Modells ist nicht von einer bestimmten Klasse von Optimierungsproblemen abhängig. Die genauere Beschreibung und Analyse wird jedoch anhand von Permutationsproblemen durchgeführt, wie sie in Kapitel 2 eingeführt wurden. Wir wiederholen die Definition kurz. Gegeben seien n Elemente $1, 2, \dots, n$ und eine $n \times n$ Kostenmatrix $C = [c(i, j)]$ mit ganzzahligen Kosten $c(i, j) \geq 0$ (vgl. Abschnitt 2.2). \mathcal{P}_n sei wieder die Menge aller Permutationen von $(1, 2, \dots, n)$. Für eine Permutation $\pi \in \mathcal{P}_n$ seien wie üblich $c(\pi) = \sum_{i=1}^n c(i, \pi(i))$ die Kosten der Permutation. Zusätzlich definieren wir die Menge aller möglichen Werte der Kostenfunktion (oder auch Qualitätsstufen) $\mathcal{C} := \{c(\pi) \mid \pi \in \mathcal{P}_n\}$. Gesucht sei wieder eine Permutation $\pi \in \mathcal{P}_n$ der n Elemente, sodass die Kosten minimal sind, d.h. $c(\pi) = \min\{c(\pi') \mid \pi' \in \mathcal{P}_n\}$.

7.2 ACO-Modell

Um das Verhalten des ACO-Algorithmus numerisch exakt und analytisch zu untersuchen, wird ein deterministisches Modell für ACO-Algorithmen eingeführt. Wir geben in diesem Abschnitt den modellierten ACO-Algorithmus an, und führen dann das ACO-Modell ein. Bei der Modellierung wird insbesondere die Veränderung der Pheromonwerte während eines Laufs des ACO-Algorithmus von Interesse sein. Wie schon in der Einführung erwähnt, basiert die Pheromonaktualisierung beim ACO-Modell auf dem Erwartungswert der Pheromonaktualisierung eines ACO-Algorithmus, der sich bei unendlich vielen Generationen von jeweils m Ameisen für eine bestimmte Stelle in der Pheromonmatrix ergibt. Wenn wir von Auswahlwahrscheinlichkeiten im Kontext des ACO-Modells sprechen, dann meinen wir immer die deterministisch ermittelte Wahrscheinlichkeit, dass die beste von m modellierten Ameisen einer

Generation eine bestimmte Entscheidung trifft. Die Erwartungswerte für die Aktualisierung werden (aus den Auswahlwahrscheinlichkeiten) deterministisch ermittelt. Die Dynamik des Modells (also die Veränderung der Pheromonwerte des Modells) ist somit unabhängig von zufallsbasierten Einzelentscheidungen einzelner Ameisen. Ähnliche Ideen haben zu den in der Einführung dieses Kapitels erwähnten Modellen für GAs geführt, die mit unendlichen Populationen arbeiten (z.B. Vose und Liepins, 1991).

Der von uns in diesem Kapitel modellierte ACO-Algorithmus ist in Algorithmus 7.5 aufgeführt. Die Pheromonwerte werden so initialisiert, dass die

Algorithmus 7.5 Modellierter ACO-Algorithmus, Problemistanzgröße n

```

repeat
  Initialisiere alle Pheromonwerte  $\tau_{ij}$ 
  for Ameise  $k \in \{1, \dots, m\}$  do
     $S := \{1, 2, \dots, n\}$  // Menge der verfügbaren Elemente
    for  $i = 1$  to  $n$  do
      wähle  $j \in S$  mit Wahrscheinlichkeit  $p_{ij}$  entsprechend
      Gleichung (7.1) oder Gleichung (7.2)
       $S := S - \{j\}$ 
       $\pi(i) := j$ 
    end for
  end for
  for all  $(i,j)$  do
     $\tau_{ij} := (1 - \rho) \cdot \tau_{ij}$  // Verdunstung
  end for
  for all  $(i,j) \in$  beste Lösung der Iteration do
     $\tau_{ij} := \tau_{ij} + \rho$  // Aktualisierung
  end for
until Abbruchkriterium erreicht

```

Zeilen- und Spaltensummen jeweils genau 1 ergeben. Es werden nur Vorwärts-Ameisen verwendet. Falls es innerhalb einer Generation mehrere unterschiedliche Ameisen mit gleicher bester Lösungsqualität gibt, dann wird zur Aktualisierung wie üblich zufällig eine dieser Ameisen ausgewählt. Zur Berechnung der Auswahlwahrscheinlichkeiten wird entweder die lokale Pheromonauswertung entsprechend

$$p_{ij} = \frac{\tau_{ij}}{\sum_{h \in S} \tau_{ih}} \quad (7.1)$$

oder die Summen-Pheromonauswertung entsprechend

$$p_{ij} = \frac{\sum_{k=1}^i \tau_{kj}}{\sum_{h \in \mathcal{S}} (\sum_{k=1}^i \tau_{kh})} \quad (7.2)$$

ohne Heuristik und mit dem Exponenten der Pheromonwerte $\alpha = 1$ verwendet. Der Aktualisierungswert beim ACO-Algorithmus an einer Stelle der Pheromonmatrix ist immer $\Delta = 0$ oder $\Delta = \rho$. Die Zeilensummen der Pheromonmatrix (und somit auch die Spaltensummen) bleiben im aufgeführten Algorithmus immer konstant 1, da die Menge des verdunsteten Pheromons pro Zeile genau dem Aktualisierungswert entspricht.

Das ACO-Modell aktualisiert die Pheromonwerte entsprechend dem oben erläuterten Erwartungswerts. Pro Zeile und Spalte wird wie beim ACO-Algorithmus jeweils ρ verdunstet und aktualisiert, jedoch wird im Allgemeinen jedes Matrixelement verändert. Die konstanten Zeilen- und Spaltensummen sind wie beim ACO-Algorithmus ein invariantes Merkmal. Die neue Pheromonmatrix wird als Grundlage für die Berechnung der Erwartungswerte der nächsten Iteration verwendet. Man beachte, dass dies dazu führt, dass die Pheromonwerte des ACO-Modells mathematisch korrekt nur über eine Generation hinweg den erwarteten Pheromonwerten des ACO-Algorithmus entsprechen. Der Pseudocode für das ACO-Modell mit m Ameisen ist in Algorithmus 7.6 dargestellt. Dabei sei $\sigma_{ij}^{(m)}$ die Wahrscheinlichkeit, dass die beste von m

Algorithmus 7.6 ACO-Modell, Problemistanzgröße n

```

Initialisiere alle Pheromonwerte  $\tau_{ij}$  wie beim ACO-Algorithmus
repeat
  for all (i,j) do
     $\tau_{ij} := (1 - \rho) \cdot \tau_{ij}$  // Verdunstung
     $\tau_{ij} := \tau_{ij} + \rho \cdot \sigma_{ij}^{(m)}$  // Aktualisierung
  end for
until Abbruchkriterium erreicht

```

Ameisen in einer Generation Element j auf Platz i wählt. Aus dieser Auswahlwahrscheinlichkeit ergibt sich der erwartete Aktualisierungswert an einer Stelle der Pheromonmatrix des ACO-Modells durch die Multiplikation mit der Verdunstungsrate.

Im Folgenden werden wir erläutern, wie wir die Werte $\sigma_{ij}^{(m)}$, zunächst für $m = 1$, und dann für beliebige m berechnen können. Es sei darauf hingewiesen, dass die folgende formale Beschreibung in Abschnitt 7.4.2 ab Seite 146 an einem Beispiel veranschaulicht wird.

Um die erwartete Aktualisierung berechnen zu können, müssen die Wahrscheinlichkeiten der Einzelentscheidungen einer Ameise gegeben sein. Es sei wie üblich $[\tau_{ij}]$ eine Pheromonmatrix. Die Wahrscheinlichkeit $\sigma_\pi = \sigma_\pi^{(1)}$, dass eine Ameise die Lösung $\pi \in \mathcal{P}_n$ wählt, lässt sich als Produkt der Wahrscheinlichkeiten der Einzelentscheidungen einer Ameise für die lokale und die Summen-Pheromonauswertung wie folgt berechnen.

$$\sigma_\pi = \prod_{i=1}^n \frac{\tau_{i,\pi(i)}}{\sum_{j=i}^n \tau_{i,\pi(j)}} \quad (\text{lokale Pheromonauswertung}) \quad (7.3)$$

$$\sigma_\pi = \prod_{i=1}^n \frac{\sum_{k=1}^i \tau_{k,\pi(i)}}{\sum_{j=i}^n \sum_{k=1}^i \tau_{k,\pi(j)}} \quad (\text{Summen-Pheromonauswertung}) \quad (7.4)$$

Falls wir nicht ausdrücklich auf etwas anderes hinweisen, wird in diesem Kapitel die lokale Pheromonauswertung zur Berechnung der Auswahlwahrscheinlichkeiten verwendet. Die Wahrscheinlichkeit $\sigma_{ij} = \sigma_{ij}^{(1)}$, dass eine Ameise das Element j auf Platz i wählt, ist

$$\sigma_{ij} = \sum_{\pi \in \mathcal{P}_n} \sigma_\pi \cdot g(\pi, i, j) \quad (7.5)$$

Dabei ist die Hilfsfunktion g wie folgt definiert.

$$g(\pi, i, j) = \begin{cases} 1 & \text{falls } \pi(i) = j \\ 0 & \text{sonst} \end{cases} \quad (7.6)$$

Der Übersichtlichkeit wegen werden wir Hilfsfunktionen dieser Art in diesem Kapitel immer mit g bezeichnen, auch wenn g mit unterschiedlichen Parametern eine jeweils unterschiedliche Funktion angibt.

Es sei ein Problem P mit dazugehöriger Kostenmatrix und eine Pheromonmatrix gegeben. $\sigma_\pi^{(m)}$ sei die Wahrscheinlichkeit, dass die beste von m Ameisen die Lösungspermutation π gefunden hat. $\sigma_{ij}^{(m)}$ sei die Wahrscheinlichkeit, dass die beste von m Ameisen in einer Generation Element j auf Platz i wählt. Wir nennen die $\sigma_{ij}^{(m)}$ die Auswahlwahrscheinlichkeiten des ACO-Modells bei m Ameisen. Man beachte, dass die Gleichung (7.5) die Auswahlwahrscheinlichkeiten des ACO-Modells für $m = 1$ Ameise beschreibt. Es sei $\mathcal{I}_{\min}(P, \pi_1, \dots, \pi_m)$ die Menge aller Indizes der Permutationen π_1, \dots, π_m mit minimalen Kosten, d.h.

$$\mathcal{I}_{\min}(P, \pi_1, \dots, \pi_m) = \{i, i \in [1, m] \mid c(\pi_i) = \min\{c(\pi_j) \mid j \in [1, m]\}\}$$

Die Wahrscheinlichkeit $\sigma_\pi^{(m)}$ kann wie folgt berechnet werden.

$$\sigma_\pi^{(m)} = \sum_{(\pi_1, \dots, \pi_m) \in \mathcal{P}_n^m} \left(\prod_{k=1}^m \sigma_{\pi_k} \right) \cdot g(\pi_1, \dots, \pi_m, \pi)$$

Dabei gibt die Hilfsfunktion

$$g(\pi_1, \dots, \pi_m, \pi) = \frac{|\{k \in \mathcal{I}_{\min}(P, \pi_1, \dots, \pi_m) \mid \pi_k = \pi\}|}{|\mathcal{I}_{\min}(P, \pi_1, \dots, \pi_m)|}$$

den Anteil der Lösungspermutationen, die gleich π sind, an der Multimenge¹ aller Permutationen mit minimalen Kosten an. Der Wert $\prod_{k=1}^m \sigma_{\pi_k}$ gibt die Wahrscheinlichkeit an, dass die m Ameisen die Lösungen π_1, \dots, π_m wählen. Die Wahrscheinlichkeit $\sigma_{ij}^{(m)}$ ist

$$\sigma_{ij}^{(m)} = \sum_{(\pi_1, \dots, \pi_m) \in \mathcal{P}_n^m} \left(\prod_{k=1}^m \sigma_{\pi_k} \right) \cdot g(\pi_1, \dots, \pi_m, i, j) \quad (7.7)$$

wobei

$$g(\pi_1, \dots, \pi_m, i, j) = \frac{|\{k \in \mathcal{I}_{\min}(P, \pi_1, \dots, \pi_m) \mid \pi_k(i) = j\}|}{|\mathcal{I}_{\min}(P, \pi_1, \dots, \pi_m)|} \quad (7.8)$$

den Anteil der Lösungen, für die $\pi(i) = j$ gilt, an der Multimenge aller Permutationen mit minimalen Kosten angibt. Man beachte hierbei, dass bei gleicher bester Lösungsqualität unterschiedlicher Lösungspermutationen beim modellierten ACO-Algorithmus eine zufällig ausgewählte Ameise die Aktualisierung durchführt. Nach einer Iteration des ACO-Modells, welche die Lösungskonstruktion von m Ameisen simuliert, wird die Pheromonaktualisierung im Modell wie folgt durchgeführt.

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \sigma_{ij}^{(m)}$$

Man beachte, dass, wie beim ACO-Algorithmus auch, die Zeilen- und Spaltensummen nach dieser Aktualisierung immer 1 sind.

Im Folgenden wird eine weitere Variante angegeben, wie die Auswahlwahrscheinlichkeit, dass die beste von m Ameisen Element j auf Platz i wählt, berechnet werden kann. In Abschnitt 7.3 werden wir homogene Permutationsprobleme einführen. Das sind Probleme, die aus mehreren unabhängigen identischen Teilproblemen zusammengesetzt sind. Mit der alternativen Methode

¹Eine Multimenge ist eine Menge, in der ein und dasselbe Element mehrfach vorkommen kann.

wird bei der Betrachtung dieser homogenen Permutationsprobleme ein analytischer Zusammenhang zu dem Verhalten des Modells auf einem der elementaren Teilprobleme des eingeschränkten Problems hergestellt. Des Weiteren wird mit der alternativen Methode deutlich gemacht, dass sich die Dynamik eines ACO-Algorithmus als eine einfach zu formulierende gewichtete Summe über alle unterschiedlichen Qualitätsstufen beschreiben lässt. Wir werden diese Aussage im Folgenden formalisieren. Auch die alternative Variante zur Berechnung der Auswahlwahrscheinlichkeiten wird in Abschnitt 7.4.2 anhand eines Beispiels veranschaulicht.

\mathcal{C} sei die Menge aller möglichen Qualitätsstufen (vgl. Abschnitt 7.1 auf Seite 131). Es sei $\xi_x^{(m)}$ die Wahrscheinlichkeit, dass die beste von m Ameisen in einer Generation eine Lösung mit genau der Qualität $x \in \mathcal{C}$ findet. Es sei $\omega_{ij}^{(x)}$ die Wahrscheinlichkeit, dass eine Ameise, die eine Lösung mit Qualität $x \in \mathcal{C}$ gefunden hat, Element j auf Platz i gewählt hat. Dann lässt sich $\sigma_{ij}^{(m)}$ wie folgt berechnen.

$$\sigma_{ij}^{(m)} = \sum_{x \in \mathcal{C}} \xi_x^{(m)} \cdot \omega_{ij}^{(x)} \quad (7.9)$$

Dabei ist

$$\xi_x^{(m)} = \sum_{\pi \in \mathcal{P}_{n,c}(\pi)=x} \sigma_\pi^{(m)}$$

und

$$\omega_{ij}^{(x)} = \frac{\sum_{\pi \in \mathcal{P}_{n,c}(\pi)=x} \sigma_\pi \cdot g(\pi, i, j)}{\sum_{\pi \in \mathcal{P}_{n,c}(\pi)=x} \sigma_\pi} \quad (7.10)$$

Die Auswahlwahrscheinlichkeit, dass die beste von m Ameisen eine bestimmte Entscheidung trifft, lässt sich also durch eine gewichtete Summe der Wahrscheinlichkeiten $\omega_{ij}^{(x)}$ über alle Qualitätsstufen berechnen. Man beachte, dass die Anzahl der Ameisen m nur die Gewichte $\xi_x^{(m)}$ der Wahrscheinlichkeiten $\omega_{ij}^{(x)}$ bei einzelnen Qualitätsstufen verändert. $\omega_{ij}^{(x)}$ ist von m unabhängig. Je mehr Ameisen in einer Generation verwendet werden, desto höher wird das Gewicht für die Werte $\omega_{ij}^{(x)}$ der optimalen Qualitätsstufe. Auch dies werden wir im Abschnitt 7.4.2 anhand eines ausführlichen Beispiels erläutern.

7.3 Eingeschränkte Permutationsprobleme

Viele komplexe, praxisorientierte Probleme bestehen aus Teilproblemen, die oft nahezu oder vollständig unabhängig voneinander sind. Um das Verhalten von ACO-Algorithmen auf solchen Problemen zu untersuchen, wird der Sachverhalt auf eine idealisierte Weise modelliert. Wir betrachten hierfür eingeschränkte Permutationsprobleme, die aus mehreren voneinander unabhängigen Teilproblemen bestehen. Natürlich ist es besser, die voneinander unabhängigen Teilprobleme einzeln zu lösen. Hierfür muss man aber wissen, dass es sich beim betrachteten Problem um ein solches handelt. Die Identifikation der Teilprobleme ist oft nicht ohne weiteres möglich.

Für ein Permutationsproblem P der Größe n_0 sei P^q ein *eingeschränktes Permutationsproblem*, welches aus q unabhängigen Permutationsproblemen P besteht. Bevor wir die formale Definition einführen, geben wir ein Beispiel an.

Beispiel: Gegeben sei eine Instanz P_1 eines einfachen Problems mit der Kostenmatrix

$$C_1 = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix} \quad (7.11)$$

Diese Instanz P_1 entspricht bis auf den Größenunterschied der in Abschnitt 2.3 auf Seite 16 eingeführten einfachen Problem Instanz P_A , auf der die Verzerrung der Auswahlwahrscheinlichkeiten der unterschiedlichen Pheromonauswertungsmethoden untersucht wurde. Die Instanz P_1 kann als eine sehr einfache Instanz eines Ein-Maschinen-Planungsproblems gesehen werden, bei der eine Menge von Jobs bei gegebenen Laufzeiten und Fälligkeitsterminen auf einer Maschine eingeplant werden muss. Gesucht ist eine Permutation der Jobs, für welche die Summe der Abweichungen der einzelnen Jobs von deren jeweils optimalem Fälligkeitstermin minimiert wird. Für unser Beispiel haben wir drei Jobs, jeder der Jobs hat Laufzeit 1 und Job i hat das Fälligkeitsdatum i , $i \in [1, 3]$. Das entsprechende (homogene) eingeschränkte Permutationsproblem P_1^3 hat folgende Kostenmatrix.

$$C_1^{(3)} = \begin{pmatrix} 0 & 1 & 2 & \infty & \infty & \infty & \infty & \infty & \infty \\ 1 & 0 & 1 & \infty & \infty & \infty & \infty & \infty & \infty \\ 2 & 1 & 0 & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & 1 & 2 & \infty & \infty & \infty \\ \infty & \infty & \infty & 1 & 0 & 1 & \infty & \infty & \infty \\ \infty & \infty & \infty & 2 & 1 & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & 2 \\ \infty & \infty & \infty & \infty & \infty & \infty & 1 & 0 & 1 \\ \infty & \infty & \infty & \infty & \infty & \infty & 2 & 1 & 0 \end{pmatrix} \quad (7.12)$$

Formal definieren wir für ein Permutationsproblem P der Größe n_0 ein *eingeschränktes Permutationsproblem* P^q , welches aus q unabhängigen gleich großen Teilproblemen besteht, wie folgt.

Es seien C_1, C_2, \dots, C_q die Kostenmatrizen der q Teilprobleme von P^q , und es sei $c_{ij}^{(r)}$ das Element (i, j) der Matrix C_r , $r \in [1, q]$. Ein eingeschränktes Permutationsproblem P^q ist ein Problem, bei dem ein Element $(r-1) \cdot n_0 + j$, $r \in [1, q]$, $j \in [1, n_0]$ auf einem der Plätze $(r-1) \cdot n_0 + 1, (r-1) \cdot n_0 + 2, \dots, (r-1) \cdot n_0 + n_0$ platziert werden muss. Die Kosten, die entstehen, wenn Element $(r-1) \cdot n_0 + j$, $j \in [1, n_0]$ auf Platz $(r-1) \cdot n_0 + i$, $i \in [1, n_0]$ platziert wird, sind $c_{ij}^{(r)}$. Es sei $C = [c_{ij}]$ die zu P^q gehörende Kostenmatrix, sodass $c_{(r-1)n_0+i, (r-1)n_0+j} = c_{ij}^{(r)}$ für $r \in [1, q]$ und $i, j \in [1, n_0]$ gilt. Anderenfalls gilt $c_{ij} = \infty$. Man beachte, dass es bei dieser Definition eingeschränkter Permutationsprobleme nicht erlaubt ist, eine Entscheidung mit unendlich hohen Kosten zu machen. P nennen wir ein elementares Teilproblem von P^q , und die q Instanzen von P , die zusammen das Problem P^q erzeugen, nennen wir elementare Teilinstanzen von P^q . Falls alle Kostenmatrizen C_1, C_2, \dots, C_q gleich sind, d.h. $C_1 = C_2 = \dots = C_q$, dann bezeichnen wir P^q als homogenes eingeschränktes Permutationsproblem oder einfach *homogenes Permutationsproblem*. Anderenfalls bezeichnen wir das Problem als heterogenes eingeschränktes Permutationsproblem oder einfach *heterogenes Permutationsproblem*.

Im Folgenden untersuchen wir, wie man das Verhalten von ACO-Algorithmen auf eingeschränkten (evtl. heterogenen) Permutationsproblemen annähern kann, indem man das Verhalten des ACO-Modells auf einem einzelnen elementaren Teilproblem P analysiert. Hierfür betrachten wir eine beliebige der q Teilinstanzen von P^q , beispielsweise die r -te Teilinstanz, sowie die Qualität der Lösungen, welche die m Ameisen auf den restlichen $q-1$ elementaren Teilproblemen gefunden haben (diese $q-1$ Teilprobleme bilden eine

Instanz des Problems P^{q-1}). Betrachten wir zunächst die Lösungsqualität der Ameisen auf dem Teilproblem P^{q-1} . Offensichtlich haben Ameisen, die schon auf P^{q-1} bessere Lösungen gefunden haben, auch größere Chancen, auf dem Problem P^q die bessere Lösung zu finden. Wir modellieren dies, indem wir annehmen, dass manche Ameisen in einer Generation einen *Malus* erhalten. Bei der Berechnung der Aktualisierungswerte einer Iteration benötigen wir die Auswahlwahrscheinlichkeiten $\sigma_{ij}^{(m)}$, dass die beste von m Ameisen Element j auf Platz i wählt. Man beachte, dass beim Vergleich der Lösungsqualität von mit Malus behafteten Ameisen die Qualität der Lösungspermutationen der Ameisen inklusive deren jeweiligen Malus verglichen wird. Der Malus einer Ameise modelliert den Abstand hinsichtlich der Lösungsqualität zur besten Ameise einer Iteration auf P^{q-1} . Es sei d_{\max} der maximale Abstand zweier Lösungsqualitäten des r -ten Teilproblems. Wir rechnen

$$d_{\max} := \max\{x \mid x \in \mathcal{C}_r\} - \min\{x \mid x \in \mathcal{C}_r\},$$

wobei \mathcal{C}_r die Menge aller möglichen Qualitätsstufen des r -ten elementaren Teilproblems ist.

Es sei für jedes $i \in [1, m]$ der zugehörige Malus einer Ameise i mit $d_i \geq 0$ bezeichnet. Wir nehmen immer an, dass Ameise 1 keinen Malus hat, d.h. $d_1 = 0$, und dass die Ameisen o.B.d.A. entsprechend ihrer Lösungsqualität sortiert sind, d.h. $d_2 \leq \dots \leq d_m$. Der Malus d_i , $i \in [2, m]$, ist das Minimum aus $d_{\max} + 1$ und dem modellierten Unterschied der Kosten der Permutation der Ameise i auf P^{q-1} minus den Kosten der Permutation der Ameise 1 auf P^{q-1} . Es sei $\xi_x^{(m; d_2, \dots, d_m)}$ die Wahrscheinlichkeit, dass die (inklusive Malus) beste aller m Ameisen eine Lösung mit Qualität $x \in \mathcal{C}$ (ohne Malus) gefunden hat, wobei Ameise $i \in [2, m]$ einen Malus von d_i hat. Man beachte hierbei, dass die Ameise mit der geringsten Lösungsqualität aufgrund der Malusse nicht notwendigerweise die beste Ameise sein muss. Für die Beschreibung eines Lösungsqualitätsunterschiedes, der größer als d_{\max} ist, werden wir auch die Schreibweise $> d_{\max}$ verwenden. Somit beschreibt sowohl $d_i > d_{\max}$ als auch $d_i = d_{\max} + 1$ eine Situation, bei der eine Ameise einen Malus hat, der größer als d_{\max} ist. Mit (d_2, \dots, d_m) , $d_i \in [1, d_{\max} + 1]$, $i \in [2, m]$ bezeichnen wir eine *Malussituation*. Wir sprechen von einer *Konkurrenz* der Ameisen, wenn wir auf dem betrachteten Problem eine Malussituation $(0, \dots, 0)$ haben. Im Falle $m = 1$, oder falls wir die Malussituation $(> d_{\max}, \dots, > d_{\max})$ haben, sprechen wir von einer Situation ohne Konkurrenz. Bei allen anderen Malussituationen sprechen wir von schwacher Konkurrenz.

Die Auswahlwahrscheinlichkeiten $\sigma_{ij}^{(m; d_2, \dots, d_m)}$ für m Ameisen auf dem r -

ten elementaren Teilproblem, wobei eine Malussituation (d_2, \dots, d_m) gegeben ist, berechnen wir dann wie folgt (vgl. Gleichung (7.25) auf Seite 151).

$$\sigma_{ij}^{(m;d_2,\dots,d_m)} = \sum_{x \in \mathcal{C}} \xi_x^{(m;d_2,\dots,d_m)} \cdot \omega_{ij}^{(x)} \quad (7.13)$$

Die Auswahlwahrscheinlichkeit $\sigma_{ij}^{(m;d_2,\dots,d_m)}$ gibt die Wahrscheinlichkeit an, dass die (inklusive Malus) beste der m Ameisen auf einem elementaren Teilproblem P_r einer Instanz des Problems P^q das Element j auf Platz i wählt, wobei der Lösungsqualitätsunterschied der Ameisen auf P^{q-1} durch die Malussituation (d_2, \dots, d_m) modelliert wird.

Im Folgenden werden wir die *Wahrscheinlichkeiten von Malussituationen* berechnen, d.h. die Wahrscheinlichkeiten dafür, dass sich auf $q-1$ der q Teilprobleme ein bestimmter Lösungsqualitätsunterschied der Ameisen ergeben hat. Mit $\phi_{d_2,\dots,d_m}^{(m)}$ bezeichnen wir die Wahrscheinlichkeit, dass von m Ameisen eine Malussituation (d_2, \dots, d_m) auf P^{q-1} angenommen wird, d.h. der Lösungsqualitätsunterschied der i -ten Ameise und der besten Ameise ist d_i , falls $d_i \leq d_{\max}$ gilt, und der Lösungsqualitätsunterschied der i -ten Ameise und der besten Ameise ist größer als d_{\max} falls $d_i = d_{\max} + 1$ gilt. Da wir für die Beschreibung eines Lösungsqualitätsunterschiedes, der größer als d_{\max} ist, auch die Schreibweise $> d_{\max}$ verwenden, gilt z.B. $\phi_{2,>d_{\max}}^{(3)} = \phi_{2,d_{\max}+1}^{(3)}$.

Wir berechnen nun die Auswahlwahrscheinlichkeiten für die elementaren Teilprobleme eines eingeschränkten Permutationsproblems mit Hilfe der Wahrscheinlichkeiten von Malussituationen. Es sei \mathcal{D} die Menge aller durch Vektoren beschriebenen Malussituationen (d_2, \dots, d_m) mit ganzzahligen Werten $d_2 \leq \dots \leq d_m$, $0 \leq d_i \leq d_{\max} + 1$, $i \in [2, m]$. Dann erhalten wir für das r -te elementare Teilproblem von P^q durch eine Summe über alle möglichen Malussituationen

$$\sigma_{ij}^{(m)} = \sum_{(d_2,\dots,d_m) \in \mathcal{D}} \phi_{d_2,\dots,d_m}^{(m)} \cdot \sigma_{ij}^{(m;d_2,\dots,d_m)} \quad (7.14)$$

$$\begin{aligned} &= \sum_{(d_2,\dots,d_m) \in \mathcal{D}} \phi_{d_2,\dots,d_m}^{(m)} \cdot \sum_{x \in \mathcal{C}} \xi_x^{(m;d_2,\dots,d_m)} \cdot \omega_{ij}^{(x)} \\ &= \sum_{x \in \mathcal{C}} w_x \cdot \omega_{ij}^{(x)} \end{aligned} \quad (7.15)$$

mit Gewichten

$$w_x = \sum_{(d_2,\dots,d_m) \in \mathcal{D}} \phi_{d_2,\dots,d_m}^{(m)} \cdot \xi_x^{(m;d_2,\dots,d_m)}$$

$$\mathbf{C} = \left(\begin{array}{c} \begin{array}{ccc} \mathbf{C}_1 & & \\ & \mathbf{C}_2 & \mathbf{0} \\ & \cdot & \cdot \\ & \cdot & \cdot \\ \mathbf{0} & & \mathbf{C}_{q-1} \end{array} & \begin{array}{c} \vdots \\ \mathbf{0} \\ \vdots \end{array} \\ \dots \mathbf{0} \dots & \mathbf{C}_q \end{array} \right)$$

Abbildung 7.1: Darstellung der Kostenmatrix \mathbf{C} einer Instanz des eingeschränkten Permutationsproblems P^q und der Berechnung der Auswahlwahrscheinlichkeiten des ACO-Modells: Die grundlegende Idee zur Berechnung der Auswahlwahrscheinlichkeiten eines beliebigen (hier des letzten) Teilproblems besteht darin, zunächst die Wahrscheinlichkeiten zu berechnen, dass auf P^{q-1} (eingefärbter Bereich) eine gewisse Malussituation angenommen wird. Die Auswahlwahrscheinlichkeiten auf dem (hier letzten) elementaren Teilproblem ergeben sich aus einer gewichteten Summe der Auswahlwahrscheinlichkeiten, die sich auf dem elementaren Teilproblem ausgehend von allen unterschiedlichen Malussituationen ergeben. Die Gewichte der Auswahlwahrscheinlichkeiten entsprechen den Wahrscheinlichkeiten, dass die entsprechenden Malussituationen auf P^{q-1} auftreten (vgl. Gleichung (7.14) und Abschnitt 7.4.2).

Dies ist ein sehr interessantes Ergebnis, weil es zeigt, dass sich der Einfluss des Problems P^{q-1} auf das verbleibende elementare Teilproblem P_r von P^q nur auf die Veränderung der Gewichte der verschiedenen Qualitätsstufen bei der Berechnung der Auswahlwahrscheinlichkeiten des elementaren Teilproblems $P = P_r$ auswirkt (vgl. Formel (7.9) auf Seite 136). Die Vorgehensweise zur Berechnung der Auswahlwahrscheinlichkeiten des ACO-Modells ist für ein eingeschränktes Permutationsproblem in Abbildung 7.1 veranschaulicht.

Für $m = 1$ Ameise treten keine Malussituationen auf, und die Auswahlwahrscheinlichkeiten ergeben sich auf jedem elementaren Teilproblem entsprechend Gleichung (7.5) auf Seite 134. Wir betrachten den Fall $m = 2$ Ameisen genauer. Die Wahrscheinlichkeit, Element j des r -ten Teilproblems auf Platz i , d.h. Element $(r-1)n_0 + j$, $r \in [1, q]$, $j \in [1, n_0]$ des Problems P^q auf Platz $(r-1)n_0 + i$, $i \in [1, n_0]$ zu platzieren, wird wie folgt berechnet (vgl. auch

Gleichung (7.14) auf Seite 140).

$$\sigma_{(r-1)n_0+i, (r-1)n_0+j}^{(2)} = \phi_{>d_{\max}}^{(2)} \cdot \sigma'_{ij} + \sum_{d=1}^{d_{\max}} \phi_d^{(2)} \cdot \sigma_{ij}^{\prime(2;d)} + \phi_0^{(2)} \cdot \sigma_{ij}^{\prime(2)} \quad (7.16)$$

Dabei bezeichnen σ'_{ij} , $\sigma_{ij}^{\prime(2;d)}$ und $\sigma_{ij}^{\prime(2)}$ die Auswahlwahrscheinlichkeiten des ACO-Modells auf einem elementaren Teilproblem P bei der entsprechenden Malussituation. Beispielsweise ist $\sigma_{ij}^{\prime(2)}$ die Wahrscheinlichkeit, dass die beste von 2 Ameisen Element j auf Platz h des elementaren Teilproblems P_r wählt, wobei keine Ameise einen Malus hat. Die Auswahlwahrscheinlichkeiten $\sigma_{(r-1)n_0+i, (r-1)n_0+j}^{(2)}$ des Problems P^q werden aus einer gewichteten Summe von Auswahlwahrscheinlichkeiten der elementaren Teilprobleme berechnet werden, die sich ausgehend von den entsprechenden Malussituationen ergeben (vgl. Gleichung (7.16)). Die Gewichte sind dabei die Wahrscheinlichkeiten dafür, dass zwei Ameisen auf dem Problem P^{q-1} einen gewissen Unterschied bezüglich ihrer Lösungsqualität haben, und sich somit die entsprechende Malussituation ergibt. Sehr interessant ist folgende Beobachtung. Wenn sich der Algorithmus in einer Phase befindet, in der die zwei Ameisen hinsichtlich ihrer Lösungsqualität mit hoher Wahrscheinlichkeit einen Unterschied $d > d_{\max}$ haben, dann sind die Auswahlwahrscheinlichkeiten, dass die beste von m Ameisen auf P^q Element $(r-1)n_0+j$ auf Platz $(r-1)n_0+i$ wählt ungefähr σ'_{ij} , d.h. die Auswahlwahrscheinlichkeiten entsprechen nahezu dem Fall, dass eine einzelne Ameise Element j auf Platz i des elementaren Teilproblems wählt. Wir sprechen hier wieder von einer *Verzerrung der Auswahlwahrscheinlichkeiten*, da die Pheromonwerte nicht die tatsächlichen Auswahlwahrscheinlichkeiten widerspiegeln. Der Fall, dass die Lösungsqualität beider Ameisen gleich ist, entspricht dem letzten Teil der Formel (7.16). Diese Wahrscheinlichkeit wird in der Konvergenzphase sehr groß und entspricht dem Fall, dass die beste der beiden Ameisen bei reiner Konkurrenz Element j auf Platz i wählt. Alle anderen Fälle entsprechen einer Situation, bei der eine der beiden Ameisen einen Malus hat, der höchstens d_{\max} groß ist (mittlerer Teil der Formel).

Da die Entscheidungen der Ameisen auf den elementaren Teilproblemen unabhängig voneinander sind, wächst für eine größere Anzahl der Teilprobleme q die Wahrscheinlichkeit $\phi_{>d_{\max}}^{(2)}$. Dies werden wir in Abschnitt 7.6 bei der Untersuchung des dynamischen Verhaltens veranschaulichen. Eine wichtige Konsequenz dieser Beobachtung ist, dass der (positiv wirkende) Einfluss der Konkurrenz beider Ameisen um eine gute Lösung mit wachsendem q schwächer wird, und die Auswahlwahrscheinlichkeiten immer stärker einer ungewünschten Verzerrung unterliegen (vgl. auch die empirischen Untersuchungen

der Verzerrung der Auswahlwahrscheinlichkeiten z.B. in Abschnitt 2.4 ab Seite 19). Die Auswahlwahrscheinlichkeiten eines Teilproblems des eingeschränkten Permutationsproblems mit großen q entsprechen daher nahezu den Auswahlwahrscheinlichkeiten einer einzelnen Ameise auf einem entsprechenden elementaren Teilproblem.

7.4 Homogene Permutationsprobleme

Nach der Modellbeschreibung für homogene Permutationsprobleme in Abschnitt 7.4.1 werden wir in Abschnitt 7.4.2 ein ausführliches Beispiel für die Berechnung von Auswahlwahrscheinlichkeiten angeben.

7.4.1 ACO-Modell

Nehmen wir nun zusätzlich an, dass das eingeschränkte Permutationsproblem homogen ist, d.h. alle Teilmatrizen der Kostenmatrix des Problems P^q , welche zu den einzelnen elementaren Teilinstanzen gehören, sind identisch. Wenn wir bei homogenen Permutationsproblemen davon sprechen, dass wir die Auswahlwahrscheinlichkeiten für P^q berechnen, dann meinen wir damit die Auswahlwahrscheinlichkeiten auf jedem beliebigen elementaren Teilproblem. Denn da das Problem P^q eine homogene Struktur hat, ergeben sich auch auf jedem beliebigen elementaren Teilproblem die gleichen Auswahlwahrscheinlichkeiten.

Die Hauptschwierigkeit, um für ein homogenes Permutationsproblem P^q die Auswahlwahrscheinlichkeiten $\sigma_{(r-1)n_0+i, (r-1)n_0+j}^{(m)}$ zu berechnen, liegt in der Berechnung der Wahrscheinlichkeiten $\phi_{d_2, \dots, d_m}^{(m)}$, dass die Malussituation (d_2, \dots, d_m) angenommen wird. Zur Berechnung verwenden wir homogene Markovketten. Der Zustandsraum der Markovkette ist durch die Menge aller Tupel (d_2, \dots, d_m) , $0 \leq d_i \leq (q-1) \cdot d_{\max}$, $i \in [2, m]$ definiert. Ein Übergang in der Markovkette entspricht der Auswertung eines elementaren Teilproblems für jede der m Ameisen. Die Übergangsmatrix definiert die Wahrscheinlichkeiten, um unter Berücksichtigung der Ergebnisse der m Ameisen auf einem elementaren Teilproblem von einem Zustand (d_2, \dots, d_m) zu einem Zustand (d'_2, \dots, d'_m) zu gelangen. Man beachte, dass wir formal eine Malussituation so definiert haben, dass ein Lösungsqualitätsunterschied, der größer als ein vorgegebenes d_{\max} ist, durch $d_{\max} + 1$ bezeichnet wird. Daher entspricht ein Zustand der homogenen Markovkette nicht notwendigerweise einer Malussituation. Wir werden daher einfach von Zuständen, und nicht von Malussituationen sprechen. Die Wahrscheinlichkeit, um in einem Schritt von einem

Zustand $(0, \dots, 0)$ wieder genau in den gleichen Zustand der Markovkette überzugehen, entspricht genau der Wahrscheinlichkeit, dass alle Ameisen auf dem zuletzt betrachteten elementaren Teilproblem eine Lösung mit gleicher Lösungsqualität gefunden haben. Für das Problem P_1 (vgl. Gleichung (7.11)) ist diese Wahrscheinlichkeit

$$\left(\xi_0^{(1)}\right)^m + \left(\xi_2^{(1)}\right)^m + \left(\xi_4^{(1)}\right)^m, \quad (7.17)$$

falls $\xi_x^{(1)}$ die Wahrscheinlichkeit bezeichnet, dass eine einzelne Ameise eine Lösung mit Qualität x auf dem Problem P_1 findet. Die weiteren Übergangswahrscheinlichkeiten werden analog definiert. In Abbildung 7.2 ist diese Vorgehensweise für P_1^q und $m = 3$ Ameisen veranschaulicht. Die Wahrscheinlichkeit einer bestimmten Malussituation $(d_2, d_3) \in \{(0, 0), (0, 2), (0, 4), (2, 2), (2, 4), (4, 4)\}$ nach $q - 1$ ausgewerteten elementaren Teilproblemen entspricht der entsprechenden Zustandswahrscheinlichkeit eines Zustands der Markovkette nach $q - 1$ Schritten. Die Wahrscheinlichkeiten, dass eine der 4 weiteren Malussituationen $(d_2, d_3) \in \{(0, 5), (2, 5), (4, 5), (5, 5)\}$ angenommen wird, ergeben sich durch die Summe der Zustandswahrscheinlichkeiten der jeweils schraffierten Zustandsklassen.

Die Wahrscheinlichkeit, nach $q - 1$ Schritten in einem gegebenen Zustand zu sein, kann durch eine in $m - 1$ Variablen rekursive Funktion berechnet werden. Die benötigte Rekursionstiefe ist $q - 1$. Für kleine Systeme kann diese Wahrscheinlichkeit noch schnell berechnet werden. Die Berechnung benötigt beispielsweise für $m = 2$ und $q = 128$ ca. 0.02 Sekunden auf einem PC mit einem 500MHz Pentium III Prozessor, für $m = 3$ und $q = 64$ werden ca. 0.3 Sekunden benötigt.

Für den Fall $m = 2$ Ameisen kann $\phi_d^{(2)}$ auch alternativ wie folgt berechnet werden. Es sei ψ_d (ψ'_d) die Wahrscheinlichkeit, dass der Unterschied der Lösungsqualität der ersten Ameise minus der Lösungsqualität der zweiten Ameise auf dem Problem P^{q-1} (entsprechend P) genau d ist. Für diese Überlegung nehmen wir nicht an, dass die erste Ameise unbedingt eine bessere Lösung findet, d.h. d kann negativ sein. Der Lösungsqualitätsunterschied auf dem Teilproblem P^{q-1} kann als Ergebnis eines verallgemeinerten eindimensionalen Random Walks (Irrfahrt) beschrieben werden. Jede Auswertung eines der $q - 1$ elementaren Teilprobleme entspricht einem Schritt des Random Walks. Es sei k_i die Anzahl der elementaren Teilinstanzen von P^{q-1} , auf denen der Lösungsqualitätsunterschied der ersten und zweiten Ameise $i \in [-d_{\max}, d_{\max}]$

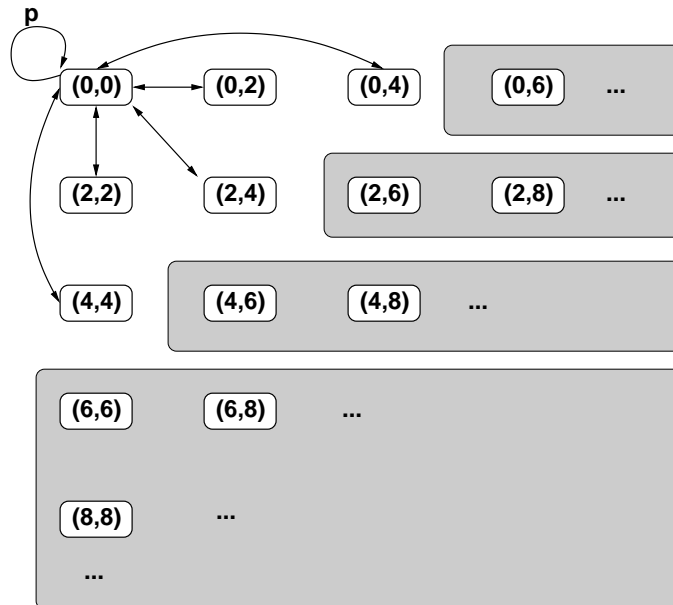


Abbildung 7.2: Homogene Markovkette zur Berechnung der Zustandswahrscheinlichkeiten bei $m = 3$ Ameisen für das homogene Permutationsproblem P_1^q : Die Zustände der Markovkette entsprechen Tupeln (d_2, d_3) , wobei d_2 der Abstand der besten zur zweitbesten Ameise, und d_3 der Abstand der besten zur drittbesten (schlechtesten) Ameise ist. Der Zustand $(0, 0)$ entspricht der Malussituation, dass alle drei Ameisen die gleiche Qualität haben. Ein Zustandsübergang entspricht der Auswertung eines elementaren Teilproblems. Für den Zustand $(0, 0)$ sind alle möglichen Zustandsübergänge eingezeichnet. Vor der Auswertung des ersten Teilproblems befinden wir uns mit Wahrscheinlichkeit 1.0 im Zustand $(0, 0)$. Beispielsweise die Wahrscheinlichkeit p des Zustandsübergangs von $(0, 0)$ nach $(0, 0)$ ergibt sich aus den Wahrscheinlichkeiten, dass alle drei Ameisen die gleiche Lösungsqualität auf einem elementaren Teilproblem finden. Wir berechnen $p = \left(\xi_0^{(1)}\right)^3 + \left(\xi_2^{(1)}\right)^3 + \left(\xi_4^{(1)}\right)^3$, wobei $\xi_x^{(1)}$ die Wahrscheinlichkeit bezeichnet, dass eine einzelne Ameise eine Lösung mit Qualität x auf dem Problem P_1 findet. In den schraffierten Bereichen sind die Zustände zusammengefasst, die einzelnen Malussituationen entsprechen. Insgesamt existieren 10 unterschiedliche Klassen von Zuständen (6 dieser Klassen enthalten genau einen Zustand, 4 der Klassen enthalten mehr als einen Zustand). Diese 10 Malussituationen müssen bei der Berechnung der Auswahlwahrscheinlichkeiten unterschieden werden (vgl. auch Abbildung 7.14 auf Seite 167).

ist. Wir definieren

$$I_d = \left\{ (k_{-d_{\max}}, k_{-d_{\max}+1}, \dots, k_{d_{\max}}) \mid \begin{aligned} q-1 &= \sum_{i=-d_{\max}}^{d_{\max}} k_i, \quad d = \sum_{i=-d_{\max}}^{d_{\max}} i \cdot k_i \end{aligned} \right\} \quad (7.18)$$

Dann können die ψ_d entsprechend der Multinomialverteilung wie folgt berechnet werden (vgl. auch Beispiel in Abschnitt 7.4.2 ab Seite 152).

$$\psi_d = \sum_{(k_{-d_{\max}}, \dots, k_{d_{\max}}) \in I_d} \frac{(q-1)!}{k_{-d_{\max}}! \cdot \dots \cdot k_{d_{\max}}!} \cdot \prod_{i=-d_{\max}}^{d_{\max}} (\psi'_i)^{k_i} \quad (7.19)$$

Offensichtlich gilt

$$\phi_0^{(2)} = \psi_0$$

und aus Symmetriegründen

$$\phi_d^{(2)} = 2 \cdot \psi_d$$

für $d \neq 0$. Des Weiteren gilt

$$\phi_{>d_{\max}}^{(2)} = 2 \cdot \sum_{d=d_{\max}+1}^{(q-1) \cdot d_{\max}} \psi_d = 1 - \sum_{y=0}^{d_{\max}} \phi_y^{(2)}$$

7.4.2 Beispiel

In diesem Abschnitt berechnen wir beispielhaft für eine homogene Problem-Instanz unter anderem die Auswahlwahrscheinlichkeiten, die sich für $m = 2$ Ameisen ergeben. Wir betrachten das homogene eingeschränkte Permutationsproblem P_1^q mit der dazugehörigen Kostenmatrix des elementaren Teilproblems P_1 , die in Gleichung (7.11) gegeben ist. Die möglichen Qualitätsstufen für das Problem P_1 sind 0, 2 und 4. Die (einzige) optimale Lösung für das Problem P_1 ergibt sich, wenn Element i auf Platz i , $i \in [1, 3]$ platziert wird. Es sei $l \in [1, q]$ und $i, j \in [1, 3]$. Für das homogene Permutationsproblem P_1^q (vgl. Gleichung (7.16) auf Seite 142) gilt

$$\sigma_{3(r-1)+i, 3(r-1)+j}^{(2)} = \phi_{>4}^{(2)} \cdot \sigma'_{ij} + \sum_{d \in \{2, 4\}} \phi_d^{(2)} \cdot \sigma_{ij}^{\prime(2;d)} + \phi_0^{(2)} \cdot \sigma_{ij}^{\prime(2)} \quad (7.20)$$

Die Berechnung der Auswahlwahrscheinlichkeiten für das homogene Permutationsproblem P_1^q ergibt sich als eine gewichtete Summe von Auswahlwahrscheinlichkeiten, die sich auf dem Permutationsproblem P_1 ausgehend von

den Malussituationen 0, 2, 4 und > 4 ergeben. Die Gewichte entsprechen den Wahrscheinlichkeiten, dass ein Lösungsqualitätsunterschied 0, 2, 4 oder > 4 auf P_1^{q-1} eintritt. Die Auswahlwahrscheinlichkeiten für P_1 , ausgehend von unterschiedlichen Malussituationen, wollen wir im Folgenden berechnen.

Es sei als Beispiel folgende Pheromonmatrix gegeben.

$$\begin{pmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \end{pmatrix} = \begin{pmatrix} 0.1 & 0.3 & 0.6 \\ 0.6 & 0.1 & 0.3 \\ 0.3 & 0.6 & 0.1 \end{pmatrix} \quad (7.21)$$

Auswahlwahrscheinlichkeiten bei einem Malus > 4

Zunächst betrachten wir die Situation, bei der die absolute Differenz der beiden Ameisen auf P_1^{q-1} größer als 4 ist (bzw. eine Ameise, bei der Berechnung der Auswahlwahrscheinlichkeiten für P_1 , einen Malus > 4 hat). Die Auswertung der Lösungsqualität auf dem Teilproblem hat keinen Einfluss mehr darauf, welche der beiden Ameisen die bessere sein wird. Die Wahrscheinlichkeit, dass die nicht mit dem Malus behaftete Ameise auf dem Problem P_1 Element j auf Platz i wählt, ist identisch zur Wahrscheinlichkeit, dass die bessere Ameise Element j auf Platz i wählt. Man beachte, dass wir davon sprechen, dass eine Ameise auf P_1 besser als eine andere Ameise ist, falls die Qualität der Lösung der Ameise auf P_1 inklusive deren Malus besser als die Qualität der Lösung der anderen Ameise auf P_1 inklusive deren Malus ist. Falls eine Ameise einen Malus > 4 hat, findet also keine Konkurrenz zwischen den beiden Ameisen statt. Die Auswahlwahrscheinlichkeiten ergeben sich als eine Summe über alle möglichen Lösungspermutationen. Betrachten wir exemplarisch die Permutation $\pi_1 = (3, 2, 1)$. Wenn wir die Funktion g aus Gleichung (7.6) auf Seite 134 als Matrix darstellen, dann ergibt sich

$$g(\pi_1) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Um die Wahrscheinlichkeit zu berechnen, dass die Lösung π_1 gewählt wird, berechnen wir das Produkt der Wahrscheinlichkeiten der Einzelentscheidungen. Die Wahrscheinlichkeit σ_{π_1} , dass Permutation π_1 gewählt wird, ist somit (vgl. Gleichung (7.3) auf Seite 134)

$$\sigma_{\pi_1} = \frac{0.6}{1} \cdot \frac{0.1}{0.1+0.6} \cdot \frac{0.3}{0.3} \approx 0.086 \quad (7.22)$$

Die Auswahlwahrscheinlichkeiten für die Situation ohne Konkurrenz ergeben sich somit entsprechend Gleichung (7.5) auf Seite 134 wie folgt.

$$\begin{aligned}
\begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} &= \sum_{\pi \in \mathcal{P}_n} \sigma_{\pi} \cdot g(\pi) \\
&= \pi_1 \cdot g(\pi_1) + \dots \\
&\approx 0.086 \cdot \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} + \dots \\
&\approx \begin{pmatrix} 0.1 & 0.3 & 0.6 \\ 0.714 & 0.111 & 0.175 \\ 0.186 & 0.589 & 0.225 \end{pmatrix} \quad (7.23)
\end{aligned}$$

Eine einzelne Ameise wählt auf \mathcal{P}_1 somit beispielsweise mit Wahrscheinlichkeit 0.714 Element 1 auf Platz 2.

Auswahlwahrscheinlichkeiten ohne Malus

Betrachten wir nun die modellierte Situation, dass beide Ameisen auf $q - 1$ Teilinstanzen exakt die gleiche Lösungsqualität erreichen. Wir haben in diesem Fall eine reine Konkurrenzsituation der beiden Ameisen. Zur Berechnung der Auswahlwahrscheinlichkeiten auf dem Problem \mathcal{P}_1 , die sich in diesem Falle aus einer Summe über alle möglichen Kombinationen von Lösungspermutationen ergibt, betrachten wir hier zunächst exemplarisch die Permutationen $\pi_1 = (3, 2, 1)$ und $\pi_2 = (2, 3, 1)$. Beide Lösungspermutationen haben die Lösungsqualität 4. Die Wahrscheinlichkeiten, dass die entsprechenden Lösungen gewählt werden, sind $\sigma_{\pi_1} \approx 0.086$ und $\sigma_{\pi_2} = 0.1$ (vgl. Gleichung (7.22)). Schreiben wir die Funktion g als Matrix (vgl. Gleichung (7.8) auf Seite 135), dann ergeben sich die Auswahlwahrscheinlichkeiten entsprechend Gleichung (7.7) auf Seite 135 als eine Summe über alle möglichen Kombinationen von Permutationen.

$$\begin{aligned}
\begin{pmatrix} \sigma_{11}^{(2)} & \sigma_{12}^{(2)} & \sigma_{13}^{(2)} \\ \sigma_{21}^{(2)} & \sigma_{22}^{(2)} & \sigma_{23}^{(2)} \\ \sigma_{31}^{(2)} & \sigma_{32}^{(2)} & \sigma_{33}^{(2)} \end{pmatrix} &= \sum_{\pi \in \mathcal{P}_n} \sum_{\pi' \in \mathcal{P}_n} \sigma_{\pi} \cdot \sigma_{\pi'} \cdot g(\pi, \pi', i, j) \\
&= \sigma_{\pi_1} \cdot \sigma_{\pi_2} \cdot g(\pi_1, \pi_2, i, j) + \dots
\end{aligned}$$

$$\begin{aligned}
&\approx 0.086 \cdot 0.1 \cdot \underbrace{\begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 \end{pmatrix}}_{g(\pi_1, \pi_2)} + \dots \\
&\approx \begin{pmatrix} 0.175 & 0.405 & 0.420 \\ 0.695 & 0.109 & 0.196 \\ 0.130 & 0.486 & 0.384 \end{pmatrix} \quad (7.24)
\end{aligned}$$

Dies bedeutet beispielsweise, dass die auf dem Problem P_1 bessere der beiden Ameisen mit Wahrscheinlichkeit 0.175 Element 1 auf Platz 1 wählt, wenn keine der Ameisen mit einem Malus behaftet ist.

Da die optimale Lösung immer Element i auf Platz i , $i \in [1, 3]$ wählt, könnte man vermuten, dass die entsprechenden Auswahlwahrscheinlichkeiten auf den optimalen Entscheidungen größer werden, wenn wir die Wahrscheinlichkeiten mit der Situation ohne Konkurrenz des vorherigen Abschnitts vergleichen. Das ist in unserem Beispiel richtig für die Fälle, dass Element 1 auf Platz 1 und Element 3 auf Platz 3 platziert wird. Aber unser Beispiel zeigt auch, dass das im Allgemeinen nicht der Fall sein muss. Denn die Wahrscheinlichkeit bei einem Konkurrenzsystem mit zwei Ameisen beim Problem P_1 und einer Pheromonmatrix, wie sie in Gleichung (7.21) gegeben ist, Element 2 auf Platz 2 zu platzieren, ist $\sigma_{22}^{(2)} = 0.109$ und somit kleiner als die entsprechende Wahrscheinlichkeit $\sigma_{22} = 0.111$ der Situation ohne Konkurrenz.

In Gleichung (7.9) auf Seite 136 wurde ein alternatives Verfahren zur Berechnung der Auswahlwahrscheinlichkeiten für m Ameisen ohne Malus vorgestellt. Die Auswahlwahrscheinlichkeiten ergeben sich nicht als Summe über Kombinationen von Permutationen, sondern als eine Summe über alle möglichen Qualitätsstufen. Zusätzlich benötigt werden die Wahrscheinlichkeiten $\omega_{ij}^{(x)}$. Mit $\omega_{ij}^{(x)}$ wird die Wahrscheinlichkeit bezeichnet, dass eine Ameise, die eine Lösung mit Qualität $x \in \mathcal{C}$ gefunden hat, Element j auf Platz i wählt. Für unser Beispielproblem P_1 haben wir drei Qualitätsstufen. Die drei Lösungen mit der schlechtesten Qualitätsstufe 4 sind die Permutationen $\pi_1 = (3, 2, 1)$, $\pi_2 = (3, 1, 2)$ und $\pi_3 = (2, 3, 1)$, die mit Wahrscheinlichkeiten $\sigma_{\pi_1} = 0.086$, $\sigma_{\pi_2} = 0.514$ und $\sigma_{\pi_3} = 0.1$ gewählt werden. Schreiben wir $\omega_{ij}^{(x)}$ als Matrix, dann ergibt sich mit Gleichung (7.10) auf Seite 136 somit beispielsweise für $x = 4$

$$\omega_{ij}^{(4)} = \begin{pmatrix} 0 & 0.143 & 0.857 \\ 0.735 & 0.122 & 0.143 \\ 0.265 & 0.735 & 0 \end{pmatrix}$$

Die Wahrscheinlichkeit $\xi_4^{(2)}$, dass die beste der beiden Ameisen auf P_1 eine Lösung mit Qualität 4 findet, entspricht der Wahrscheinlichkeit, dass beide Ameisen eine Lösung mit Qualität 4 finden, da 4 die schlechteste Lösungsqualität auf P_1 ist. Die Wahrscheinlichkeit, dass eine Ameise eine Lösung mit Qualität 4 findet, entspricht der Wahrscheinlichkeit, dass eine Ameise die Permutation $(3, 2, 1)$, $(3, 1, 2)$ oder $(2, 3, 1)$ findet. Wir rechnen $\xi_4^{(2)} = (0.6 \cdot \frac{0.1}{0.7} + 0.6 \cdot \frac{0.6}{0.7} + 0.3 \cdot \frac{0.3}{0.9})^2 = 0.49$. Die alternative Variante zur Berechnung der Auswahlwahrscheinlichkeiten führt dann (vgl. Gleichung (7.9) auf Seite 136) zu

$$\begin{aligned} \begin{pmatrix} \sigma_{11}^{(2)} & \sigma_{12}^{(2)} & \sigma_{13}^{(2)} \\ \sigma_{21}^{(2)} & \sigma_{22}^{(2)} & \sigma_{23}^{(2)} \\ \sigma_{31}^{(2)} & \sigma_{32}^{(2)} & \sigma_{33}^{(2)} \end{pmatrix} &= \sum_{x \in \mathcal{C}} \xi_x^{(2)} \cdot \omega_{ij}^{(x)} \\ &= \xi_4^{(2)} \cdot \omega_{ij}^{(4)} + \dots \\ &= 0.49 \cdot \begin{pmatrix} 0 & 0.143 & 0.857 \\ 0.735 & 0.122 & 0.143 \\ 0.265 & 0.735 & 0 \end{pmatrix} + \dots \\ &\approx \begin{pmatrix} 0.175 & 0.405 & 0.420 \\ 0.695 & 0.109 & 0.196 \\ 0.130 & 0.486 & 0.384 \end{pmatrix} \end{aligned}$$

Diese Matrix ist erwartungsgemäß identisch zur Matrix in Gleichung (7.24).

Auswahlwahrscheinlichkeiten bei einem Malus 2 oder 4

Damit wir die Auswahlwahrscheinlichkeiten des homogenen Permutationsproblems P_1^q aus Gleichung (7.20) vollständig berechnen können, müssen wir noch Auswahlwahrscheinlichkeiten auf P_1 für die Fälle berechnen, bei denen eine der beiden Ameisen mit einem Malus 2 oder 4 behaftet ist. Bei der Berechnung der Auswahlwahrscheinlichkeiten auf P_1 ausgehend von einer entsprechenden Malussituation, gehen wir prinzipiell genauso wie bei der alternativen Variante zur Berechnung der Auswahlwahrscheinlichkeiten der Situation ohne Malus vor. Es wird aber nicht Gleichung (7.9), sondern die Gleichung (7.13) auf

Seite 140 zur Berechnung der Auswahlwahrscheinlichkeiten für eine Situation verwendet, bei der eine (oder mehrere) Ameisen mit einem Malus behaftet sind. Wir verwenden wieder die Matrizen $\omega_{ij}^{(x)}$ und berechnen nun beispielhaft die Wahrscheinlichkeit $\xi_4^{(2;2)}$. Dieser Wert gibt die Wahrscheinlichkeit an, dass bei zwei Ameisen die inklusive Malus beste Ameise auf P_1 eine Lösung mit Qualität 4 findet, wobei eine der beiden Ameisen mit einem Malus von 2 behaftet ist. Die Ameise ohne Malus muss eine Permutation mit Lösungsqualität 4 wählen, da sonst die beste der beiden Ameisen eine Lösungsqualität kleiner als 4 hätte. Die Ameise wählt also eine der Permutationen $(3, 2, 1)$, $(3, 1, 2)$ oder $(2, 3, 1)$. Die Wahrscheinlichkeit, dass eine Ameise eine Lösung mit Qualität 4 wählt, ist $\xi_4^{(1)} = 0.6 \cdot \frac{0.1}{0.7} + 0.6 \cdot \frac{0.6}{0.7} + 0.3 \cdot \frac{0.3}{0.9} = 0.7$. Wenn die Ameise mit Malus eine Lösung mit Qualität 0 wählt, dann kann die Ameise mit Qualität 4 nicht mehr die beste sein. Wenn die Ameise mit Malus eine Lösung mit Qualität 4 wählt, dann ist die Ameise ohne Malus immer die beste. Wenn die Ameise mit dem Malus eine Lösung mit Qualität 2 wählt (die Wahrscheinlichkeit hierfür beträgt $\xi_2^{(1)} = 0.3 \cdot \frac{0.6}{0.9} + 0.1 \cdot \frac{0.3}{0.4} = 0.275$), dann ist die resultierende Lösungsqualität inklusive Malus bei beiden Ameisen 4. In 50% aller Fälle ist dann die Ameise mit Malus (die ohne Malus Qualität 2 hat), und in 50% aller Fälle die Ameise ohne Malus (die Qualität 4 hat) die beste Ameise. Die Wahrscheinlichkeit, dass die (inklusive Malus) beste Ameise eine Lösung mit Qualität 4 (ohne Malus) hat, beträgt somit $\xi_4^{(2;2)} = \xi_4^{(1)} \cdot \xi_4^{(1)} + \frac{1}{2} \cdot \xi_4^{(1)} \cdot \xi_4^{(2)} = 0.7 \cdot 0.7 + \frac{1}{2} \cdot 0.7 \cdot 0.275 \approx 0.586$. Für die Situation, bei der eine der beiden Ameisen einen Malus von 2 hat, ergeben sich die Auswahlwahrscheinlichkeiten auf P_1 wie folgt (vgl. Gleichung (7.13) auf Seite 140).

$$\begin{aligned}
 \begin{pmatrix} \sigma_{11}^{(2;2)} & \sigma_{12}^{(2;2)} & \sigma_{13}^{(2;2)} \\ \sigma_{21}^{(2;2)} & \sigma_{22}^{(2;2)} & \sigma_{23}^{(2;2)} \\ \sigma_{31}^{(2;2)} & \sigma_{32}^{(2;2)} & \sigma_{33}^{(2;2)} \end{pmatrix} &= \sum_{x \in \mathcal{C}} \xi_x^{(2;2)} \cdot \omega_{ij}^{(x)} & (7.25) \\
 &= \xi_4^{(2;2)} \cdot \omega_{ij}^{(4)} + \dots \\
 &\approx 0.586 \cdot \begin{pmatrix} 0 & 0.143 & 0.857 \\ 0.735 & 0.122 & 0.143 \\ 0.265 & 0.735 & 0 \end{pmatrix} + \dots \\
 &\approx \begin{pmatrix} 0.146 & 0.351 & 0.503 \\ 0.698 & 0.118 & 0.184 \\ 0.156 & 0.531 & 0.313 \end{pmatrix}
 \end{aligned}$$

Dies bedeutet, dass die inklusive Malus bessere der beiden Ameisen beispiels-

weise Element 2 mit Wahrscheinlichkeit 0.531 auf Platz 3 platziert, wenn eine der beiden Ameisen einen Malus von 2 hat.

Entsprechend ergibt sich für den Fall, dass eine Ameise einen Malus von 4 hat

$$\begin{pmatrix} \sigma_{11}^{(2;4)} & \sigma_{12}^{(2;4)} & \sigma_{13}^{(2;4)} \\ \sigma_{21}^{(2;4)} & \sigma_{22}^{(2;4)} & \sigma_{23}^{(2;4)} \\ \sigma_{31}^{(2;4)} & \sigma_{32}^{(2;4)} & \sigma_{33}^{(2;4)} \end{pmatrix} \approx \begin{pmatrix} 0.109 & 0.299 & 0.593 \\ 0.708 & 0.118 & 0.174 \\ 0.183 & 0.583 & 0.234 \end{pmatrix}$$

Falls eine der beiden Ameisen einen Malus hat, dann liegen die Auswahlwahrscheinlichkeiten normalerweise zwischen den Auswahlwahrscheinlichkeiten, die sich bei den Situationen ohne Konkurrenz und einem Konkurrenzsystem ohne Malusse ergeben. Der Fall Element 2 auf Platz 2 zu platzieren, ist jedoch wieder ein Gegenbeispiel, denn es gilt $\sigma_{22}^{(2)} < \sigma_{22} < \sigma_{22}^{(2;2)}$.

Obwohl das, wie wir im Beispiel gesehen haben, nicht allgemein gültig ist, kann beobachtet werden, dass die Auswahlwahrscheinlichkeiten für einen größer werdenden Einfluss des Malusses immer mehr den Auswahlwahrscheinlichkeiten der Situation ohne Konkurrenz entsprechen. Die Auswahlwahrscheinlichkeiten des Konkurrenzsystems werden also im Allgemeinen in eine ungünstige Richtung verzerrt.

Auswahlwahrscheinlichkeiten für P_1^q

In diesem Abschnitt berechnen wir die Auswahlwahrscheinlichkeiten auf dem homogenen Permutationsproblem P_1^q . Man beachte, dass wir die Wahrscheinlichkeiten, dass die auf P_1^q beste von m Ameisen eine Lösung mit Qualität x auf einem Problem P_1 des Problems P_1^q findet, mit $\xi_x^{(m)}$ bezeichnen. Die Wahrscheinlichkeiten, dass die beste von m Ameisen eine Lösung mit Qualität x auf einem Teilproblem P_1 findet (nicht als elementares Teilproblem des Problems P_1^q gesehen), wird mit $\xi_x^{l(m)}$ bezeichnet wird. $\xi_x^{l(1)}$ ist Wahrscheinlichkeit, dass eine Ameise auf einem elementaren Teilproblem eine Lösung mit Qualität x findet (vgl. Gleichung (7.17) auf Seite 144). Die Wahrscheinlichkeiten, dass eine Ameise eine Lösung des Problems P_1 mit Kosten 0, 2 oder 4 findet, sind $\xi_0^{l(1)} = 0.1 \cdot (0.1/0.4) = 0.025$, $\xi_2^{l(1)} = 0.3 \cdot (0.6/0.9) + 0.1 \cdot (0.3/0.4) = 0.275$, und entsprechend $\xi_4^{l(1)} = 0.6 \cdot (0.6/0.7) + 0.6 \cdot (0.1/0.7) + 0.3 \cdot (0.3/0.9) = 0.7$.

Betrachten wir exemplarisch das homogene Problem P_1^3 . Die möglichen Unterschiede der Lösungsqualitäten zweier Ameisen auf dem Problem P_1 sind $-4, -2, 0, 2, 4$, sodass $d_{\max} = 4$ ist. Die Wahrscheinlichkeiten, dass zwei Ameisen auf dem Problem P_1 Lösungen finden, deren Qualitätsunterschied $-4, -2,$

0, 2 oder 4 ist, lassen sich wie folgt berechnen: $\psi'_0 = 0.025^2 + 0.275^2 + 0.7^2 \approx 0.566$, $\psi'_{-2} = \psi'_2 \approx 0.199$, $\psi'_{-4} = \psi'_4 \approx 0.018$. Die Menge I_0 (vgl. Gleichung (7.18) auf Seite 146) ist

$$I_0 = \{(0, 0, 0, 0, 2, 0, 0, 0, 0), (0, 0, 1, 0, 0, 0, 1, 0, 0), (1, 0, 0, 0, 0, 0, 0, 0, 1)\}$$

Die Mengen I_2 und I_4 werden analog definiert. Die Wahrscheinlichkeit, dass zwei Ameisen eine Lösung mit gleicher Qualität für P_1^2 finden ist

$$\begin{aligned}\phi_0^{(2)} &= \psi_0 \\ &= (\psi'_0)^2 + \psi'_{-2}\psi'_2 + \psi'_2\psi'_{-2} + \psi'_{-4}\psi'_4 + \psi'_4\psi'_{-4} \\ &\approx 0.401\end{aligned}$$

und analog $\phi_2^{(2)} = 2\psi_2 \approx 0.466$, $\phi_4^{(2)} = 2\psi_4 \approx 0.119$ und $\phi_{d>4}^{(2)} = 2\psi_{>4} \approx 0.015$.

Die Matrix der Auswahlwahrscheinlichkeiten für das Problem P_1^3 berechnet sich dann als eine gewichtete Summe der Matrizen der Auswahlwahrscheinlichkeiten bei unterschiedlichen Malussituationen, wobei die Werte $\phi_0^{(2)}$, $\phi_2^{(2)}$, $\phi_4^{(2)}$ und $\phi_{d>4}^{(2)}$ die Gewichte der Matrizen sind (vgl. Gleichung (7.20) auf Seite 146). Wir rechnen somit für das r -te elementare Teilproblem $r \in [1, q]$ und $i, j \in [1, 3]$

$$\sigma_{3(r-1)+i, 3(r-1)+j}^{(2)} = \phi_{d>4}^{(2)} \cdot \sigma'_{ij} + \sum_{d=2,4} \phi_d^{(2)} \cdot \sigma_{ij}^{/(2;d)} + \phi_0^{(2)} \cdot \sigma_{ij}^{(2)}$$

Somit gilt für $r \in [1, 3]$

$$\begin{aligned}&\begin{pmatrix} \sigma_{3(r-1)+1, 3(r-1)+1}^{(2)} & \sigma_{3(r-1)+1, 3(r-1)+2}^{(2)} & \sigma_{3(r-1)+1, 3(r-1)+3}^{(2)} \\ \sigma_{3(r-1)+2, 3(r-1)+1}^{(2)} & \sigma_{3(r-1)+2, 3(r-1)+2}^{(2)} & \sigma_{3(r-1)+2, 3(r-1)+3}^{(2)} \\ \sigma_{3(r-1)+3, 3(r-1)+1}^{(2)} & \sigma_{3(r-1)+3, 3(r-1)+2}^{(2)} & \sigma_{3(r-1)+3, 3(r-1)+3}^{(2)} \end{pmatrix} \approx \\ &\approx 0.015 \cdot \begin{pmatrix} 0.1 & 0.3 & 0.6 \\ 0.714 & 0.111 & 0.175 \\ 0.186 & 0.589 & 0.225 \end{pmatrix} + \dots \\ &\approx \begin{pmatrix} 0.152 & 0.366 & 0.482 \\ 0.699 & 0.114 & 0.187 \\ 0.149 & 0.520 & 0.331 \end{pmatrix} \quad (7.26)\end{aligned}$$

Auf dem homogenen Permutationsproblem P_1^3 der Größe 9×9 wird somit, bei der durch (7.21) gegebenen Pheromonmatrixinitialisierung und einem System mit zwei Ameisen, die bessere der beiden beispielsweise mit Wahrscheinlichkeit 0.366 Element 2 auf Platz 1 (oder Element 5 auf Platz 4, oder Element 8 auf Platz 7) platzieren.

Man beachte, dass sich die in Gleichung (7.26) berechneten Auswahlwahrscheinlichkeiten entsprechend der Gleichung (7.15) als eine gewichtete Summe der Matrizen $\omega_{ij}^{(x)}$, $x \in \{0, 2, 4\}$ darstellen lassen, und die Veränderung der Problemgröße durch die Veränderung der Anzahl der elementaren Teilprobleme q nur die Gewichte der Matrizen $\omega_{ij}^{(x)}$ verändert.

Schließlich berechnen wir beispielhaft die Wahrscheinlichkeit $\xi_0^{(2)}$, dass die auf P_1^3 bessere der beiden Ameisen eine Lösung mit Qualität 0 auf einem elementaren Teilproblem P_1 des Problems P_1^3 findet, wie folgt.

$$\xi_0^{(2)} \approx 0.152 \cdot \frac{0.114}{0.114 + 0.187} \approx 0.058$$

Diese Wahrscheinlichkeit ist erwartungsgemäß kleiner als die entsprechende Wahrscheinlichkeit, dass die bessere von zwei Ameisen auf dem Problem P_1 eine Lösung mit Qualität 0 findet, welche sich durch $0.175 \cdot \frac{0.109}{0.109 + 0.196} = 0.063$ berechnen lässt (vgl. Matrix (7.24) auf Seite 149).

7.5 Fixpunktanalyse

Die Pheromonwerte eines ACO-Algorithmus spiegeln die Häufigkeit von Entscheidungen wider, die zu guten Lösungen führen. Daher ist eine wünschenswerte Eigenschaft von ACO-Algorithmen, dass die Wahrscheinlichkeiten, die während eines Laufs zur Lösungskonstruktion verwendet werden, den zugehörigen Pheromonwerten entsprechen. Wie wir bei den empirischen Untersuchungen der Verzerrung der Auswahlwahrscheinlichkeiten gesehen haben, ist dies oft nicht der Fall, da die Einzelentscheidungen einer Ameise bei der Konstruktion einer Lösung nicht voneinander unabhängig sind. In diesem Abschnitt wird diese Verzerrung für die lokale Pheromonauswertung und Vorwärts-Ameisen genauer analysiert.

Eine Pheromonmatrix mit der Eigenschaft, dass jeder Pheromonwert τ_{ij} genau der Auswahlwahrscheinlichkeit σ_{ij} , also der Wahrscheinlichkeit mit einer einzelnen Ameise Element j auf Platz i zu wählen, entspricht, nennen wir Fixpunktpheromonmatrix oder einfach *Fixpunktmatrix*. Einzelne Zeilen der Pheromonmatrix (Fixpunktmatrix) nennen wir auch *Pheromonvektoren*

(*Fixpunktvektoren*). Eine Fixpunktmatrix verändert sich beim ACO-Modell nicht, wenn das Modell mit nur einer Ameise verwendet wird. Eine Fixpunktmatrix kann sich nur verändern, wenn $m \geq 2$ Ameisen in einer Iteration verwendet werden. Dies ist eine wünschenswerte Eigenschaft, da es bedeutet, dass die Veränderung der Pheromonmatrix auf der Konkurrenz der Ameisen basiert und nicht auf unerwünschten Verzerrungseffekten. Es taucht die Frage auf, welche Matrizen Fixpunktmatrizen sind, und wie viele solcher Matrizen existieren. Als Beispiel betrachten wir ein Problem der Größe $n = 3$ mit folgender Pheromonmatrix. Man beachte, dass die Zeilen- und Spaltensummen auf 1 normiert sind.

$$\begin{pmatrix} \tau_{11} & \tau_{12} & 1 - \tau_{11} - \tau_{12} \\ \tau_{21} & \tau_{22} & 1 - \tau_{21} - \tau_{22} \\ 1 - \tau_{11} - \tau_{21} & 1 - \tau_{12} - \tau_{22} & \tau_{11} + \tau_{12} + \tau_{21} + \tau_{22} - 1 \end{pmatrix} \quad (7.27)$$

Wir wollen die Fixpunktmatrizen für die Matrix aus (7.27) berechnen, d.h. Matrizen, bei denen für alle $i, j \in [1, 3]$ gilt $\tau_{ij} = \sigma_{ij}$. Offensichtlich entsprechen bei der Verwendung von Vorwärts-Ameisen die Auswahlwahrscheinlichkeiten der ersten Zeile für eine einzelne immer den zugehörigen Pheromonwerten der Pheromonmatrix (vgl. auch Matrix (7.23) auf Seite 148). Da die gesamte Pheromonmatrix durch die vier Werte τ_{11} , τ_{12} , τ_{21} und τ_{22} festgelegt wird, müssen wir nur die Auswahlwahrscheinlichkeiten für $m = 1$ des ersten und zweiten Elements in der zweiten Zeile bestimmen. Diese sind

$$\sigma_{21} := \frac{\tau_{12} \tau_{21}}{1 - \tau_{22}} + \frac{(1 - \tau_{11} - \tau_{12}) \tau_{21}}{\tau_{21} + \tau_{22}}$$

und

$$\sigma_{22} := \frac{\tau_{11} \tau_{22}}{1 - \tau_{21}} + \frac{(1 - \tau_{11} - \tau_{12}) \tau_{22}}{\tau_{21} + \tau_{22}}$$

Die Lösung des Gleichungssystems

$$\begin{aligned} \sigma_{21} - \tau_{21} &= 0, \\ \sigma_{22} - \tau_{22} &= 0 \end{aligned}$$

liefert uns die Fixpunktvektoren der zweiten Zeile. Die vier möglichen Lösungen sind

$$(1) \quad \tau_{21} = 0, \quad \tau_{22} = \frac{\tau_{11} + \tau_{12} - 1}{\tau_{11} - 1}$$

$$(2) \tau_{21} = \frac{\tau_{11} + \tau_{12} - 1}{\tau_{12} - 1}, \tau_{22} = 0$$

$$(3) \tau_{21} = \frac{\tau_{12}}{\tau_{11} + \tau_{12}}, \tau_{22} = \frac{\tau_{11}}{\tau_{11} + \tau_{12}}$$

$$(4) \tau_{21} = 1 - 2 \cdot \tau_{11}, \tau_{22} = 1 - 2 \cdot \tau_{12}$$

Mit Hilfe der Eigenwerte der Jacobimatrix der Vektorfunktion $[f_1, f_2] = [\sigma_{21} - \tau_{21}, \sigma_{22} - \tau_{22}]$ lässt sich die Stabilität der vier Fixpunktvektoren bestimmen. Für den Fall (1) ergibt sich beispielsweise die Jacobimatrix

$$\left(\begin{array}{cc} \frac{\partial f_1}{\partial \tau_{21}} & \frac{\partial f_1}{\partial \tau_{22}} \\ \frac{\partial f_2}{\partial \tau_{21}} & \frac{\partial f_2}{\partial \tau_{22}} \end{array} \right) \Bigg|_{\substack{\tau_{21}=0 \\ \tau_{22}=\frac{\tau_{11}+\tau_{12}-1}{\tau_{11}-1}}} = \left(\begin{array}{cc} 1 - 2\tau_{11} & 0 \\ \frac{2\tau_{11}^2 - 3\tau_{11} + \tau_{11}\tau_{12} + 1}{\tau_{11} - 1} & \tau_{11} - 1 \end{array} \right) \quad (7.28)$$

mit den Eigenwerten $1 - 2 \cdot \tau_{11}$ und $\tau_{11} - 1$. Ein Fixpunkt ist genau dann stabil, wenn alle Eigenwerte der Jacobimatrix negativ sind. Der Fall (1) liefert also genau dann einen stabilen Fixpunktvektor, wenn $\tau_{11} > 0.5$ ist. Die Analyse der Jacobimatrizen ergibt, dass die Fälle (1), (2) und (3) symmetrisch sind: für $\tau_{12} > 0.5$ ist der durch (2) beschriebene Fixpunktvektor stabil und für $1 - \tau_{11} - \tau_{12} > 0.5$ ist der durch (3) beschriebene Fixpunktvektor stabil. Die technisch ein wenig aufwendigere Analyse für den Fall (4) ergibt, dass der durch (4) beschriebene Fixpunktvektor genau in allen anderen Fällen stabil ist. Wir haben somit für jedes mögliche Paar (τ_{11}, τ_{12}) genau einen stabilen Fixpunktvektor der zweiten Zeile, der als Attraktor im Bereich der zulässigen Pheromonwerte wirkt. Die drei anderen jeweils instabilen Fixpunktvektoren liegen teilweise außerhalb des zulässigen Parameterbereichs der Pheromonwerte, der durch $\tau_{21}, \tau_{22}, \tau_{23} \in (0, 1)$, $\tau_{21} + \tau_{22} + \tau_{23} = 1$ beschrieben ist.

Für die Darstellung von Tripeln (also Fixpunktvektoren oder Pheromonvektoren der Größe 3), wählen wir die in Abbildung 7.3 veranschaulichte Methode. In einem gleichschenkligen Dreieck mit Höhe 1 beschreiben die Abstände zu den Seiten des Dreiecks die entsprechenden (Pheromon-)Werte. Es sei z.B. der Pheromonvektor $(\tau_{i1}, \tau_{i2}, \tau_{i3})$ gegeben. Der minimale Abstand zur rechten Linie entspricht τ_{i1} . Der minimale Abstand zur unteren Linie entspricht τ_{i2} . Der minimale Abstand zur linken Linie entspricht τ_{i3} . Ein Punkt im Dreieck gibt einen Pheromonvektor an. Eine Linie innerhalb des Dreiecks beschreibt die dynamische Veränderung von Pheromonvektoren.

Die Fixpunktanalyse hat gezeigt, dass die Fixpunktvektoren der zweiten (und dritten) Zeile nur von den Pheromonwerten der ersten Zeile abhängen. Abbildung 7.4 zeigt für unterschiedliche Pheromonvektoren $(\tau_{11}, \tau_{12}, \tau_{13})$ die

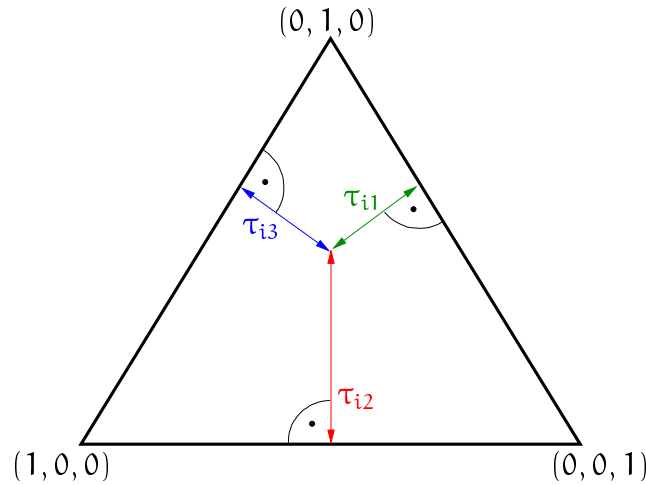


Abbildung 7.3: Darstellung des Tripels $(\tau_{i1}, \tau_{i2}, \tau_{i3})$ durch einen Punkt im dargestellten Dreieck: der kürzeste Abstand zur rechten Linie gibt den Wert τ_{i1} an; der kürzeste Abstand zur unteren Linie gibt den Wert τ_{i2} an; der kürzeste Abstand zur linken Linie gibt den Wert τ_{i3} an; die Höhe des gleichschenkligen Dreiecks beträgt 1; jeder Punkt in der dargestellten Ebene wird durch ein Tripel von Werten definiert; für alle Punkte innerhalb des Dreiecks gilt, dass diese Werte positiv sind; die Summe der Werte ergibt immer 1; für Punkte außerhalb des Dreiecks gilt, dass ein oder mehrere Werte im Tripel negativ sind; die Ecken des Dreiecks entsprechen den dort angegebenen Tripel

entsprechenden Fixpunktvektoren der zweiten Zeile. Man beachte, dass die erste Zeile der Fixpunktmatrix immer der ersten Zeile der Pheromonmatrix entspricht, und dass die Zeilen- und Spaltensummen immer 1 ergeben. Wir sprechen von einer stabilen Fixpunktmatrix, wenn der Fixpunktvektor der zweiten (und somit auch der dritten) Zeile stabil ist. Aus Symmetriegründen betrachten wir in Abbildung 7.4 nur die Fälle, für die $\tau_{11} \geq 1/3$ gilt. Der zu den Pheromonwerten (oder auch Auswahlwahrscheinlichkeiten, da $m = 1$) $(\tau_{11}, \tau_{12}, \tau_{13}) = (1/3, 1/3, 1/3)$ gehörige stabile Fixpunktvektor der zweiten Zeile des ACO-Modells liegt genau bei $(1/3, 1/3, 1/3)$. Eine solche Matrix wird typischerweise als Initialisierungsmatrix für ACO-Algorithmen verwendet. Die Veränderungen von Pheromonwerten der jeweils ersten Iteration des ACO-Modells bei dieser Initialisierung basiert somit immer auf Konkurrenz, und nicht auf einer Verzerrung der Auswahlwahrscheinlichkeiten aufgrund eines Fixpunktvektors, der als Attraktor für Pheromonvektoren wirkt. Die drei zugehörigen instabilen Fixpunktvektoren der zweiten Zeile $(1/2, 1/2, 0)$,

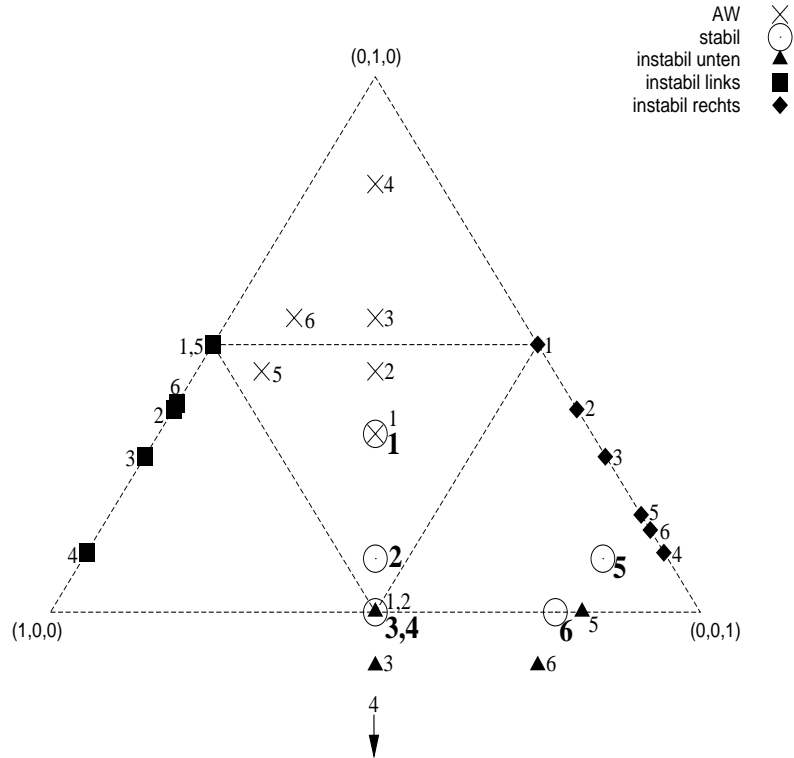


Abbildung 7.4: Stabile und instabile Fixpunktvektoren der zweiten Zeile für 6 unterschiedliche Pheromonvektoren $(\tau_{11}, \tau_{12}, \tau_{13})$; dargestellt sind die Pheromonvektoren $(\tau_{11}, \tau_{12}, \tau_{13})$ der ersten Zeile, die identisch zu den Auswahlwahrscheinlichkeiten (AW) der ersten Zeile sind, und die sich für jeden Pheromonvektor ergebenden 4 Fixpunktvektoren der zweiten Zeile, für die $(\tau_{21}, \tau_{22}, \tau_{23}) = (\sigma_{21}, \sigma_{22}, \sigma_{23})$ gilt; der stabile Fixpunkt zum entsprechenden durch ein Kreuz dargestellten Pheromonvektor (AW) ist durch einen Kreis dargestellt; Dreieck, Raute und Quadrat entsprechen den 3 instabilen Fixpunktvektoren; das innere Dreieck enthält alle Punkte, für die alle Werte des Tripels $< 1/2$ sind; die Nummern an den Symbolen bezeichnen die AWs und die dazugehörigen Fixpunktvektoren

$(0, 1/2, 1/2)$ und $(1/2, 0, 1/2)$ liegen an den Ecken des inneren Dreiecks, welches genau die Punkte beschreibt, für die einer der Werte des dargestellten Tripels gleich $1/2$ ist.

Für andere Pheromonvektoren $(\tau_{11}, \tau_{12}, \tau_{13})$ haben die Pheromonwerte der ersten Zeile und die Werte des zugehörigen stabilen Fixpunktvektors der zweiten Zeile ein gegensätzliches Verhalten (vgl. Abbildung 7.4). Für einen wach-

senden Wert τ_{1j} wird der zugehörige Wert des stabilen Fixpunktvektors der zweiten Zeile kleiner. Für einen Pheromonvektor $(\tau_{11}, \tau_{12}, \tau_{13})$ mit $\tau_{12} > 0.5$ liegt der zugehörige Wert des stabilen Fixpunktvektors auf der unteren Kante des Dreiecks.

Wenn der Pheromonvektor $(\tau_{11}, \tau_{12}, \tau_{13})$ im inneren Dreieck der Abbildung 7.4 liegt, dann befindet sich auf jeder Seite des äußeren Dreiecks ein instabiler Fixpunktvektor der zweiten Zeile. Für $\tau_{12} > 1/2$ liegt einer der instabilen Fixpunktvektoren außerhalb des Bereichs der zulässigen Pheromonvektoren und hat einen Wert von $\tau_{22} < 0$.

Das Richtungsfeld für Veränderungen des Pheromonvektors $(\tau_{21}, \tau_{22}, \tau_{23})$ für den Fall $(\tau_{11}, \tau_{12}, \tau_{13}) = (1/3, 1/3, 1/3)$ ist in Abbildung 7.5 dargestellt. Der Punkt $(1/3, 1/3, 1/3)$ ist ein Attraktor. Bei einem ACO-Modell ohne Konkurrenz (also $m = 1$), strebt die zweite Zeile der Pheromonmatrix wie im Richtungsfeld dargestellt, zum stabilen Fixpunktvektor $(1/3, 1/3, 1/3)$. Der linke Teil der Grafik zeigt den Fall der lokalen Pheromonauswertung, der rechte Teil der Grafik zeigt das entsprechende Richtungsfeld für eine Summen-Pheromonauswertung entsprechend der Gleichung (7.4). Im hier dargestellten Fall $(\tau_{11}, \tau_{12}, \tau_{13}) = (1/3, 1/3, 1/3)$ sind die Richtungsfelder hinsichtlich einer Rotation um 60 Grad um den Fixpunkt $(1/3, 1/3, 1/3)$ symmetrisch. Eine interessante Beobachtung bei der lokalen Pheromonauswertung ist, dass es Punkte $(\tau_{21}, \tau_{22}, \tau_{23})$ gibt, bei denen für ein $i \in [1, 3]$ ein Wert $\tau_{2i} > 1/3$ ist, und dieser Wert nicht auf direktem Wege zum stabilen Fixpunktvektor strebt, sondern sogar noch wächst. Dies zeigt sehr deutlich, dass Effekte der Verzerrung schon bei sehr kleinen Problemen mit nur einer Ameise pro Iteration ein nicht offensichtliches Verhalten zeigen können. Für die Summen-Pheromonauswertung zeigen die Vektoren direkt in Richtung des stabilen Fixpunktes. Das ist ein Hinweis darauf, dass die Summen-Pheromonauswertung, wie wir in den vorherigen Kapiteln empirisch gesehen haben, eine geeignete Methode für eine bestimmte Klasse von Permutationsproblemen ist.

Die Fixpunktanalyse für dieses kleine System hat gezeigt, dass es für $m = 1$ Ameise immer nur genau eine stabile Fixpunktmatrix gibt. Da sich 1.) die Berechnung der Auswahlwahrscheinlichkeiten eines beschränkten Permutationsproblems P^q als gewichtete Summe aus Auswahlwahrscheinlichkeiten auf den Teilproblemen beschreiben lässt, und sich, wie wir auch in den nächsten Abschnitten sehen werden, 2.) im Allgemeinen mit großer Wahrscheinlichkeit auf P^{q-1} eine Malussituation mit großem Malus ergibt, ist die Fixpunktanalyse ein Hinweis darauf, dass die Verzerrung der Auswahlwahrscheinlichkeiten ein wichtiger Faktor ist, um das Optimierungsverhalten von ACO-Algorithmen

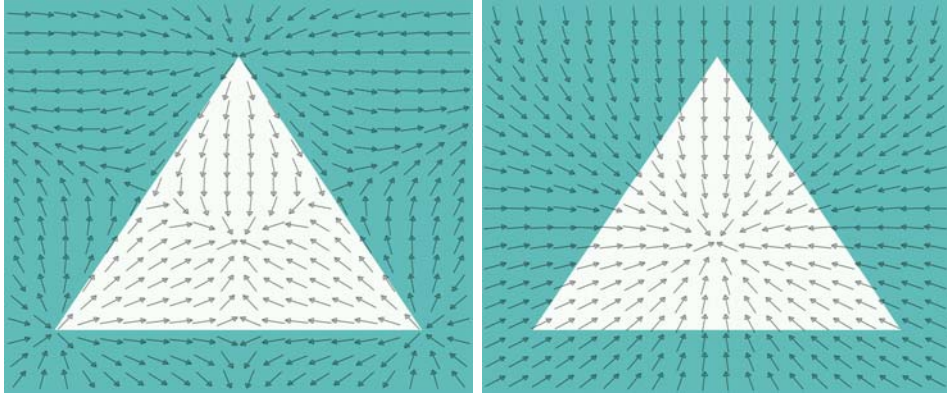


Abbildung 7.5: Richtungsfeld zur Darstellung der Veränderung der zweiten Zeile $(\tau_{21}, \tau_{22}, \tau_{23})$ für das ACO-Modell mit einer Ameise und $\tau_{11} = \tau_{12} = \tau_{13} = 1/3$; links ist das Feld für die lokale Pheromonauswertung dargestellt, rechts das Feld für die Summen-Pheromonauswertung; zulässige Pheromonvektoren liegen im weißen Gebiet

zu verstehen. Denn mit großer Wahrscheinlichkeit werden die Auswahlwahrscheinlichkeiten eines beschränkten Permutationsproblems P^q im Modell auch mit $m \geq 2$ Ameisen in Richtung des stabilen Fixpunktvektors gelenkt.

7.6 Dynamisches Verhalten

In diesem Abschnitt untersuchen wir das dynamische Verhalten des ACO-Modells sowohl auf kleinen (allgemeinen) Permutationsproblemen als auch auf größeren eingeschränkten Permutationsproblemen. Ein zentraler Punkt der Untersuchung wird die Frage sein, ob die Verzerrung der Auswahlwahrscheinlichkeiten, welche im vorherigen Abschnitt beobachtet wurde, einen signifikanten Einfluss auf das dynamische Verhalten des ACO-Modells mit Konkurrenz (d.h. mehr als eine Ameise pro Iteration erzeugen Lösungen) haben kann. Als Beispielprobleme verwenden wir eingeschränkte homogene Permutationsprobleme unterschiedlicher Größen mit dem Problem P_1 der Größe $n_0 = 3$ als elementares Teilproblem. Die Kostenmatrix für dieses Problem ist in Gleichung (7.11) auf Seite 137 gegeben.

Zur Darstellung des dynamischen Verhaltens eines Pheromonvektors verwenden wir die in Abbildung 7.3 auf Seite 157 dargestellte Methode zur Darstellung von Pheromonvektoren in einem gleichschenkligen Dreieck. Einzelne Pheromonvektoren entsprechen Punkten im Dreieck. Die dynamische Verän-

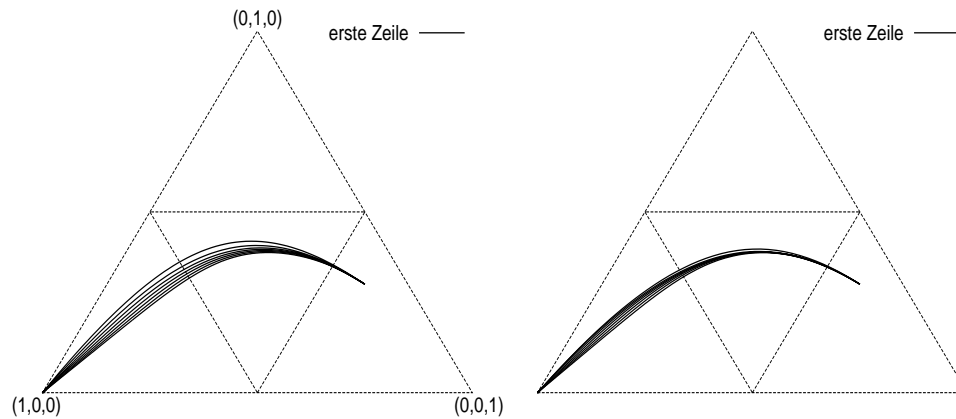


Abbildung 7.6: ACO-Modell für P_1^q ; $q = 2, 4, 8, 16, 32, 64, 128$; links 2 Ameisen, rechts 3 Ameisen; dargestellt ist die Dynamik der Werte $\tau_{11}, \tau_{12}, \tau_{13}$; (Werte der ersten Zeile der zugehörigen Fixpunktmatrix sind identisch); die Initialisierung ist $(\tau_{11}, \tau_{12}, \tau_{13}) = (0.1, 0.3, 0.6)$; Kurven, die weiter unten liegen, entsprechen größeren Werten für q

derung von Pheromonvektoren über die Zeit hinweg entsprechen Linien im Dreieck.

Die Abbildungen 7.6-7.9 zeigen das dynamische Verhalten des ACO-Modells für homogene Permutationsprobleme P_1^q mit unterschiedlichen Werten für q und $m \in \{2, 3\}$ Ameisen. Die Verdunstungsrate ρ ist 0.1 und die Pheromoninitialisierung wurde wie in Gleichung (7.21) auf Seite 147 gewählt, d.h. die initial gewählte Pheromonmatrix wird durch die Werte $\tau_{11} = 0.1$, $\tau_{12} = 0.3$, $\tau_{21} = 0.6$ und $\tau_{22} = 0.1$ definiert. Jede Abbildung zeigt mehrere Kurven für unterschiedlich viele Teilprobleme. Die gewählte Anzahl elementarer Teilprobleme ist $q = 2, 4, 8, 16, 32, 64$ und 128. Zusätzlich zeigt jede der Abbildungen 7.6-7.8 die zugehörige Position des stabilen Fixpunktvektors des ACO-Modells mit einer Ameise pro Generation.

Im Gegensatz zur Situation mit nur einer Ameise pro Iteration, bei der die Pheromonwerte gegen den stabilen Fixpunktvektor konvergieren, strebt das ACO-Modell mit Konkurrenz zur optimalen Lösung. Jedoch folgen die Pheromonvektoren im Allgemeinen nicht dem direkten Weg zur optimalen Lösung. Die aktuelle Position des Fixpunktvektors hat insbesondere für große q einen starken Einfluss auf das System. Man beachte, dass die erste Zeile der Pheromonmatrix immer der ersten Zeile der Fixpunktmatrix entspricht. Daher basiert die Veränderung der Pheromonwerte der ersten Zeile immer auf

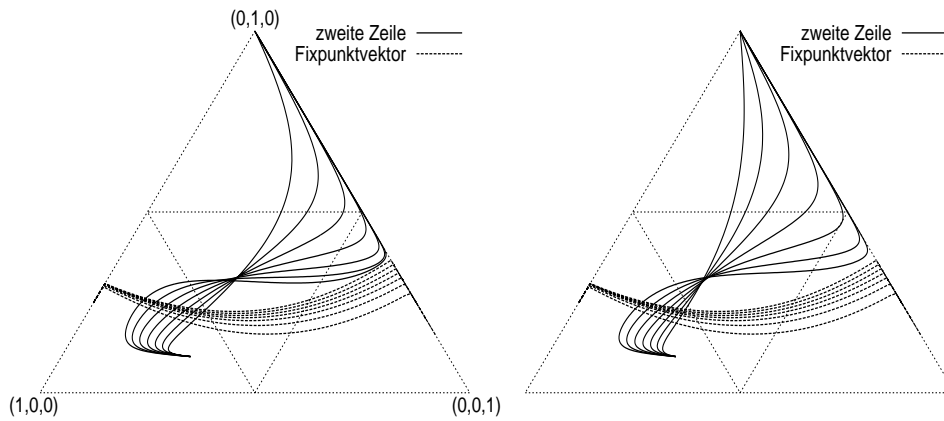


Abbildung 7.7: ACO-Modell für P_1^q ; $q = 2, 4, 8, 16, 32, 64, 128$; links 2 Ameisen, rechts 3 Ameisen; dargestellt ist die Dynamik der Werte $\tau_{21}, \tau_{22}, \tau_{23}$ und die zweite Zeile der zugehörigen stabilen Fixpunktmatrix; die Initialisierung ist $(\tau_{11}, \tau_{12}, \tau_{13}) = (0.6, 0.1, 0.3)$; der Fixpunktvektor startet bei $(0.75, 0.25, 0)$; Pheromonvektorkurven (Fixpunktvektorkurven), die in der rechten Hälfte des Dreiecks weiter unten (oben) liegen, entsprechen größeren Werten für q

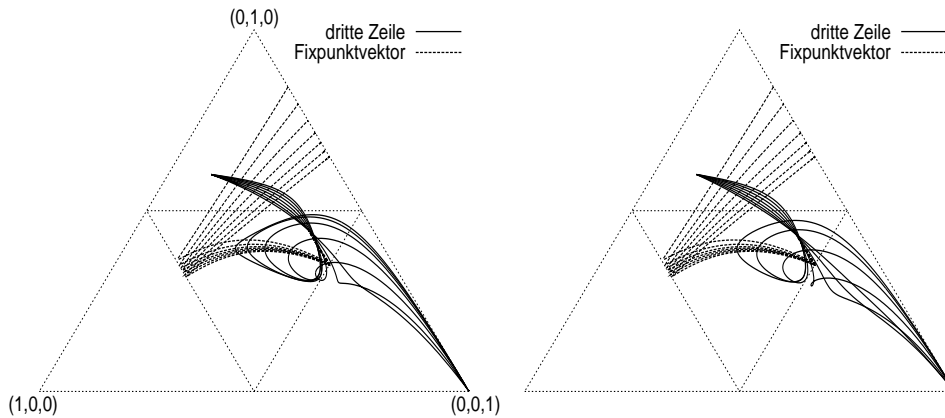


Abbildung 7.8: ACO-Modell für P_1^q ; $q = 2, 4, 8, 16, 32, 64, 128$; links 2 Ameisen, rechts 3 Ameisen; dargestellt ist die Dynamik der Werte $\tau_{31}, \tau_{32}, \tau_{33}$ und die dritte Zeile der zugehörigen stabilen Fixpunktmatrix; die Initialisierung ist $(\tau_{11}, \tau_{12}, \tau_{13}) = (0.3, 0.6, 0.1)$; der Fixpunktvektor startet bei $(0.15, 0.45, 0.4)$; Kurven, die in der oberen Hälfte des Dreiecks weiter oben liegen, entsprechen größeren Werten für q

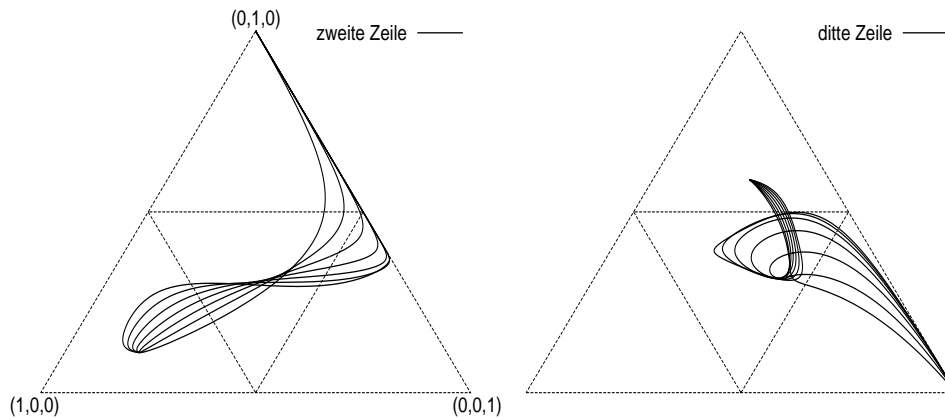


Abbildung 7.9: ACO-Modell für P_1^q ; $q = 2, 4, 8, 16, 32, 64, 128$ und $m = 2$ Ameisen; dargestellt ist die Dynamik der Auswahlwahrscheinlichkeiten $\sigma_{i1}, \sigma_{i2}, \sigma_{i3}$ in der zweiten Zeile (links) und in der dritten Zeile (rechts); die den entsprechenden Pheromonvektoren $(\tau_{21}, \tau_{22}, \tau_{23}) = (0.6, 0.1, 0.3)$ und $(\tau_{31}, \tau_{32}, \tau_{33}) = (0.3, 0.6, 0.1)$ zugehörigen Startpunkte der Auswahlwahrscheinlichkeiten sind $(\sigma_{21}, \sigma_{22}, \sigma_{23}) = (5/7, 31/280, 7/40)$ und $(\sigma_{31}, \sigma_{32}, \sigma_{33}) = (13/70, 33/56, 9/40)$

reiner Konkurrenz. Die Verzerrung der Auswahlwahrscheinlichkeiten aufgrund des als Attraktor wirkenden Fixpunktvektors hat keinen Einfluss auf die erste Zeile. Dies ist auch der Grund, wieso die Pheromonvektoren in Abbildung 7.6 die Optimallösung nahezu auf direktem Weg erreichen. Zwischen dem Modell mit zwei und drei Ameisen besteht in der ersten Zeile nahezu kein Unterschied.

Dies ist deutlich anders für die Pheromonvektoren der zweiten Zeile (vgl. Abbildung 7.7). In der dritten Zeile ist die Verzerrung am deutlichsten zu erkennen (vgl. Abbildung 7.8). In der mittleren Phase des Laufs bewegt sich der Pheromonvektor der zweiten und insbesondere der dritten Zeile für große q weg von der optimalen Lösung. Jeder der zurückgelegten Pfade der Fixpunktvektoren in den Abbildungen 7.7 und 7.8 besteht aus drei Teilpfaden. Diese Dreiteilung liegt daran, dass der Pheromonvektor drei unterschiedliche Gebiete durchläuft, die sich aufgrund der Aufteilung in die kleineren Dreiecke ergeben. Jedes der kleinen Dreiecke entspricht genau einem der vier Fälle bei der Fixpunktanalyse. Je nach Position des Pheromonvektors der ersten Zeile, wird der stabile Fixpunktvektor der zweiten und dritten Zeile entsprechend einem der vier Fälle berechnet. In Abbildung 7.7 liegen der erste und dritte

Teil des Fixpunktpfads genau auf den Grenzen des äußeren Dreiecks. In Abbildung 7.8 sieht man, dass der dreigeteilte Fixpunktpfad an den Seiten des inneren Dreiecks zweimal geknickt wird.

In Abbildung 7.8 haben fast alle Pfade der Pheromonvektoren (alle mit $q \geq 8$ für $m = 2$ Ameisen und alle mit $q \geq 16$ für $m = 3$ Ameisen) eine Schleife, welche durch die Richtungsumkehrung des Fixpunktvektors verursacht ist. Für $m = 3$ Ameisen ist die Konkurrenz stärker als für $m = 2$ Ameisen, daher bewegt sich der Pheromonvektor stärker entlang des direkten Wegs zur optimalen Lösung hin. Beispielsweise bewegen sich in Abbildung 7.7 die Kurven für $m = 2$ Ameisen stärker nach rechts als für $m = 3$ Ameisen. In Abbildung 7.8 hat die Kurve für $q = 4$ und $m = 2$ Ameisen eine stärkere Krümmung auf dem Weg zur optimalen Lösung als die Kurve für $m = 3$ Ameisen. Je größer die Anzahl q elementarer Teilprobleme wird, desto größer wird auch die Abweichung des Pfades vom direkten Weg zur optimalen Lösung. In Abbildung 7.7 ist die Kurve für $q = 2$ nur leicht gekrümmt, die Kurve für $q = 128$ hat jedoch eine starke Krümmung und folgt im Mittelteil sehr deutlich dem entsprechenden Fixpunktvektor der stabilen Fixpunktmatrix. Noch deutlicher ist das in Abbildung 7.8 zu sehen, bei der für $q = 2$ keiner der Pheromonvektoren auf dem Weg zur Optimallösung eine Schleife hat, während für $q = 128$ durch den Einfluss des Fixpunktvektors eine große Schleife auf dem Weg zur optimalen Ecke gemacht wird. Der Grund hierfür ist, dass die größere Anzahl elementarer Teilprobleme zu einem kleineren Einfluss der Konkurrenz führt (vgl. Abschnitt 7.3), da mit hoher Wahrscheinlichkeit eine Malussituation mit großem Malus angenommen wird. Bei einer Situation ohne Konkurrenz (vgl. ACO-Modell mit $m = 1$ Ameise) basieren die Änderungen der Pheromonvektoren auf dem entsprechenden als Attraktor wirkenden Fixpunktvektor.

Abbildung 7.9 zeigt, im Vergleich zum linken Teil der Abbildungen 7.7 und 7.8, dass sich die Auswahlwahrscheinlichkeiten σ_{i1} , σ_{i2} und σ_{i3} , $i \in \{2, 3\}$ ähnlich wie die Pheromonwerte der zweiten und dritten Zeile verhalten. Die Tatsache, dass die Kurven der Pheromonvektoren und der Vektoren für die Auswahlwahrscheinlichkeiten nicht identisch sind, bedeutet, dass wir eine Verzerrung der Auswahlwahrscheinlichkeiten beobachten. Die ist insbesondere an den unterschiedlichen Startpunkten der Kurve zu erkennen (zur Berechnung der Startpunkt vgl. Gleichung (7.5) auf Seite 134).

Das zeitliche Zusammenspiel und die exakte Position der Pheromonvektoren des ACO-Modells und des zugehörigen Fixpunktvektors der stabilen Fixpunktmatrix sind in Abbildungen 7.10 und 7.11 für P_1^q mit $q = 64$ ele-

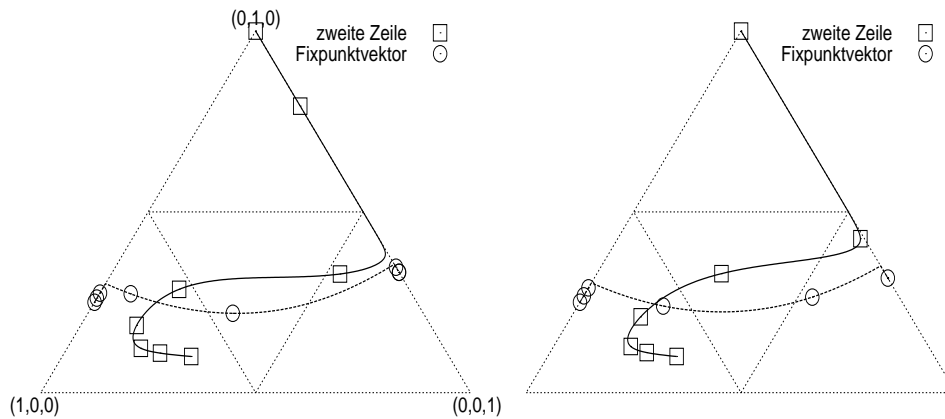


Abbildung 7.10: ACO-Modell für P_1^q ; $q = 64$; links 2 Ameisen, rechts 3 Ameisen; Kreise und Vierecke stellen die Pheromonvektoren $(\tau_{21}, \tau_{22}, \tau_{23})$ und die Fixpunktvektoren der zweiten Zeile zu den Zeitpunkten 0, 10, 20, 50, 100, 200, 500 und 1000 dar; der Startpunkt der zweiten Zeile liegt bei $(\tau_{21}, \tau_{22}, \tau_{23}) = (0.6, 0.1, 0.3)$; wenn nicht alle der 8 Werte dargestellt sind, sind die jeweils letzten identisch

mentaren Teilproblemen gezeigt. Auch hier kann man sehen, dass sich die Pheromonvektoren etwa 200 Iterationen lang (etwas mehr für $m = 2$ Ameisen und etwas weniger für $m = 3$ Ameisen) stark in Richtung des Fixpunktvektors der stabilen Fixpunktmatrix bewegen, bis der Einfluss der Konkurrenz wesentlich stärker wird.

Abbildung 7.12 zeigt die Veränderung der Pheromonvektoren in der zweiten und dritten Zeile der Pheromonmatrix, wenn die Summen-Pheromonauswertung und $m = 2$ Ameisen verwendet werden. Die Pfade der Pheromonvektoren der ersten Zeile sind denen, die sich bei Verwendung der lokalen Pheromonauswertung ergeben, so ähnlich, dass auf deren Darstellung verzichtet wurde (vgl. Abbildung 7.6). Der Vergleich mit den Abbildungen 7.7 und 7.8 für $m = 2$ Ameisen zeigt, dass der Pheromonpfad dem direkten Weg zur optimalen Lösung stärker folgt als bei der Verwendung der lokalen Pheromonauswertung. Dieses Verhalten konnte erwartet werden, da die kleine Probleminstanz P_1 einem Ein-Maschinen-Planungsproblem entspricht, bei die Summe der Abweichungen minimiert werden soll. Auf solchen Problemen hat sich die Summen-Pheromonauswertung als sehr gut erwiesen (vgl. Abschnitt 3.1).

Wir haben gesehen, dass sich das Verhalten des ACO-Modells auf einem beschränkten Permutationsproblem P^q aus einer gewichteten Summe von Aus-

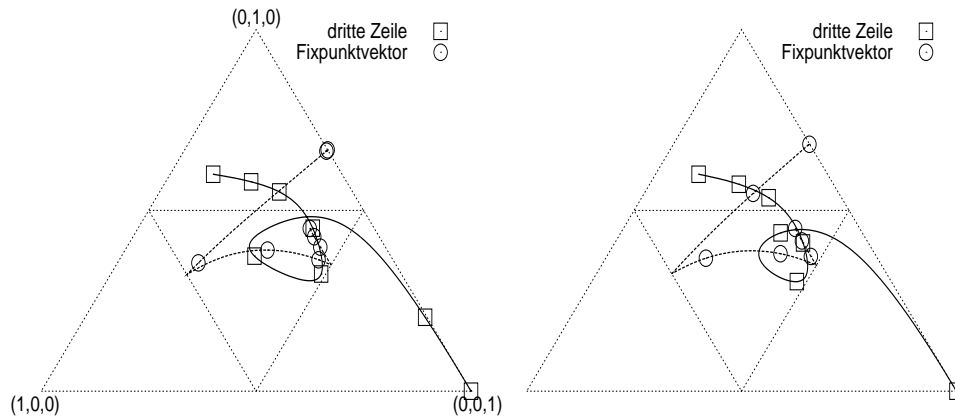


Abbildung 7.11: ACO-Modell für P_1^q ; $q = 64$; links 2 Ameisen, rechts 3 Ameisen; Kreise und Vierecke stellen die Pheromonvektoren $(\tau_{31}, \tau_{32}, \tau_{33})$ und die Fixpunktvektoren der dritten Zeile zu den Zeitpunkten 0, 10, 20, 50, 100, 200, 500 und 1000 dar; der Startpunkt der dritten Zeile liegt bei $(\tau_{31}, \tau_{32}, \tau_{33}) = (0.3, 0.6, 0.1)$; wenn nicht alle der 8 Werte dargestellt sind, sind die jeweils letzten identisch

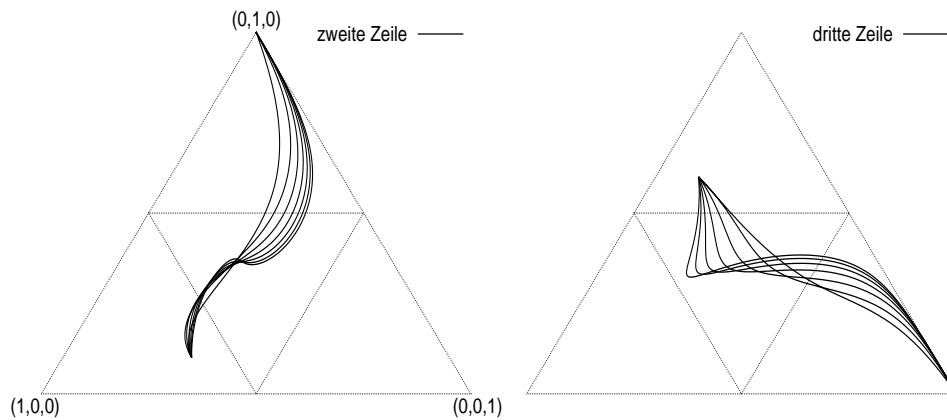


Abbildung 7.12: ACO-Modell für P_1^q und Summen-Pheromonauswertung für $q = 2, 4, 8, 16, 32, 64, 128$; links ist die Veränderung der Pheromonvektoren $(\tau_{21}, \tau_{22}, \tau_{23})$ bei einem Startpunkt $(\tau_{21}, \tau_{22}, \tau_{23}) = (0.6, 0.1, 0.3)$ dargestellt; rechts ist die Veränderung der Pheromonvektoren $(\tau_{21}, \tau_{22}, \tau_{23})$ bei einem Startpunkt $(\tau_{21}, \tau_{22}, \tau_{23}) = (0.3, 0.6, 0.1)$ dargestellt

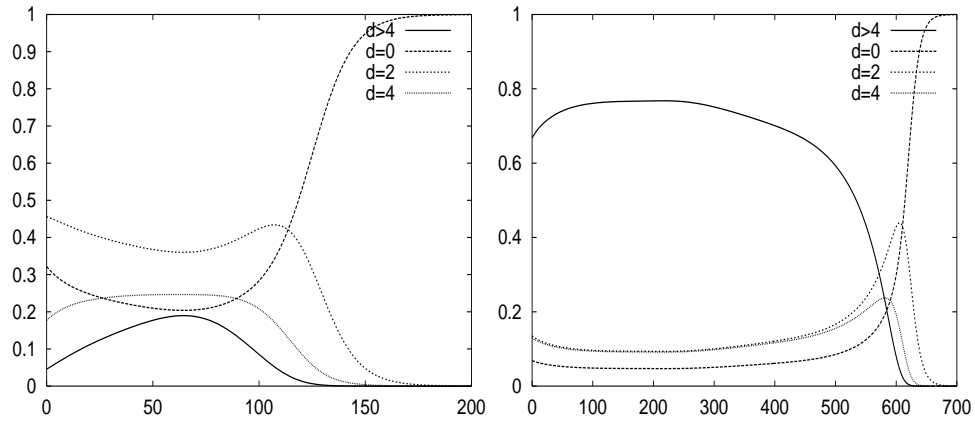


Abbildung 7.13: ACO-Modell für P_1^q mit einem elementaren Teilproblem P_1 der Größe $n_0 = 3$ und $m = 2$ Ameisen; links: $q = 4$, rechts: $q = 64$; Veränderung der Wahrscheinlichkeiten $\phi_d^{(2)}$ auf dem Teilproblem P_1^{q-1} ; dargestellt sind $\phi_0^{(2)}$, $\phi_2^{(2)}$, $\phi_{d=4}^{(2)}$ und $\phi_{d>4}^{(2)}$

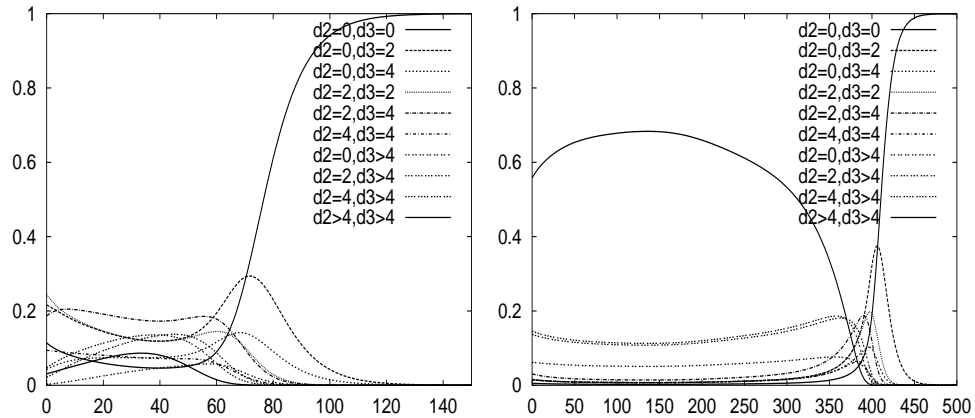


Abbildung 7.14: ACO-Modell für P_1^q mit einem elementaren Teilproblem P_1 der Größe $n_0 = 3$ und $m = 3$ Ameisen; links: $q = 4$, rechts: $q = 64$; Veränderung der Wahrscheinlichkeiten $\phi_{d_2, d_3}^{(3)}$ auf dem Teilproblem P_1^{q-1} ; dargestellt sind $\phi_{0,0}^{(3)}$, $\phi_{0,2}^{(3)}$, $\phi_{0,4}^{(3)}$, $\phi_{0,>4}^{(3)}$, $\phi_{2,2}^{(3)}$, $\phi_{2,4}^{(3)}$, $\phi_{2,>4}^{(3)}$, $\phi_{4,4}^{(3)}$, $\phi_{4,>4}^{(3)}$ und $\phi_{>4,>4}^{(3)}$

wahlwahrscheinlichkeiten auf P ergibt, die sich ausgehend von verschiedenen Malussituationen (bzw. unterschiedlichen Stufen der Konkurrenz) ergeben. Um den relativen Einfluss der Konkurrenzstufen (keine Konkurrenz, reine Konkurrenz und schwache Konkurrenz) zu untersuchen, wurden für alle Lösungsqualitätsunterschiede die Wahrscheinlichkeiten berechnet, dass sich die zwei Ameisen in einer Iteration auf dem Problem P_1^{q-1} um genau diesen Wert unterscheiden. Die möglichen Lösungsqualitätsunterschiede auf einem elementaren Teilproblem sind 0, 2 oder 4. Abbildung 7.13 zeigt die zeitliche Veränderung der Wahrscheinlichkeiten $\phi_0^{(2)}$, $\phi_2^{(2)}$, $\phi_{d=4}^{(2)}$ und $\phi_{d>4}^{(2)}$ für eine kleine Anzahl von elementaren Teilproblemen ($q = 4$) und eine große Anzahl von Teilproblemen ($q = 64$). Für drei Ameisen pro Iteration sind die entsprechenden zehn Wahrscheinlichkeiten in Abbildung 7.14 dargestellt (vgl. auch Abbildung 7.2 auf Seite 145, welche die Berechnung der zehn Wahrscheinlichkeiten erläutert). Beide Abbildungen zeigen, dass für eine große Anzahl elementarer Teilprobleme die Wahrscheinlichkeit $\phi_{d>4}^{(2)}$ (und entsprechend $\phi_{>4,>4}^{(3)}$) über einen Großteil des gesamten Laufs hinweg mehr als 50% beträgt. Dies entspricht der Wahrscheinlichkeit für eine konkurrenzlose Situation, bei der sich die Lösungsqualität der besten Ameise um mehr als 4 von der Lösungsqualität aller anderen Ameisen unterscheidet. Somit hat sowohl die reine als auch die schwache Konkurrenz nur einen relativ geringen Einfluss auf die Auswahlwahrscheinlichkeiten des Problems P^q . Erst wenn das ACO-Modell beginnt zu konvergieren, werden die Ameisen stärker in eine Konkurrenzsituation gelenkt. Dies war auch schon bei der Analyse der Dynamik der Pheromonvektoren zu sehen. Der stärkere Einfluss der Konkurrenz tritt bei $q = 64$ elementaren Teilproblemen und $m = 2$ Ameisen nach etwa 550 Iterationen ein. Bei $m = 3$ Ameisen tritt aufgrund der stärkeren Konkurrenzsituation der Effekt schon nach ca. 300 Iterationen ein.

Im Folgenden werden wir zeigen, dass der Einfluss der Verzerrung der Auswahlwahrscheinlichkeiten so groß werden kann, dass im ACO-Modell die optimale Lösung eines Problems nicht gefunden wird. Dies wird sogar für relativ kleine Permutationsprobleme der Fall sein. Hierfür betrachten wir ein Problem P_2 mit der folgenden Kostenmatrix.

$$C_2 = \begin{pmatrix} 0 & 1 & 100 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (7.29)$$

Abbildung 7.15 zeigt das Verhalten des ACO-Modells für das Problem P_2^q bei einer Initialisierung der Pheromonmatrix mit $\tau_{11} = 0.1$, $\tau_{12} = 0.2$, $\tau_{21} = 0.1$

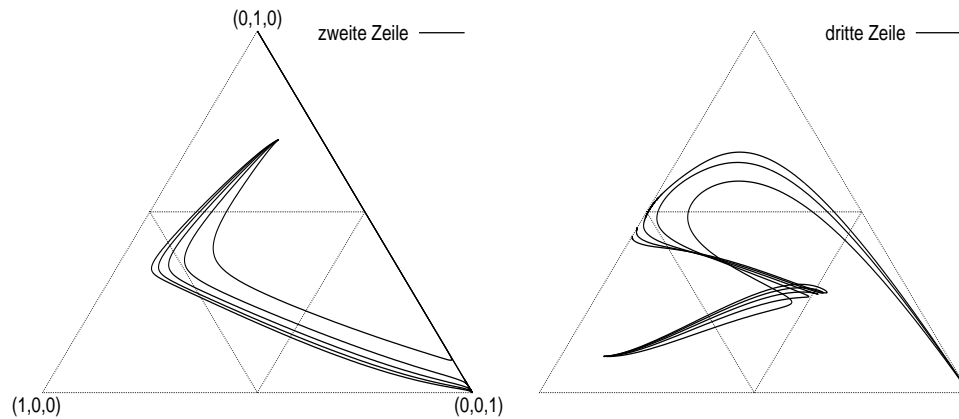


Abbildung 7.15: ACO-Modell für das Problem P_2^q mit $m = 2$ Ameisen und $q = 6, 10, 14, 18, 22$; die Initialisierung der Pheromonmatrix ist durch $\tau_{11} = 0.1$, $\tau_{12} = 0.2$, $\tau_{21} = 0.1$ und $\tau_{22} = 0.7$ festgelegt; links: die Veränderung der Pheromonwerte $\tau_{21}, \tau_{22}, \tau_{23}$; rechts: die Veränderung der Pheromonwerte $\tau_{31}, \tau_{32}, \tau_{33}$; für 6, 10 und 14 elementare Teilprobleme konvergiert das System zur Optimallösung mit $(\tau_{21}, \tau_{22}, \tau_{23}) = (0, 1, 0)$ (dies ist in der Abbildung schwer ersichtlich) und $(\tau_{31}, \tau_{32}, \tau_{33}) = (0, 0, 1)$; für 18 und 22 elementare Teilprobleme konvergiert die Pheromonmatrix gegen eine suboptimale Lösung mit $(\tau_{21}, \tau_{22}, \tau_{23}) = (0, 0, 1)$ und $(\tau_{31}, \tau_{32}, \tau_{33}) = (1 - x, x, 0)$, $x \in (0, 1)$

und $\tau_{22} = 0.7$ für unterschiedliche viele Teilprobleme q . Für $q \in \{6, 10, 14\}$ elementare Teilprobleme konvergiert das System zur optimalen Lösung. Für eine größere Anzahl elementarer Teilprobleme (in Abbildung 7.15 für $q \in \{18, 22\}$) ist der Einfluss der Verzerrung der Auswahlwahrscheinlichkeiten so groß, dass das System zu einer suboptimalen Lösung konvergiert. Die Abbildung zeigt auch, dass schon für eine kleine Anzahl elementarer Teilprobleme das Verhalten des ACO-Modells sehr stark durch die Verzerrung der Auswahlwahrscheinlichkeiten beeinflusst wird. Die Konkurrenz wird jedoch früh genug größer, sodass das ACO-Modell noch zur optimalen Lösung konvergiert. Falls das ACO-Modell nicht zur optimalen Lösung konvergiert, strebt das System zu einer Pheromonmatrix mit folgenden Werten. Eine Lösungspermutation, die basierend auf dieser Matrix erzeugt wird, verursacht für jedes $x \in (0, 1)$ die

Kosten 2.

$$\begin{pmatrix} x & 1-x & 0 \\ 0 & 0 & 1 \\ 1-x & x & 0 \end{pmatrix}, x \in (0, 1)$$

Diese Untersuchungen bezogen sich auf eine speziell gewählte initiale Pheromonmatrix. Wir haben das Verhalten mit allen 666 möglichen und zulässigen initialen Pheromonmatrizen getestet, die aus Kombinationen der Pheromonwerte $\tau_{ij} \in \{0.1, 0.2, \dots, 0.8\}$ erzeugt wurden (ein Pheromonwert 0.9 kann so bei der initialen Pheromonmatrix nicht vorkommen, da die zwei weiteren Werte der Zeile mindestens 0.1 sind, und somit die Zeilensumme ungleich 1.0 wäre). Abbildung 7.16 zeigt für $q \in \{1, 2, \dots, 20, 25, \dots, 60\}$ elementare Teilprobleme die Häufigkeit der Fälle, für die das System nicht in der Lage war, die optimale Lösung zu finden. Selbst für das Problem P_2^2 mit nur 2 elementaren Teilproblemen wurde in 83 der 666 Fälle die optimale Lösung nicht gefunden. Der Anteil der nicht optimal gelösten Probleme wird mit wachsender Anzahl elementarer Teilprobleme deutlich größer. Für P_2^3 konnten 101 Probleme nicht optimal gelöst werden, und für P_2^{60} konnten 296 der 666 Probleme nicht optimal gelöst werden. Dies zeigt deutlich, wie durch die Vergrößerung des Einflusses der Verzerrung der Auswahlwahrscheinlichkeiten bedingt durch eine größere Anzahl elementarer Teilprobleme q das Optimierungsverhalten schlechter wird.

Die Untersuchungen dieses Abschnitts haben gezeigt, dass das dynamische Verhalten des ACO-Modells stark von der Verzerrung der Auswahlwahrscheinlichkeiten abhängt. Der Einfluss kann so groß werden, dass selbst für kleine Probleme, bei denen eine Konkurrenz zwischen mehreren Ameisen vorhanden ist, evtl. nur eine suboptimale Lösung gefunden werden kann. Im Allgemeinen gilt, je kleiner die Anzahl der Ameisen und je größer eine Probleminstanz ist, desto stärker wird der Einfluss des stabilen Fixpunktvektors und somit der Verzerrung der Auswahlwahrscheinlichkeiten.

7.7 Simulationen

In diesem Abschnitt präsentieren wir Ergebnisse unterschiedlicher Läufe des ACO-Algorithmus auf eingeschränkten Permutationsproblemen und vergleichen sie mit den Resultaten des ACO-Modells. Es ist offensichtlich, dass das Modell das Verhalten des ACO-Algorithmus nicht sinnvoll widerspiegelt, falls das Optimierungsproblem klein und die Verdunstungsrate ρ groß ist. In diesem Fall haben einzelne Entscheidungen der Ameisen einen starken Einfluss auf das

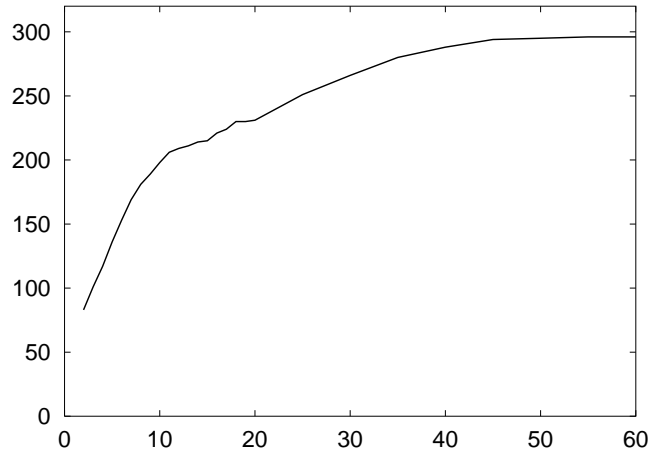


Abbildung 7.16: ACO-Modell für P_2^q ; $q = 1, 2, \dots, 20, 25, \dots, 60$ und 2 Ameisen; das elementare Teilproblem P_2 hat eine Größe von $n_0 = 3$; dargestellt ist die Anzahl der initialen Pheromonmatrizen, für die das Modell nicht zur optimalen Lösung konvergiert; hierfür wurden alle 666 Matrizen, für die $\tau_{ij} \in \{0.1, 0.2, \dots, 0.8\}$, $i, j \in [1, 3]$ eine zulässige Pheromonmatrix für das ACO-Modell ergibt, als Initialisierungsmatrix gewählt

Optimierungsverhalten, und unterschiedliche Läufe können sehr unterschiedliche Ergebnisse z.B. bezüglich der Lösungsqualität liefern. Im Allgemeinen ist es schwierig, auf sinnvolle Art und Weise ein Durchschnittsverhalten eines ACO-Algorithmus zu definieren. Hinsichtlich der Lösungsqualität scheint dies noch möglich — doch der Durchschnitt von Pfaden der Pheromonvektoren auf dem Weg zu (möglicherweise unterschiedlichen) Konvergenzpunkten bei unterschiedlichen Läufen, ist wesentlich komplizierter zu definieren. Um einen Vergleich aussagekräftig machen zu können, müssen wir die Varianz des Optimierungsverhaltens unterschiedlicher Läufe einschränken. Daher wählen wir eine sehr kleine Verdunstungsrate $\rho = 0.0001$. Für das ACO-Modell wurde ein Wert von $\rho = 0.1$ gewählt, da ansonsten die Laufzeit für das ACO-Modell auf großen Problemen mit 3 Ameisen sehr groß wird. Wenn wir Daten des ACO-Modells mit Daten des ACO-Algorithmus vergleichen, dann vergleichen wir die t -te Iteration des ACO-Modells mit der $1000 \cdot t$ -ten Iteration des ACO-Algorithmus. Die Menge des Pheromons, die in einer Iteration beim ACO-Modell auf die Pheromonmatrix addiert wird, entspricht bei den gewählten Verdunstungsraten der Menge des Pheromons, die beim ACO-Algorithmus in 1000 Iterationen aufaddiert wird. Man beachte, dass die modellierte Ak-

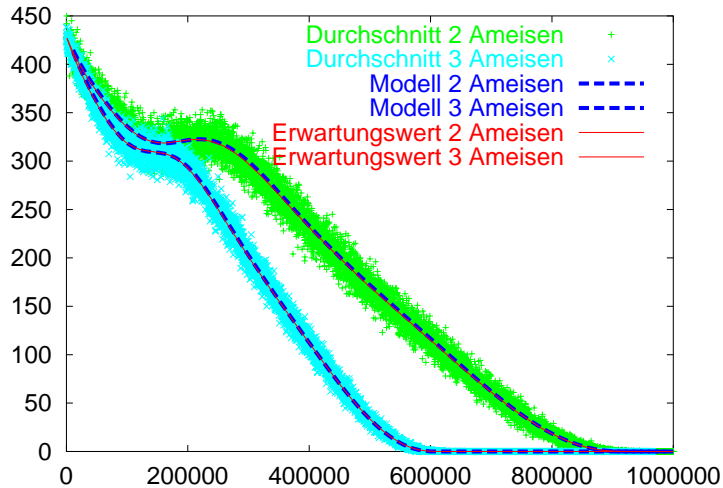


Abbildung 7.17: Lösungsqualität des ACO-Modells und des ACO-Algorithmus auf dem Problem P_1^{128} mit $m = 2$ und $m = 3$ Ameisen über 1000000 (und entsprechend 1000) Iterationen hinweg; Durchschnitt m Ameisen: durchschnittliche Lösungsqualität, die von m Ameisen in der entsprechenden Iteration vom ACO-Algorithmus gefunden wurde; Erwartungswert m Ameisen: die erwartete Qualität einer Ameise, welche die Pheromonmatrix des ACO-Algorithmus in der entsprechenden Iteration als Berechnungsgrundlage hat; Modell m Ameisen: Lösungsqualität, wie sie durch das ACO-Modell vorhergesagt wird; die Kurven für den Erwartungswert der Qualität von m Ameisen beim ACO-Algorithmus und der Lösungsqualität des ACO-Modells sind nahezu identisch

tualisierungsmethode nicht völlig der des ACO-Algorithmus entspricht. Beim ACO-Algorithmus ist das Pheromon, welches innerhalb von 1000 Iterationen verdunstet wird, zum Teil auch Pheromon, welches erst in den letzten 1000 Iterationen addiert wurde. Bei einem 1000 Generationen entsprechenden Schritt im ACO-Modell wird nur „altes“ Pheromon verdunstet. Zusätzlich hat beim ACO-Algorithmus das Pheromon, welches in einer Iteration auf die Pheromonmatrix addiert wird, einen unmittelbaren Einfluss auf die Entscheidungen in der nächsten Iteration. Das ACO-Modell aktualisiert die Pheromonwerte nur alle 1000 modellierten Iterationen (also in jeder Iteration des ACO-Modells).

Abbildung 7.17 zeigt das Optimierungsverhalten bezüglich der Lösungsqualität des ACO-Algorithmus auf dem Beispielproblem P_1^q , welches in Abschnitt 7.3 eingeführt wurde. Für die Untersuchungen wurde ein Problem mit $q = 128$ elementaren Teilproblemen benutzt (für die Kostenmatrix siehe (7.11)

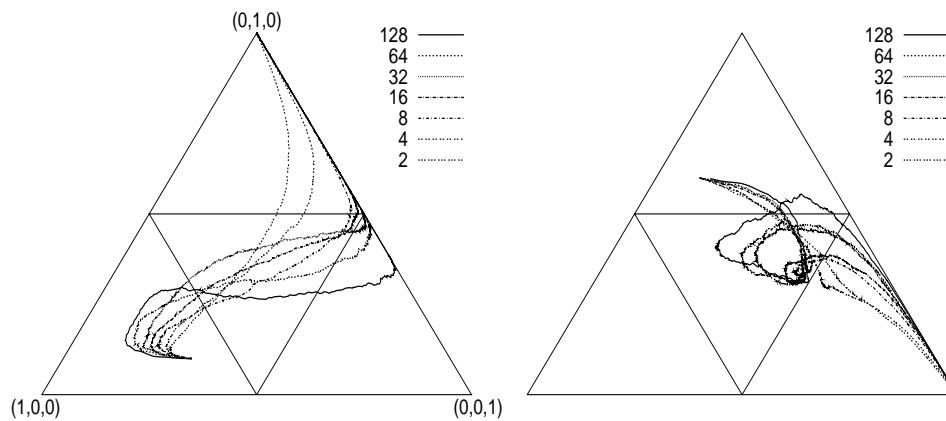


Abbildung 7.18: ACO-Algorithmus für P_1^q mit einem elementaren Teilproblem P_1 der Größe $n_0 = 3$, $q = 2, 4, 8, 16, 32, 64, 128$ und $m = 2$ Ameisen; dargestellt sind die Pheromonvektoren eines jeweils zufällig gewählten Teilproblems; links dargestellt ist die Veränderung der Pheromonwerte $\tau_{21}, \tau_{22}, \tau_{23}$ ausgehend vom Punkt $(\tau_{21}, \tau_{22}, \tau_{23}) = (0.6, 0.1, 0.3)$; rechts dargestellt ist die Veränderung der Pheromonwerte $\tau_{31}, \tau_{32}, \tau_{33}$ ausgehend vom Punkt $(\tau_{31}, \tau_{32}, \tau_{33}) = (0.3, 0.6, 0.1)$

auf Seite 137). Man beachte, dass bei diesem eingeschränkten Permutationsproblem aufgrund der Pheromonwerte des ACO-Algorithmus der Erwartungswert für die Lösungsqualität eines Teilproblems, und somit auch die erwartete Lösungsqualität der Gesamtlösung, berechnet werden kann. Die Abbildung zeigt, dass die Lösungsqualität des ACO-Modells nahezu der erwarteten Lösungsqualität des ACO-Algorithmus entspricht (die entsprechenden Kurven in Abbildung 7.17 sind beinahe identisch).

Die beobachtete durchschnittliche Lösungsqualität für $m = 2$ und $m = 3$ Ameisen (also der Durchschnitt der Lösungsqualitäten der Lösungen, die in einer Iteration gefunden wurden) schwankt um die erwartete Lösungsqualität. Eine sehr interessante Beobachtung ist, dass sich die Lösungsqualität sowohl beim ACO-Modell als auch beim ACO-Algorithmus in einer bestimmten Phase des Laufs sogar verschlechtert. Dies liegt nicht an zufällig schlechten Entscheidungen, sondern wird auch vom deterministischen Modell so vorhergesagt. Bei einem stärkeren Einfluss der Konkurrenz ($m = 3$) tritt dieser Effekt nicht auf, die vom Modell vorhergesagte Lösungsqualität verbessert sich von Iteration zu Iteration. Dieses Verhalten lässt sich wie beim Modell auch beim ACO-Algorithmus beobachten.

Abbildung 7.18 zeigt Ergebnisse des ACO-Algorithmus auf Problemen P_1^q mit unterschiedlich vielen elementaren Teilproblemen für $m = 2$ Ameisen. Jede Kurve der Abbildung bezieht sich auf ein zufällig gewähltes elementares Teilproblem eines Laufs, da die Berechnung eines sinnvollen durchschnittlichen Verhaltens der Pheromonvektoren der einzelnen Teilprobleme von P_1^q schwierig ist. Man beachte hierbei, dass unter Umständen beim ACO-Algorithmus manche elementaren Teilprobleme optimal und andere nicht optimal gelöst werden. Die dargestellten Ergebnisse basieren auf Pheromonwerten einzelner elementarer Teilprobleme einzelner Läufe. Daher sind die Kurven nicht sehr glatt und unterliegen zufälligen Schwankungen. Die Abbildung zeigt nichtsdestotrotz beim Vergleich mit den Abbildungen 7.7 und 7.8, dass das ACO-Modell das Verhalten der Pheromonvektoren des ACO-Algorithmus sehr gut vorhersagt.

Abbildung 7.19 zeigt die Pheromonwerte des ACO-Modells für P_1^q mit $q = 64$ und $m = 2$ Ameisen. Zusätzlich sind die Pheromonwerte aller 64 elementaren Teilprobleme des ACO-Algorithmus zu unterschiedlichen Zeitpunkten eingezeichnet. Die vom Modell vorhergesagten Pheromonwerte zu den entsprechenden Zeitpunkten sind durch Quadrate dargestellt. Die Pheromonwerte des ACO-Algorithmus unterliegen, bedingt durch die Zufallsentscheidungen des Algorithmus, einer zeitlich abhängigen Varianz. Die Pheromonwerte des Algorithmus entsprechen mehr oder weniger den durch das Modell vorhergesagten Pheromonwerten. Bei Iterationen 10000 und 20000 des ACO-Algorithmus ist zu erkennen, dass die Pheromonwerte des Algorithmus leicht weiter fortgeschritten sind, als die entsprechenden vom ACO-Modell vorhergesagten Pheromonwerte zu den Zeitpunkten 10 und 20 des ACO-Modells.

Beim ACO-Modell können natürlich nicht alle Aspekte des Verhaltens von ACO-Algorithmen beobachtet werden. Für jedes eingeschränkte homogene Permutationsproblem P^q ist das vom ACO-Modell vorhergesagte Verhalten auf jedem elementaren Teilproblem inhärent identisch. Im Gegensatz hierzu kann sich der ACO-Algorithmus zufallsbedingt auf jedem elementaren Teilproblem unterschiedlich verhalten. Auf dem Problem P_2^q „löst“ das ACO-Modell mit $m = 2$ Ameisen entweder alle oder keines der elementaren Teilprobleme optimal, der ACO-Algorithmus verhält sich auf einzelnen elementaren Teilproblemen evtl. unterschiedlich. Tabelle 7.1 zeigt die Anzahl elementarer Teilprobleme, welche optimal gelöst wurden, d.h. bei denen die Werte der Pheromonmatrix für das entsprechende Teilproblem zur optimalen Lösung konvergieren. Für die initiale Pheromonmatrix wurden für jedes elementare Teilproblem die Werte $\tau_{11} = \tau_{12} = \tau_{21} = \tau_{22} = 1/3$ gewählt, was einer üblichen Initialisierung

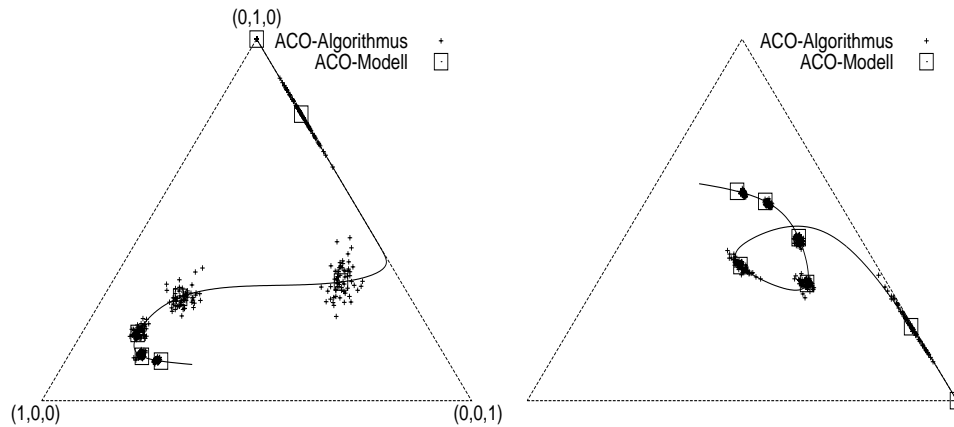


Abbildung 7.19: ACO-Algorithmus und ACO-Modell für P_1^q mit einem elementaren Teilproblem P_1 der Größe $n_0 = 3$, $q = 64$ und $m = 2$ Ameisen zu den Zeitpunkten 10000, 20000, 50000, 100000, 200000, 500000 und 1000000 des ACO-Algorithmus (und entsprechend zu den Zeitpunkten 10, 20, 50, 100, 200, 500 und 1000 des ACO-Modells) — die entsprechende Kurve des ACO-Modells (vgl. linke Seite der Abbildungen 7.7 und 7.8) sind auch abgebildet; links dargestellt ist der Pheromonvektor $(\tau_{21}, \tau_{22}, \tau_{23})$ ausgehend vom Punkt $(\tau_{21}, \tau_{22}, \tau_{23}) = (0.6, 0.1, 0.3)$; rechts dargestellt ist der Pheromonvektor $(\tau_{31}, \tau_{32}, \tau_{33})$ ausgehend vom Punkt $(\tau_{31}, \tau_{32}, \tau_{33}) = (0.3, 0.6, 0.1)$

für Pheromonmatrizen bei ACO-Algorithmen entspricht. Die Tabelle zeigt, dass für $q = 5$ alle elementaren Teilprobleme optimal gelöst wurden. Je größer die Anzahl elementarer Teilprobleme wird, desto geringer wird der Anteil der vom ACO-Algorithmus optimal gelösten elementaren Teilprobleme. Für $q = 100$ wurden im Mittel nur noch 14.7% der Teilprobleme optimal gelöst. Das ACO-Modell löst für die gewählte Initialisierungsmatrix und $q = 100$ alle Teilprobleme optimal. Man beachte allerdings, dass auch für das ACO-Modell gilt, dass für größere Werte q das Problem P^q immer schwieriger zu lösen ist. Denn für größere q wird der Anteil der möglichen initialen Pheromonmatrizen, die nicht zu einer optimalen Lösung führen, größer (vgl. Abbildung 7.16).

Tabelle 7.1: ACO-Algorithmus für das Problem P_2^q und $m = 2$ Ameisen: die Werte entsprechen der durchschnittlichen Anzahl elementarer Teilprobleme, die optimal gelöst wurden, d.h. bei denen die Werte der Pheromonmatrix für das entsprechende Teilproblem zur optimalen Lösung konvergiert sind; die Werte sind Durchschnittswerte über jeweils 10 Läufe

q	optimal	%	Varianz
5	5.0	100.0	0.0
10	8.7	87.0	0.6
15	10.1	67.3	0.9
20	10.8	54.0	2.4
25	12.0	48.0	2.4
30	12.1	40.3	1.7
40	12.5	31.3	2.7
50	14.1	28.2	3.7
75	14.6	19.5	2.4
100	14.7	14.7	2.2

7.8 Heterogene Permutationsprobleme

7.8.1 ACO-Modell

In diesem Abschnitt wenden wir die in den vorigen Abschnitten vorgestellten Methoden auf heterogene Permutationsprobleme an. Wir betrachten den Fall für $m = 2$ Ameisen. Es seien P_1 und P_2 zwei elementare Teilprobleme mit Kostenmatrizen C_1 und C_2 . Dann bezeichnen wir mit $P_1^q P_2^r$ das heterogene Permutationsproblem, welches aus q elementaren Teilproblemen P_1 und aus r elementaren Teilproblemen P_2 besteht. Die Kostenmatrix mit der Bezeichnung $C_1^q C_2^r$ wird wie die homogene Kostenmatrix in Gleichung (7.12) auf Seite 138 definiert, nur dass wir nun nicht notwendigerweise identische elementare Kostenmatrizen haben. In diesem Abschnitt werden wir heterogene Permutationsprobleme betrachten, bei denen eine Kostenmatrix das Vielfache der anderen Kostenmatrix ist, d.h. jedes Element der Matrix C_1 wird mit einer ganzzahligen Konstanten $x \geq 1$ multipliziert. Wir bezeichnen diese Probleme mit $P_1^q P_x^r$. Für $x = 1$ erhalten wir ein homogenes Permutationsproblem. Da die Struktur der Probleme P_1 und P_x gleich ist, werden wir auf den Problemen P_1^q und P_x^q jeweils für sich alleine betrachtet das identische Optimierungsver-

halten beobachten, wobei die Lösungsqualität des Problems P_x^q immer genau das x -fache des Problems P_1^q ist. Da sich die Teilprobleme des Permutationsproblems $P_1^q P_x^r$ nur um diesen Skalierungsfaktor unterscheiden, scheint es interessant, wie das Optimierungsverhalten durch diese Skalierung beeinflusst wird.

Die Betrachtung solcher heterogener Optimierungsprobleme, bei der sich die unabhängigen Teilprobleme bis auf die Skalierung nicht voneinander unterscheiden, spiegelt charakteristische Eigenschaften von Optimierungsproblemen der realen Welt wieder. Betrachten wir beispielsweise ein „Vehicle-Routing“ Problem, bei dem ein Fahrzeug ausgehend von einem Depot mehrere Kunden in unterschiedlichen Städten auf einer möglichst schnellen Tour besuchen muss. Nehmen wir an, dass sich die Standorte der Kunden im Allgemeinen in der Nähe eines Häufungspunktes befinden, da jeweils mehrere Kunden in unterschiedlichen Städten besucht werden müssen. Die Entfernungen der Städte untereinander seien deutlich größer als die Entfernungen der Kunden innerhalb einer Stadt. Wenn die Anzahl der Städte nicht zu groß ist, dann wird es wahrscheinlich so sein, dass alle Kunden innerhalb einer Stadt auf einer einzelnen Subtour besucht werden. Die Hauptschwierigkeit besteht darin, jeweils eine möglichst gute Subtour innerhalb einer Stadt zu finden, die alle Kunden der Stadt abdeckt. Falls nun beispielsweise die Anzahl der Kunden pro Stadt in etwa gleich ist, sich die Städte aber z.B. in deren Größe oder wegen des unterschiedlichen Verkehrsaufkommens bezüglich der möglichen Durchschnittsgeschwindigkeit eines Fahrzeugs unterscheiden, dann besteht das Optimierungsproblem aus mehr oder weniger unabhängigen Teilproblemen, die sich im Wesentlichen nur um einen Skalierungsfaktor unterscheiden. Es ist offensichtlich, dass es einfacher ist, die unabhängigen Teilprobleme jeweils einzeln zu lösen. Hierfür muss man jedoch die einzelnen Teilprobleme identifizieren können, was im Allgemeinen sehr schwierig ist.

Um mit dem ACO-Modell ein heterogenes Permutationsproblem der Form $P_1^q P_x^r$ zu untersuchen, wird zwischen der Berechnung der Auswahlwahrscheinlichkeiten für (jedes beliebige der q elementaren Teilprobleme) P_1 und der Berechnung der Auswahlwahrscheinlichkeiten für (jedes beliebige der r elementaren Teilprobleme) P_x unterschieden. Im ersten Fall sprechen wir von der Berechnung der Auswahlwahrscheinlichkeiten für P_1^q , im zweiten Fall sprechen wir von der Berechnung der Auswahlwahrscheinlichkeiten für P_x^q . Es wird die Methode aus Abschnitt 7.3 angewendet. Wir betrachten alle Malussituationen, die durch die Auswertung aller bis auf ein Teilproblem entsteht, und berechnen die Auswahlwahrscheinlichkeiten auf dem letzten betrachteten Teil-

problem als gewichtete Summe von Auswahlwahrscheinlichkeiten, die auf dem Teilproblem P_1 oder P_x bei den entsprechenden Malussituationen entstehen. Die Gewichte sind jeweils genau die Wahrscheinlichkeiten, dass die entsprechende Malussituation eintritt. Da wir ein heterogenes Permutationsproblem, zusammengesetzt aus zwei unterschiedlichen Typen von elementaren Teilproblemen betrachten, werden wir die Wahrscheinlichkeiten für mögliche Malussituationen auf dem Problem $P_1^q P_x^{r-1}$ und auf dem Problem $P_1^{q-1} P_x^r$ berechnen.

Beispiel: Die Kostenmatrizen C_1 und C_2 der Probleme P_1 und P_2 sind durch

$$C_1 = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix} \quad C_2 = \begin{pmatrix} 0 & 2 & 4 \\ 2 & 0 & 2 \\ 4 & 2 & 0 \end{pmatrix}$$

definiert. Wir betrachten das elementare Teilproblem P_1 der eingeschränkten Permutationsprobleme P_1^2 und $P_1 P_2$. Die möglichen Lösungsqualitätsstufen des Problems P_1 (P_2) sind 0, 2 und 4 (0, 4 und 8). Die Pheromonmatrix sei für jedes der zwei elementaren Teilprobleme wieder wie in Gleichung (7.21) auf Seite 147 initialisiert.

Der Einfluss des jeweils anderen elementaren Teilproblems auf das elementare Teilproblem P_1 wird im Folgenden dargestellt. Es gilt (vgl. Seite 153) $\phi_0^{(2)} = \psi'_0 \approx 0.566$, $\phi_2^{(2)} = \psi'_2 + \psi'_{-2} = 0.398$ und $\phi_4^{(2)} = \psi'_4 + \psi'_{-4} = 0.036$. Mit den Auswahlwahrscheinlichkeiten der unterschiedlichen Malussituationen aus Abschnitt 7.4.2 berechnen wir die Auswahlwahrscheinlichkeiten des elementaren Teilproblems P_1 des eingeschränkten Permutationsproblems. Beispielsweise die Wahrscheinlichkeit, dass die beste von zwei Ameisen auf dem elementaren Teilproblem P_1 des eingeschränkten Permutationsproblems P_1^2 Element 1 auf Platz 1 wählt, ist (vgl. Gleichung (7.16) auf Seite 142)

$$\begin{aligned} \sigma_{11}^{(2)} &= \phi_0^{(2)} \cdot \sigma_{11}'^{(2)} + \phi_2^{(2)} \cdot \sigma_{11}'^{(2;2)} + \phi_4^{(2)} \cdot \sigma_{11}'^{(2;4)} \\ &\approx 0.566 \cdot 0.175 + 0.398 \cdot 0.146 + 0.036 \cdot 0.109 \approx 0.161 \end{aligned}$$

Die Wahrscheinlichkeit, dass die beste von zwei Ameisen auf dem elementaren Teilproblem P_1 des eingeschränkten heterogenen Permutationsproblems $P_1 P_2$ Element 1 auf Platz 1 wählt, ist

$$\begin{aligned} \sigma_{11}^{(2)} &= \phi_0^{(2)} \cdot \sigma_{11}'^{(2)} + \phi_4^{(2)} \cdot \sigma_{11}'^{(2;4)} + \phi_{>4}^{(2)} \cdot \sigma_{11}' \\ &\approx 0.566 \cdot 0.175 + 0.398 \cdot 0.109 + 0.036 \cdot 0.1 \approx 0.146 \end{aligned}$$

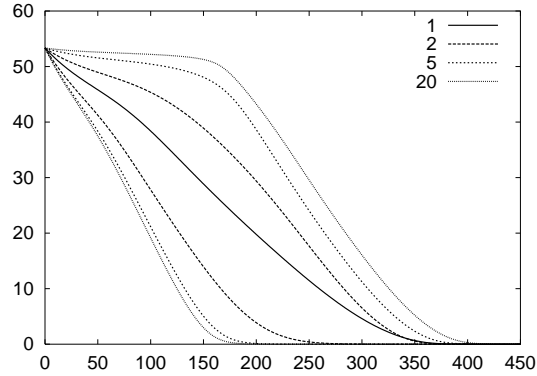


Abbildung 7.20: Durchschnittliche Lösungsqualität auf den elementaren Teilproblemen P_1^{20} und P_x^{20} des heterogenen Permutationsproblems $P_1^{20}P_x^{20}$ für $x \in \{1, 2, 5, 20\}$; die Lösungsqualität für P_x^{20} wurde durch x geteilt; die Initialisierung auf den elementaren Teilproblemen war jeweils $\tau_{ij} = 1/3$, $i, j \in [1, 3]$. Für $x \in \{2, 5, 20\}$ geben die jeweils linken Kurven das Verhalten auf P_x^{20} und die rechten Kurven das Verhalten auf P_1^{20} an; für $x = 1$ sind die Kurven identisch

7.8.2 Dynamisches Verhalten und Simulationen

Abbildung 7.20 zeigt die auf der Veränderung der Pheromonwerte basierende Entwicklung der erwarteten durchschnittlichen Lösungsqualität des ACO-Modells auf den zwei homogenen Teilproblemen P_1^{20} und P_x^{20} des Problems $P_1^{20}P_x^{20}$ für unterschiedliche Werte von x . Man beachte, dass das Optimierungsverhalten für jedes $x \geq 41$ hinsichtlich der dynamischen Veränderung der Pheromonvektoren identisch ist. Denn für jedes $x \geq 41$ ist der auf dem Problem P_1^{20} angenommene Lösungsqualitätsunterschied der beiden Ameisen entweder immer kleiner als $4 \cdot 20 = 80$, und der auf P_x^{20} angenommene Lösungsqualitätsunterschied entweder 0 oder mindestens $2 \cdot 41 = 82$. Da für alle $x \geq 41$ der kleinste Lösungsqualitätsunterschied auf P_x^{20} größer als der größte Lösungsqualitätsunterschied auf P_1^{20} ist, ergibt sich für alle $x \geq 41$ das gleiche dynamische Verhalten der Pheromonwerte.

Abbildung 7.20 zeigt, dass die durchschnittliche Lösungsqualität auf dem Teilproblem P_x^{20} des Problems $P_1^{20}P_x^{20}$ für ansteigende x deutlich schneller zur Optimallösung konvergiert als die durchschnittliche Lösungsqualität auf dem Problem P_1^{20} . Falls $x = 20$ gilt, bleibt die Lösungsqualität für P_1^{20} während der ersten 150 Generationen nahezu unverändert. Denn mit relativ hoher Wahrscheinlichkeit ist der Unterschied bezüglich der erreichten Lösungsqualität auf P_x^{20} so groß, dass sich die Pheromonwerte für P_1^{20} so verhalten, als hätten wir

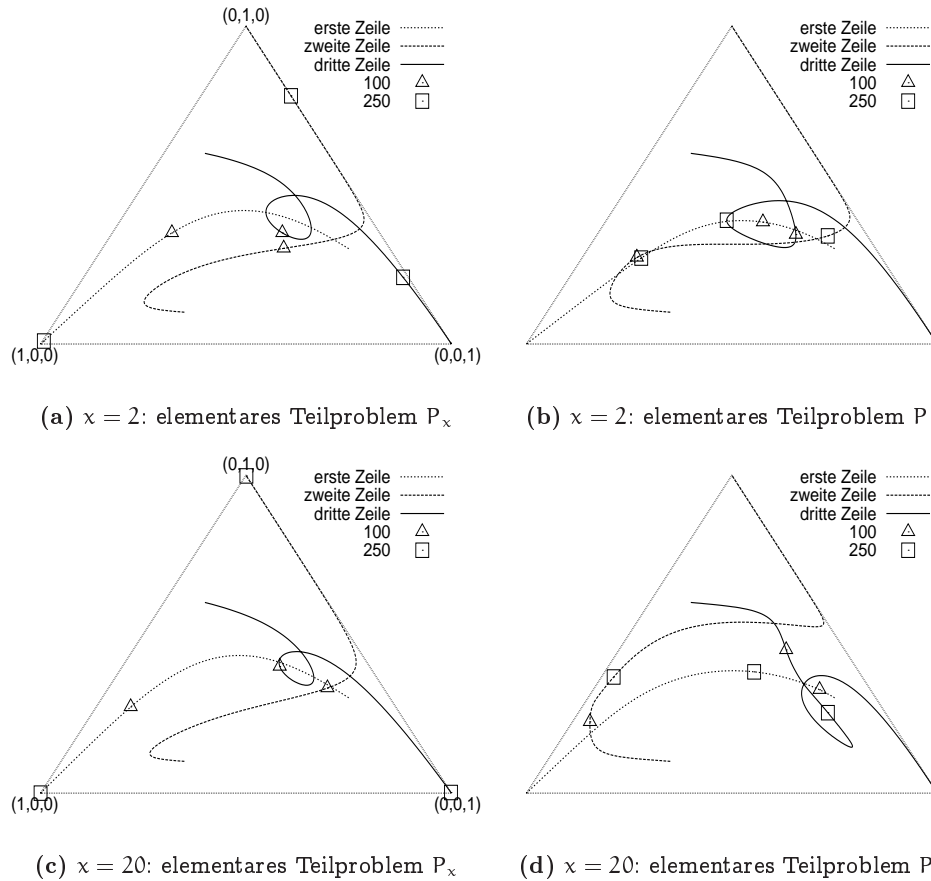


Abbildung 7.21: ACO-Modell für $P_1^{20} P_x^{20}$ mit elementaren Teilproblemen P_1 und P_x der Größe $n_0 = 3$; $x = 2$ (oben), $x = 20$ (unten); dargestellt ist die Dynamik der Pheromonvektoren $(\tau_{11}, \tau_{12}, \tau_{13})$, $(\tau_{21}, \tau_{22}, \tau_{23})$ und $(\tau_{31}, \tau_{32}, \tau_{33})$

eine Malussituation mit einem Malus > 4 . Wie wir bei den Untersuchungen der homogenen Permutationsprobleme gesehen haben, streben die Pheromonwerte dann zu den stabilen Fixpunktvektoren. Da die Initialisierungsmatrix eine entsprechende Fixpunktmatrix ist, bleiben die Pheromonvektoren auf P_1^{20} solange in einem nahezu stabilen Zustand, bis sich auf dem Problem P_1^{20} eine Konkurrenzsituation der Ameisen aufgrund der nahezu gleichen Lösungsqualität der beiden Ameisen auf dem Teilproblem P_x^{20} ergibt.

Mit größer werdendem x wird der Einfluss von P_1^{20} auf P_x^{20} immer geringer, und das Verhalten auf dem Teilproblem P_x^{20} von $P_1^{20} P_x^{20}$ wird dem Verhalten des ACO-Modells auf einem homogenen Permutationsproblem P_x^{20} immer ähnlicher.

Die Dynamik der Pheromonvektoren auf dem heterogenen Permutations-

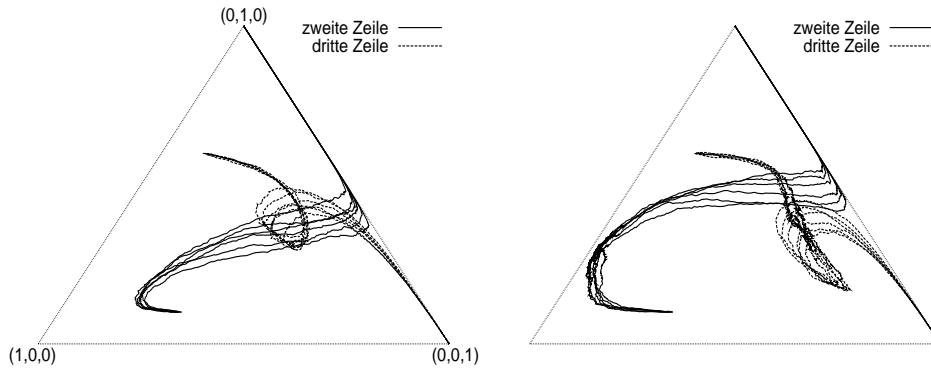


Abbildung 7.22: ACO-Algorithmus für $P_1^{20}P_x^{20}$ mit $\chi = 20$; links: Veränderung der Pheromonwerte für 6 elementare Teilprobleme P_x ; rechts: Veränderung der Pheromonwerte für 6 elementare Teilprobleme P_1

problem $P_1^{20}P_x^{20}$ ist sowohl für das ACO-Modell als auch für den ACO-Algorithmus in Abbildung 7.21 und 7.22 für Skalierungsfaktoren $\chi = 2$ und $\chi = 20$ dargestellt. Falls nicht explizit erwähnt wird, dass eine andere Initialisierung gewählt wurde, sind die Initialisierungsmatrizen der Teilprobleme jeweils wie in Gleichung (7.21) auf Seite 147 gewählt, d.h. die jeweils initial gewählte Pheromonmatrix wird durch die Werte $\tau_{11} = 0.1$, $\tau_{12} = 0.3$, $\tau_{21} = 0.6$ und $\tau_{22} = 0.1$ definiert. Beim Vergleich der Dynamik der Pheromonvektoren auf P_1 und P_x erkennt man, dass die Vektoren beim Problem P_1 die optimale Lösung viel schneller erreichen, als beim Problem P_x (siehe die entsprechenden Markierungen bei der Entwicklung der Pheromonvektoren nach 100 und 250 Generationen in Abbildung 7.21). Bei P_x sind die optimalen Eckpunkte nach ca. 250 Generationen schon nahezu erreicht. Die Kurven für P_x unterscheiden sich nicht sehr stark für $\chi = 2$ und $\chi = 20$. Im Gegensatz dazu unterscheidet sich die Dynamik der Pheromonvektoren auf dem Problem P_1 für $\chi = 2$ und $\chi = 20$ relativ stark. Für $\chi = 20$ verändern sich die Pheromonwerte auf P_1 zwischen den Generationen 100 und 250 nur relativ schwach, da das System im Wesentlichen zur stabilen Fixpunktmatrix konvergiert, die sich jedoch aufgrund der schwachen Konkurrenz in dieser Phase des Laufs nur wenig verändert.

Die Wahrscheinlichkeiten für unterschiedliche Malussituationen im ACO-Modell und die Häufigkeit von Malussituationen, wie sie beim entsprechenden ACO-Algorithmus beobachtet wurden, sind in Abbildung 7.23 dargestellt. Beim ACO-Algorithmus wurde für jeweils 1000 aufeinander folgende Generationen analysiert, welche Lösungsqualitätsunterschiede beobachtet wurden.

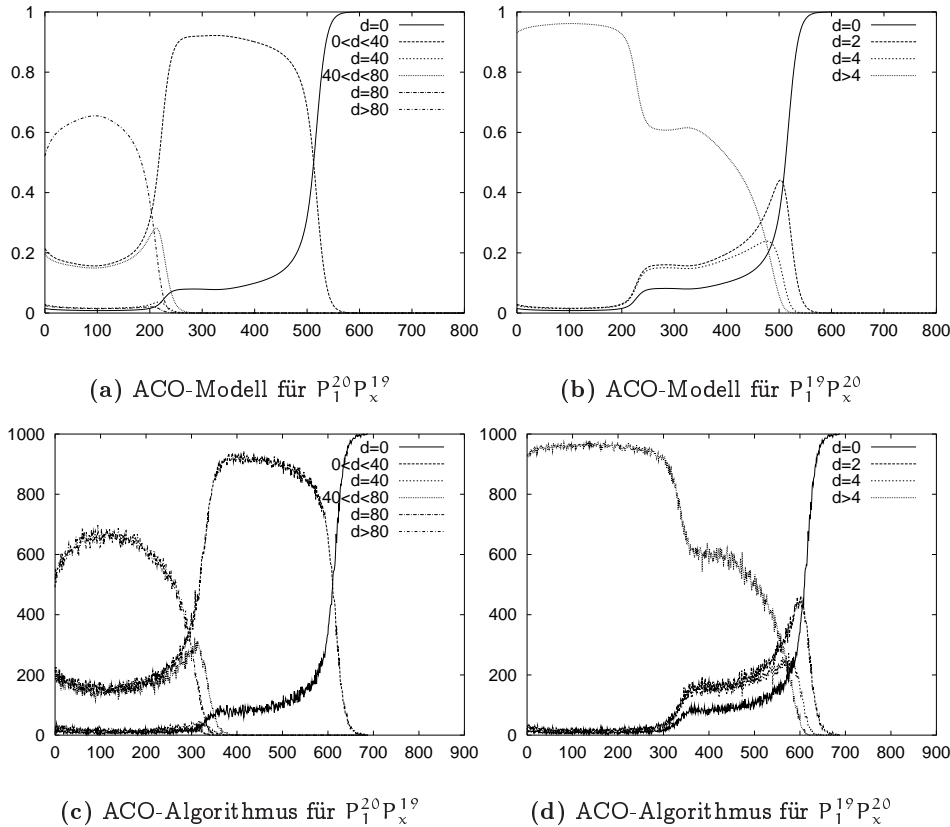


Abbildung 7.23: ACO-Modell (oben) und ACO-Algorithmus (unten) für $x = 20$; links : Veränderung der Wahrscheinlichkeiten $\phi_0^{(2)}$, $\phi_{0 < d < 40}^{(2)}$, $\phi_{d=40}^{(2)}$, $\phi_{40 < d < 80}^{(2)}$, $\phi_{d=80}^{(2)}$, $\phi_{d > 80}^{(2)}$ auf $P_1^{20}P_x^{19}$; rechts: Veränderung der Wahrscheinlichkeiten $\phi_0^{(2)}$, $\phi_2^{(2)}$, $\phi_{d=4}^{(2)}$, $\phi_{d > 4}^{(2)}$ auf $P_1^{19}P_x^{20}$; beim ACO-Algorithmus entspricht der abgetragene Wert der Anzahl der beobachteten Differenzen in jeweils 1000 Generationen (im Gegensatz zu den Wahrscheinlichkeiten beim ACO-Modell)

Die angegebenen Wahrscheinlichkeiten des ACO-Modells sind die Wahrscheinlichkeiten für bestimmte Malussituationen, die auf den Problemen $P_1^{19}P_x^{20}$ (für das Problem P_1) und $P_1^{20}P_x^{19}$ (für das Problem P_x) entstehen. Die möglichen Qualitätsstufen auf dem Problem P_1 (P_x) sind 0, 2 und 4 (und entsprechend 0, 40 und 80). Abbildung 7.23 zeigt die Wahrscheinlichkeiten $\phi_0^{(2)}$, $\phi_2^{(2)}$, $\phi_{d=4}^{(2)}$, $\phi_{d > 4}^{(2)}$ für $P_1^{19}P_x^{20}$ und $\phi_0^{(2)}$, $\phi_{0 < d < 40}^{(2)}$, $\phi_{d=40}^{(2)}$, $\phi_{40 < d < 80}^{(2)}$, $\phi_{d=80}^{(2)}$, $\phi_{d > 80}^{(2)}$ für $P_1^{20}P_x^{19}$. Man beachte, dass im Gegensatz zum entsprechenden homogenen Problem für 2 Ameisen, auf dem Problem $P_1^{20}P_x^{19}$ nicht 4 sondern 6 Malussituationen zu unterscheiden sind. Denn durch die heterogene Struktur des Problems $P_1^{20}P_x^{19}$ können für das Problem P_x Malussituationen entstehen, bei denen die Lösungs-

qualitätsunterschiede zwischen den möglichen Lösungsqualitäten des elementaren Teilproblems P_x liegen. In unserem Beispiel sind das die zwei zusätzlich möglichen Malussituationen, bei denen der Lösungsqualitätsunterschied der beiden Ameisen $0 < d < 40$ oder $40 < d < 80$ ist. Auf dem Problem P_1 kann es dagegen nicht vorkommen, dass Malussituationen entstehen, die zwischen den möglichen Lösungsqualitäten des elementaren Teilproblems P_1 liegen. In unserem Beispiel kann es nicht sein, dass bei der Betrachtung des letzten Teilproblems P_1 eine Malussituation $0 < d < 2$ oder $2 < d < 4$ vorliegt, da Kosten 1 oder 3 nicht entstehen können.

Abbildung 7.23 zeigt, dass auf $P_1^{19}P_x^{20}$ während der ersten 200 Generationen nur mit sehr geringer Wahrscheinlichkeit Konkurrenz zwischen den Ameisen entsteht ($\phi_{d>4}^{(2)} \approx 95\%$). Für $P_1^{20}P_x^{19}$ ist während der ersten ca. 200 Generationen die Wahrscheinlichkeit, dass keine Konkurrenz zu beobachten ist, ca. 60% ($\phi_{d>80}^{(2)} \approx 60\%$). Die Wahrscheinlichkeit für Konkurrenzsituationen, bei denen mehrere Lösungsqualitätsunterschiede (einerseits $0 < d < 40$ und andererseits $40 < d < 80$) zu gleichen Auswahlwahrscheinlichkeiten führen, und die daher zu Malussituationen zusammengefasst werden können (vgl. Abbildung 7.2 auf Seite 145), beträgt in den ersten 200 Generationen jeweils ca. 20%. Nach ca. 200 Generationen konvergiert die Pheromonmatrix für das Problem P_x^{20} mehr und mehr zur optimalen Lösung, und die Wahrscheinlichkeiten für eine Konkurrenzsituation auf $P_1^{19}P_x^{20}$ steigt mehr und mehr an. Nach ca. 250 Generationen sind die Pheromonwerte für P_x^{20} nahezu im konvergierten Stadium ($\phi_0^{(2)} + \phi_{0<d<40}^{(2)} \approx 100\%$). Zum gleichen Zeitpunkt ist die Wahrscheinlichkeit für eine reine nicht Malus-behaftete Konkurrenzsituation (d.h. $d = 0$) auf dem Problem P_1 ca. 10%. Die Wahrscheinlichkeit für eine Malus-behaftete Konkurrenzsituation ($d = 2$ oder $d = 4$) beträgt zusammen ca. 30%. Nach ca. 550 Generationen sind auch die Pheromonwerte für P_1^{20} nahezu konvergiert, und die Wahrscheinlichkeit die optimale Lösung zu finden, beträgt $\approx 100\%$.

7.8.3 Skalierung und Maskierung

In diesem Abschnitt werden die Implikationen der Ergebnisse des letzten Abschnitts für heterogene Permutationsprobleme bezüglich der Auswirkungen und Modifikationen bei schwierigen Optimierungsproblemen diskutiert. Es werden Techniken vorgeschlagen, die das Optimierungsverhalten von ACO-Algorithmen verbessern können.

Skalierung

Es wurde gezeigt, dass Skalierungseffekte eine wichtige Rolle beim Optimierungsverhalten von ACO-Algorithmen spielen. Das ACO-Modell konvergiert beim Problem $P_1^{20}P_x^{20}$ für $x = 1$ schneller als für $x = 20$, da im Fall $x = 20$ die Pheromonwerte des elementaren Teilproblems P_1^{20} zu einem sich nahezu nicht verändernden Fixpunktvektor streben. Erst durch die Konvergenz auf P_x^{20} tritt eine Konkurrenzsituation auf P_1^{20} ein. Dies stellt die Frage in den Raum, ob es möglich ist, allgemeine Skalierungsmethoden für ACO-Algorithmen zu entwerfen, die zur Verbesserung des Optimierungsverhaltens beitragen können. Hierfür kann man in einem Vorverarbeitungsschritt die Problem Instanz durch Skalierung von unabhängigen Teilproblemen der Problem Instanz verändern. Für komplexere Optimierungsprobleme muss natürlich in Betracht gezogen werden, dass die Identifikation von Teilproblemen im Allgemeinen schwierig ist, und die Teilprobleme im Allgemeinen nicht unabhängig voneinander sind, was eine mögliche Skalierung einschränkt.

Maskierung

Wir haben gezeigt, dass das Optimierungsverhalten von ACO-Algorithmen auf heterogenen Permutationsproblemen nur auf Teilen einer Problem Instanz auf Konkurrenz zwischen Ameisen beruht. Auf anderen Teilen verhält sich der ACO-Algorithmus so, als ob keine Konkurrenz zwischen den Ameisen herrscht, da die Wahrscheinlichkeit für eine entsprechende Malussituation sehr groß ist. Dieses Verhalten ist offensichtlich unerwünscht, da wir eine Verzerrung der Auswahlwahrscheinlichkeiten aufgrund des starken Einflusses von stabilen, als Attraktoren wirkenden, Fixpunktvektoren beobachten. Es taucht die Frage auf, ob es besser ist, die entsprechenden Teile der Pheromonmatrix zu maskieren, d.h. die entsprechenden Teile der Pheromonmatrix nicht zu verändern, solange die Konkurrenz zu schwach ist. Anhand des folgenden Beispiels, bei dem das Optimierungsverhalten signifikant durch eine solche Maskierung verbessert wird, werden wir die Frage positiv beantworten. Wir führen das Konzept der Maskierung von Teilen der Pheromonmatrix ein. Auf den entsprechenden Teilen der Pheromonmatrix werden die Pheromonwerte über eine gewisse Zeitspanne hinweg nicht verändert — weder durch Verdunstung noch durch Aktualisierung. Betrachten wir beispielsweise das Problem $P_1^{20}P_x^{20}$ mit $x = 20$. Wir initialisieren die Pheromonmatrix auf den P_x entsprechenden Teilen wieder mit den in Gleichung (7.21) auf Seite 147 gegebenen Werten. Auf den P_1 entsprechenden Teilen wählen wir exemplarisch als Initialisierung

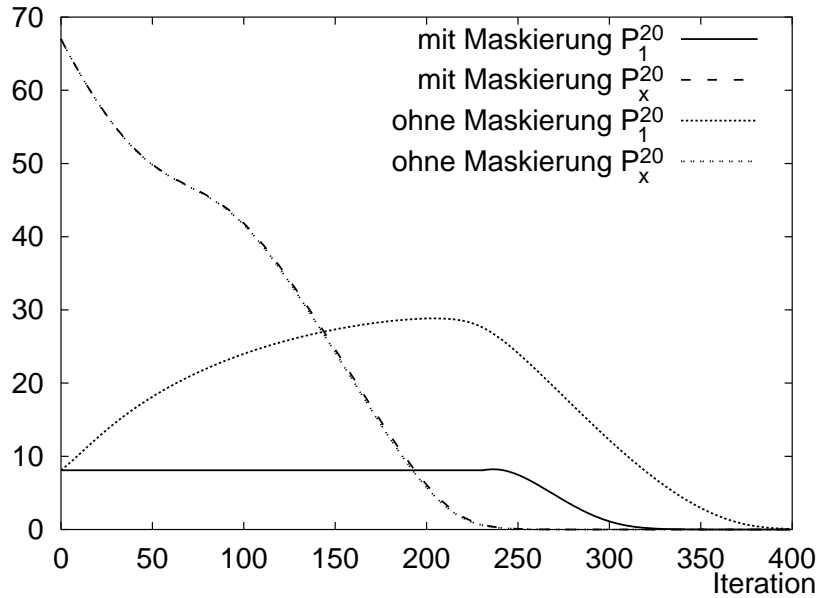


Abbildung 7.24: ACO-Modell mit und ohne Pheromonmaskierung für das Permutationsproblem $P_1^{20}P_x^{20}$ mit $x = 20$; maskiert wurde das Problem P_1^{20} in den ersten 230 Generationen; Initialisierung für P_x : $\tau_{11} = 0.1$, $\tau_{12} = 0.3$, $\tau_{21} = 0.6$ und $\tau_{22} = 0.1$; Initialisierung für P_1 : $\tau_{11} = 0.9$, $\tau_{12} = 0.09$, $\tau_{21} = 0.05$, $\tau_{22} = 0.9$; dargestellt ist die durchschnittliche Lösungsqualität für die Probleme P_1^{20} und P_x^{20} über 400 Iterationen hinweg (man beachte, dass die 2 Kurven für die durchschnittliche Lösungsqualität auf P_x^{20} nahezu identisch sind); die durchschnittliche Lösungsqualität für P_x^{20} wurde durch 20 geteilt

$\tau_{11} = 0.9$, $\tau_{12} = 0.09$, $\tau_{21} = 0.05$ und $\tau_{22} = 0.9$. Die entsprechende stabile Fixpunktmatrix ist (vgl. Abschnitt 7.5 ab Seite 154)

$$\begin{pmatrix} 0.9 & 0.09 & 0.01 \\ 0.0 & 0.1 & 0.9 \\ 0.1 & 0.81 & 0.09 \end{pmatrix} \quad (7.30)$$

Die durchschnittlich erreichten Lösungsqualitäten des ACO-Modells auf den Problemen P_1^{20} und P_x^{20} sind in Abbildung 7.24 sowohl mit als auch ohne Maskierung des Teilproblems P_1^{20} dargestellt. Dabei wurde die Lösungsqualität auf P_x^{20} mit durch x geteilt. Die Maskierung wurde in den ersten 230 Generationen durchgeführt. Ohne Maskierung wird die Lösungsqualität auf dem Teilproblem P_1^{20} in den ersten Generationen immer schlechter. Der Grund hierfür ist, dass die der Initialisierungsmatrix entsprechende Fixpunktmatrix zu einer deutlich schlechteren durchschnittlichen Lösungsqualität führt, und

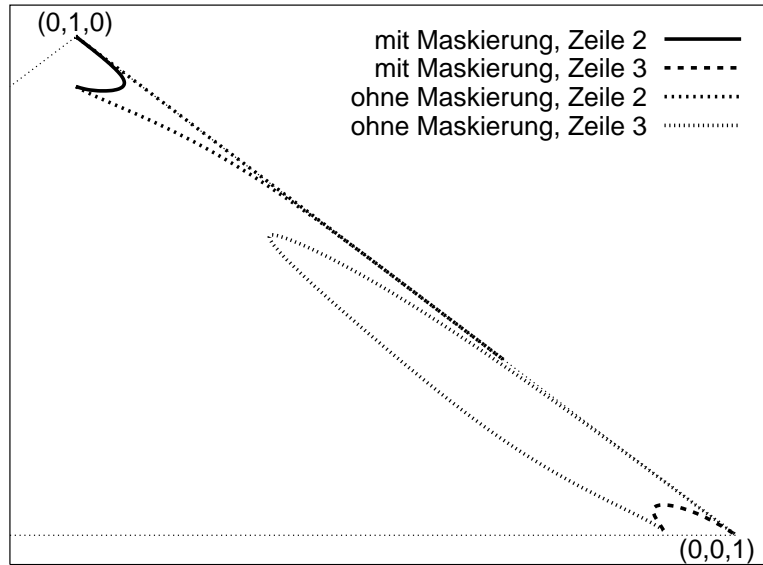


Abbildung 7.25: ACO-Modell mit und ohne Pheromonmaskierung für das Permutationsproblem $P_1^{20}P_x^{20}$ mit $x = 20$; maskiert wurde das Problem P_1^{20} in den ersten 230 Generationen; Initialisierung für P_x : $\tau_{11} = 0.1$, $\tau_{12} = 0.3$, $\tau_{21} = 0.6$ und $\tau_{22} = 0.1$; Initialisierung für P_1 : $\tau_{11} = 0.9$, $\tau_{12} = 0.09$, $\tau_{21} = 0.05$, $\tau_{22} = 0.9$; dargestellt ist die Veränderung der Pheromonwerte in der zweiten und dritten Zeile (nur die rechte Hälfte des Dreiecks aller möglichen Pheromonvektoren ist dargestellt)

die Fixpunktvektoren aufgrund der schwachen Konkurrenz auf den zu P_1 gehörenden Teilen der Pheromonmatrix als starker Attraktor für die Pheromonwerte wirken (siehe Abbildung 7.25). In Abbildung 7.25 ist deutlich zu erkennen, dass beispielsweise der Pheromonvektor der zweiten Zeile ausgehend vom Punkt $(\tau_{21}, \tau_{22}, \tau_{23}) = (0.05, 0.9, 0.05)$ stark in Richtung des Fixpunktvektors $(0, 0.1, 0.9)$ der zweiten Zeile gezogen wird, und der Pheromonvektor dann ungefähr entlang der durch $\tau_{21} = 0$ definierten rechten Linie des Dreiecks zur Optimallösung der zweiten Zeile $(\tau_{21}, \tau_{22}, \tau_{23}) = (0, 1, 0)$ konvergiert. Ebenso entwickeln sich die Werte der dritten Zeile bis zur Phase einer höheren Konkurrenz in Richtung des stabilen Fixpunktvektors $(0.1, 0.81, 0.09)$. Ein weiterer Effekt der Maskierung ist, dass das ACO-Modell durch die Maskierung schneller zur optimalen Lösung konvergiert.

7.9 Zusammenfassung

In diesem Kapitel wurde ein deterministisches Modell für ACO-Algorithmen vorgestellt und analysiert, bei dem die Auswahlwahrscheinlichkeiten und die Aktualisierungswerte mit Hilfe der im ACO-Algorithmus erwarteten Auswahlwahrscheinlichkeiten berechnet werden. Die Dynamik des Modells basiert somit, im Gegensatz zu ACO-Algorithmen, nicht auf einzelnen Zufallsentscheidungen. Eine interessante Eigenschaft des Modells ist, dass das Verhalten von ACO-Algorithmen auf eingeschränkten Permutationsproblemen durch eine Kombination von Situationen mit jeweils unterschiedlich starker Konkurrenz der Ameisen untereinander vorhergesagt werden kann.

Das Verhalten des ACO-Modells wurde analytisch untersucht. Eine Fixpunktanalyse basierend auf Pheromonwerten erklärt die in den vorherigen Kapiteln empirisch untersuchte ungünstige Verzerrung der Auswahlwahrscheinlichkeiten. Bei eingeschränkten Permutationsproblemen ist auf einem beliebigen elementaren Teilproblem mit hoher Wahrscheinlichkeit keine Konkurrenzsituation zwischen Ameisen gegeben. Daher haben die als Attraktoren wirkenden Fixpunktvektoren einen starken Einfluss auf die Dynamik der Pheromonwerte und somit auf das Optimierungsverhalten.

Die Auswahlwahrscheinlichkeit, dass die beste aller Ameisen auf einem Problem P eine bestimmte Entscheidung trifft, lässt sich im ACO-Modell durch eine gewichtete Summe der Wahrscheinlichkeiten für eine bestimmte Entscheidung über alle möglichen Qualitätsstufen beschreiben. Die Gewichte sind die Wahrscheinlichkeiten, dass die beste aller Ameisen die entsprechende Qualitätsstufe erreicht. Es wurde analytisch gezeigt, dass die Anzahl der unabhängigen Teilprobleme des eingeschränkten Permutationsproblems P^q das Optimierungsverhalten auf einem elementaren Teilproblem von P^q nur durch eine Veränderung der Gewichte beeinflusst. Sowohl für homogene als auch für heterogene Permutationsprobleme wurde dadurch die Dynamik des ACO-Modells numerisch exakt berechnet. Anhand von Simulationen wurde gezeigt, dass das ACO-Modell entscheidende Eigenschaften der Dynamik von ACO-Algorithmen sehr gut beschreibt. Es wurden auch Grenzen des Modells aufgezeigt und Aspekte von ACO-Algorithmen erwähnt, die durch die Modellbeschreibung nicht repräsentiert werden können.

Im letzten Teil dieses Kapitels wurden heterogene Permutationsprobleme analysiert, und das Modellverhalten mit dem Verhalten des ACO-Algorithmus verglichen. Auch hier zeigte sich, dass das ACO-Modell die Dynamik von ACO-Algorithmen sehr gut vorhersagt. Zur Verbesserung des Optimierungs-

verhaltens auf heterogenen Optimierungsproblemen wurden die Konzepte Skalierung und Maskierung vorgeschlagen.

Kapitel 8

Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurden Weiterentwicklungen herkömmlicher Ameisenalgorithmen vorgestellt und auf unterschiedlichen Ebenen analysiert. Die Algorithmen wurden zunächst auf einfachen Problemen untersucht, die charakteristische Eigenschaften schwieriger Probleme widerspiegeln. Ein grundlegendes Prinzip von Ameisenalgorithmen ist, dass die Pheromone ein Maß für die Qualität früherer Entscheidungen sein sollen. Anhand der Untersuchungen auf einfachen Problemen wurde jedoch gezeigt, dass die Pheromoninformation im Allgemeinen einer systematischen Verzerrung unterliegt, und dass dies zu einem schlechten Optimierungsverhalten führt. Bei der bisher üblichen Pheromonauswertungsmethode, die zur Lösungskonstruktion die Pheromonwerte in einem lokalen Sinne verwendet, werden die Auswahlwahrscheinlichkeiten von Einzelentscheidungen besonders stark verzerrt. Mit der Summen-Pheromonauswertung und der relativen Pheromonauswertung, welche die Pheromoninformation in einem globalen Sinne verwenden, konnte dieser unerwünschte Effekt deutlich verkleinert werden. Während bei herkömmlichen Ameisenalgorithmen selten ein Augenmerk auf die Entscheidungsreihenfolge gelegt wurde, die eine Ameise beim Aufbau einer Lösung durchläuft, konnte auch hier durch Verfahrensmodifikationen eine deutlich geringere Verzerrung und somit ein deutlich besseres Optimierungsverhalten erzielt werden. Die verbesserte Nutzung der Pheromoninformation zur Lösungskonstruktion wurde kombiniert mit Erweiterungen, welche zusätzlich strukturelle Anpassungen durch eine gezielte Phasenstrukturierung des Algorithmus vornehmen. Der resultierende Ameisenalgorithmus konnte neue beste Lösungen für einige der größten und am häufigsten verwendeten Testprobleme für die Projektplanung

unter beschränkten Ressourcen finden und lieferte bei einem systematischen Vergleich mit den mehr als einem Dutzend anderen bekannten Verfahren die besten Ergebnisse. Für praxisnahe Optimierungsprobleme mit mehreren Optimierungskriterien wurden Kooperationsmechanismen zwischen künstlichen Ameisenkolonien entwickelt und anhand eines bikriteriellen Planungsproblems erfolgreich umgesetzt.

Ausgehend vom empirisch nachgewiesenen Erfolg der neuen Methoden sowohl auf NP-vollständigen als auch auf einfachen Problemen, wurde ein deterministisches Modell für Ameisenalgorithmen eingeführt, welches unter anderem analytische Aussagen zum dynamischen Verhalten ermöglicht. Die insbesondere für große Optimierungsprobleme ungünstige Verzerrung der Auswahlwahrscheinlichkeiten wurde mit Hilfe einer auf Pheromonwerten basierenden Fixpunktanalyse erklärt. Es wurde untersucht, wie die Anzahl unabhängiger Teilprobleme eines Permutationsproblems die Zusammensetzung des Systems hinsichtlich unterschiedlicher Stufen der Konkurrenz zwischen Ameisen, und somit auch das Optimierungsverhalten, beeinflusst. Anhand von Simulationen wurde gezeigt, dass das numerisch exakte Modell auch für komplexe Optimierungsprozesse eine sehr genaue Vorhersage des zufallsgesteuerten Verhaltens von Ameisenalgorithmen ermöglicht.

Für weiterführende Untersuchungen bieten sich zahlreiche Möglichkeiten an, von denen hier einige der wichtigsten erwähnt werden sollen.

- Die Summen-Pheromonauswertung und die Methode einer angepassten Entscheidungsreihenfolge wurden anhand umfangreicher empirischer Untersuchungen auf dem Problem der Projektplanung unter beschränkten Ressourcen sehr erfolgreich angewendet. Die relative Pheromonauswertung liefert hinsichtlich der Verzerrung der Auswahlwahrscheinlichkeiten und den weiteren Untersuchungen dieser Arbeit sehr viel versprechende Ergebnisse. Interessant erscheint zum einen die Identifikation weiterer Methoden zur gezielten Nutzung der Pheromoninformation für bestimmte Problemklassen, und zum anderen der auch noch für die relative Pheromonauswertung ausstehende Vergleich mit Verfahren, die beste Ergebnisse auf intensiv untersuchten, standardisierten Problemstellungen erzielen.
- Ausgehend von den in dieser Arbeit vorgestellten Methoden für bikriterielle Optimierungsprobleme, die keine Ordnung der Optimierungskriterien bezüglich deren Wichtigkeit voraussetzen, bieten sich die Einführung einer Elitestrategie und Methoden zur gesteuerten Erhaltung der

Diversität der Lösungen entlang der nicht-dominierten Front an. Insbesondere ist auch die Erweiterung der in dieser Arbeit vorgestellten Verfahren auf mehr als zwei Optimierungskriterien interessant, da die Kommunikationsmechanismen der Kolonien und die Aufteilung der höherdimensionalen nicht-dominierten Front nichttriviale Erweiterungen bedingen.

- Mit dem deterministischen Modell für Ameisenalgorithmen wurde ein Grundstein für eine Reihe weiterer theoretischer Untersuchungen gelegt. Ausgehend von der modellbasierten Analyse einfacher Permutationsprobleme, steht die Erweiterung für komplexere Problemstellungen aus. Hier sind beispielsweise Probleme interessant, bei denen Abhängigkeiten zwischen den Teilproblemen bestehen. Des Weiteren stellt sich die Frage, wie sich nachweislich erfolgreiche Methoden, wie z.B. die Verwendung einer Elitestrategie, die Einführung einer minimalen Untergrenze für Pheromonwerte oder auch die unterschiedlichen in dieser Arbeit vorgestellten Verfahrenserweiterungen, die nur teilweise im Modell untersucht wurden, hinsichtlich der numerisch exakten Vorhersage verhalten. Ein nächster sehr wichtiger Schritt ist eine mathematisch exakte Abschätzung des Modellfehlers und die Untersuchung des Einflusses der Algorithmusparameter auf die Fehlerabschätzung.

Literaturverzeichnis

[Bäck et al. 1997] BÄCK, T. ; FOGEL, D.B. ; MICHALEWICZ, Z. (Herausgeber): *Handbook of Evolutionary Computation*. Bristol, New York : Institute of Physics Publishing and Oxford University Press, 1997

[Banzhaf et al. 1999] BANZHAF, W. ; DAIDA, J. ; EIBEN, A.E. ; GARZON, M.H. ; HONAVAR, V. ; JAKIELA, M. ; SMITH, R.E. (Herausgeber): *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999)*. San Francisco, CA, USA : Morgan Kaufmann, 1999

[Bauer et al. 1999] BAUER, A. ; BULLNHEIMER, B. ; HARTL, R.F. ; STRAUSS, C.: An ant colony optimization approach for the single machine total tardiness problem. In: *Proceedings of the 1999 Congress on Evolutionary Computation (CEC 1999)*, 1999, S. 1445–1450

[Bautista und Pereira 2002] BAUTISTA, J. ; PEREIRA, J.: Ant algorithms for assembly line balancing. In: (Dorigo et al., 2002), S. 65–75

[Bäck et al. 1997] BÄCK, T. ; MICHALEWICZ, Z. ; YAO, X. (Herausgeber): *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC'97)*. Piscataway, NJ, USA : IEEE Press, 1997

[Beasley 1990] BEASLEY, J.E.: OR-Library: distributing test problems by electronic mail. In: *Journal of the Operations Research Society* 41 (1990), S. 1069–1072

[Beckers et al. 1992] BECKERS, R. ; DENEUBOURG, J.-L. ; GOSS, S.: Trails and U-turns in the selection of the shortest path by the ant *Lasius Niger*. In: *Journal of Theoretical Biology* 159 (1992), S. 397–415

[Beni 1988] BENI, G.: Concept of cellular robotic systems. In: *Proceeding of the IEEE Int. Symposium on Intelligent Control*, 1988, S. 57–62

- [Beni und Wang 1989] BENI, G. ; WANG, J.: Swarm intelligence in cellular robotic systems. In: *Proceedings of the NATO Advanced Workshop on Robotics and Biological Systems*, 1989
- [den Besten et al. 1999] BESTEN, M. den ; STÜTZLE, T. ; DORIGO, M.: Scheduling single machines by ants / IRIDIA, Université Libre de Bruxelles, Belgien. 1999 (IRIDIA/99-16). – Forschungsbericht
- [den Besten et al. 2000] BESTEN, M. den ; STÜTZLE, T. ; DORIGO, M.: Ant colony optimization for the total weighted tardiness problem. In: (Schwefel et al., 2000), S. 611–620
- [Beyer et al. 2002] BEYER, H.-G. ; SCHWEFEL, H.-P. ; WEGENER, I.: How to analyse evolutionary algorithms. In: *Theoretical Computer Science* 287 (2002), S. 101–130
- [Blum et al. 2001] BLUM, C. ; ROLI, A. ; DORIGO, M.: HC-ACO: The hyper-cube framework for ant colony optimization. In: *Proceedings of the 4th Metaheuristics International Conference (MIC 2001)*, 2001, S. 399–403
- [Blum und Sampels 2002] BLUM, C. ; SAMPELS, M.: Ant colony optimization for FOP shop scheduling: A case study on different pheromone representations. In: *Proceedings of the Congress on Evolutionary Computation (CEC 2002)*, 2002, S. 1558–1563
- [Boers et al. 2001] BOERS, E.J.W. ; GOTTLIEB, J. ; LANZI, P.L. ; SMITH, R.E. ; CAGNONI, S. ; E.HART ; RAIDL, G.R. ; TIJINK, H. (Herausgeber): *Applications of Evolutionary Computing, Proceedings of the EvoWorkshops 2001*. Berlin : Springer, 2001. (Lecture Notes in Computer Science (LNCS) 2037)
- [Bolondi und Bondanza 1993] BOLONDI, M. ; BONDANZA, M.: *Parallelizzazione di un algoritmo per la risoluzione del problema del commesso viaggiatore*, Dipartimento di Elettronica, Politecnico di Milano, Italien, Diplomarbeit, 1993
- [Bonabeau et al. 1999] BONABEAU, E. ; DORIGO, M. ; THÉRAULAZ, G.: *Swarm Intelligence*. Oxford University Press, 1999
- [Bottee und Bonabeau 1998] BOTTEE, H.M. ; BONABEAU, E.: Evolving ant colony optimization. In: *Adv. Complex Systems* 1 (1998), S. 149–159

[Bouleimen und Lecocq 1998] BOULEIMEN, K. ; LECOCQ, H.: A new efficient simulated annealing algorithm for the resource constrained project scheduling problem / Service de Robotique et Automatisation, Université de Liège. 1998. – Forschungsbericht

[Branke 2002] BRANKE, J.: *Evolutionary Optimization in Dynamic Environments*. Boston : Kluwer Academic Publishers, 2002

[Brucker et al. 1999] BRUCKER, P. ; DREXEL, A. ; MÖHRING, R.H ; NEUMANN, K. ; PESCH, E.: Resource-constraint project scheduling: Notation, classification, models, and methods. In: *European Journal of Operations Research* 112 (1999), S. 3–41

[Bullnheimer et al. 1999] BULLNHEIMER, B. ; HARTL, R. ; STRAUSS, C.: A new rank based version of the ant system — a computational study. In: *Central European Journal for Operations Research and Economics* 7 (1999), Nr. 1, S. 25–38

[Cagnoni et al. 2002] CAGNONI, S. ; GOTTLIEB, J. ; HART, E. ; MIDDENDORF, M. ; RAIDL, G.R. (Herausgeber): *Applications of Evolutionary Computing, Proceedings of the EvoWorkshops 2002*. Berlin : Springer, 2002. (Lecture Notes in Computer Science (LNCS) 2279)

[Cagnoni et al. 2000] CAGNONI, S. ; POLI, R. ; LI, Y. ; SMITH, G. ; CORNE, D. ; OATES, M.J. ; HART, E. ; LANZI, P.L. ; BOERS, E.J.W. ; PAECHTER, B. ; FOGARTY, T.C. (Herausgeber): *Real-World Applications of Evolutionary Computing, Proceedings of the EvoWorkshops 2000*. Berlin : Springer, 2000. (Lecture Notes in Computer Science (LNCS) 1803)

[Colorni et al. 1994] COLORNI, A. ; DORIGO, M. ; MANIEZZO, V. ; TRUBIAN, M.: Ant system for job-shop scheduling. In: *JORBEL - Belgian Journal of Operations Research, Statistics and Computer Science* 34 (1994), S. 39–53

[Cordón et al. 2000] CORDÓN, O. ; VIANA, I. F. de ; HERRERA, F. ; MORENO, L.: A new ACO model integrating evolutionary computation concepts: The best-worst ant system. In: (Dorigo et al., 2000), S. 22–29

[Corne et al. 1999] CORNE, D. ; DORIGO, M. ; GLOVER, F. (Herausgeber): *New Ideas in Optimization*. London : McGraw-Hill, 1999

- [Crauwels et al. 1998] CRAUWELS, H.A.J. ; POTTS, C.N. ; WASSENHOVE, L.N. van: Local search heuristics for the single machine total weighted tardiness scheduling problem. In: *INFORMS Journal on Computing* 10 (1998), S. 341–359
- [Deb 2001] DEB, K.: *Multi-objective optimization using evolutionary algorithms*. Chichester, UK : Wiley, 2001
- [Deneubourg et al. 1990] DENEUBOURG, J.-L ; ARON, S. ; GOSS, S. ; PASTEBELS, J.M.: The self-organizing exploratory pattern of the argentine ant. In: *Journal of Insect Behavior* 3 (1990), S. 159–168.
- [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
- [Dorigo 1992] DORIGO, M.: *Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale (Optimization, learning and natural algorithms)* (in italienisch), Dipartimento di Elettronica , Politecnico di Milano, Italien, Dissertation, 1992. – 140 Seiten
- [Dorigo et al. 1999] DORIGO, M. ; CARO, G. D. ; GAMBARDELLA, L. M.: Ant algorithms for discrete optimization. In: *Artificial Life* 5 (1999), Nr. 2, S. 137–172
- [Dorigo und Di Caro 1999] DORIGO, M. ; DI CARO, G.: New Ideas in Optimization. In: (Corne et al., 1999), Kap. The ant colony optimization meta-heuristic, S. 11–32
- [Dorigo et al. 2002] DORIGO, M ; DI CARO, G. ; SAMPELS, M. (Herausgeber): *Proceedings of 3rd International Workshop on Ant Algorithms (ANTS'2002)*. Berlin : Springer, 2002. (Lecture Notes in Computer Science (LNCS))
- [Dorigo und Gambardella 1997a] DORIGO, M. ; GAMBARDELLA, L.M.: Ant colonies for the traveling salesman problem. In: *BioSystems* 43 (1997), S. 73–81
- [Dorigo und Gambardella 1997b] DORIGO, M. ; GAMBARDELLA, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. In: *IEEE Transactions on Evolutionary Computation* 1 (1997), S. 53–66

- [Dorigo et al. 1991a] DORIGO, M. ; MANIEZZO, V. ; COLORNI, A.: The ant system: An autocatalytic optimization process / Dipartimento di Elettronica e Informatica, Politecnico di Milano, Italien. 1991 (91016 Revised). – Forschungsbericht.
- [Dorigo et al. 1991b] DORIGO, M. ; MANIEZZO, V. ; COLORNI, A.: Positive feedback as a search strategy / Dipartimento di Elettronica e Informatica, Politecnico di Milano, Italien. 1991 (91016). – Forschungsbericht.
- [Dorigo et al. 2000] DORIGO, M. ; MIDDENDORF, M. ; STÜTZLE, T. (Herausgeber): *Abstract Proceedings of the Second International Workshop on Ant Algorithms (ANTS'2000)*. 2000
- [Dorigo und Gambardella 1996] DORIGO, Marco ; GAMBARDELLA, Luca M.: A study of some properties of Ant-Q. In: (Voigt et al., 1996), S. 656–665
- [Dorndorf et al. 2000] DORNDORF, U. ; PESCH, E. ; HUY, T. P.: A branch and bound algorithm for the resource-constrained project scheduling problem. In: *Mathematical Methods in Operations Research* 51 (2000), S. 565–594
- [Du und Leung 1990] DU, J. ; LEUNG, J.Y.-T.: Minimizing the total tardiness on one machine is NP-hard. In: *Mathematics of Operations Research* 15 (1990), S. 483–496
- [Eiben et al. 1995] EIBEN, A.E. ; BÄCK, T. ; SCHOENAUER, M. ; SCHWEFEL, H.-P. (Herausgeber): *Proceedings of Parallel Problem Solving from Nature IV (PPSN-V)*. Berlin : Springer, 1995. (Lecture Notes in Computer Science (LNCS) 1498)
- [Elmaghraby 1977] ELMAGHRABY, S. E.: *Activity networks*. New York : Wiley, 1977
- [Fonseca und Fleming 1996] FONSECA, C.M. ; FLEMING, P.J.: On the performance assessment and comparison of stochastic multiobjective optimizers. In: (Voigt et al., 1996), S. 584–593
- [Gagné et al. 2000] GAGNÉ, C. ; GRAVEL, M. ; PRICE, W.L.: Scheduling a single machine where setup times are sequence dependent using an ant-colony heuristic. In: (Dorigo et al., 2000), S. 157–160

- [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), Nr. 3, S. 237–255
- [Gambardella und Dorigo 1995] GAMBARDELLA, L.M. ; DORIGO, M.: Ant-Q: A reinforcement learning approach to the traveling salesman problem. In: *Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann, 1995, S. 252–260
- [Gambardella und Dorigo 1996] GAMBARDELLA, L.M. ; DORIGO, M.: Solving symmetric and asymmetric TSP by ant colonies. In: *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC'96)*. Piscataway, NJ : IEEE Press, 1996, S. 622–627
- [Gambardella et al. 1999a] GAMBARDELLA, L.M. ; TAILLARD, É. ; AGAZZI, G.: MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In: (Corne et al., 1999), S. 63–76
- [Gambardella et al. 1999b] GAMBARDELLA, L.M. ; TAILLARD, E.D. ; DORIGO, M.: Ant colonies for the quadratic assignment problem. In: *Journal of the Operational Research Society* 50 (1999), Nr. 2, S. 167–176
- [Glover und Kochenberger 2002] GLOVER, F. ; KOCHENBERGER, G. (Herausgeber): *Handbook of Metaheuristics*. Boston : Kluwer Academic Publishers, 2002
- [Goldberg 1989] GOLDBERG, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA : Addison Wesley, 1989
- [Goss et al. 1989] GOSS, S. ; ARON, S. ; DENEUBOURG, J.-L. ; PASTEELS, J.: Self-organized shortcuts in the argentine ant. In: *Naturwissenschaften* 76 (1989), S. 579–581
- [Grassé 1959] GRASSÉ, P.P.: La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes Natalensis* et *Cubitermes* sp. la theorie de la stigmergie: essai d'interpretation du comportement des termites constructeurs. In: *Insectes Sociaux* 6 (1959), S. 41–81
- [Guntsch et al. 2001] GUNTSCH, M. ; MIDDENDORF, M. ; SCHMECK, H.: An ant colony optimization approach to dynamic TSP. In: (Spector et al., 2001), S. 860–867

- [Gutjahr 2000] GUTJAHR, W.: A graph-based ant system and its convergence. In: *Future Generations Computing Systems* 16 (2000), S. 873–888
- [Gutjahr 2002] GUTJAHR, W.: ACO algorithms with guaranteed convergence to the optimal solution. In: *Information Processing Letters* 82 (2002), Nr. 3, S. 145–153
- [Hartmann 1998] HARTMANN, S.: A competitive genetic algorithm for resource-constrained project scheduling. In: *Naval Research Logistics* 45 (1998), Nr. 7, S. 733–750
- [Hartmann 1999] HARTMANN, S.: Self-adapting genetic algorithm with an application to project scheduling / Universität zu Kiel. 1999 (506). – Forschungsbericht
- [Hartmann und Kolisch 2000] HARTMANN, S. ; KOLISCH, R.: Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. In: *European Journal of Operational Research* 127 (2000), S. 394–407
- [Holland 1975] HOLLAND, J.: *Adaptation in Natural and Artificial Systems*. Ann Arbor, Michigan : MIT Press, 1975
- [Iredi et al. 2001] IREDI, S. ; MERKLE, D. ; MIDDENDORF, M.: Bi-criterion optimization with multi colony ant algorithms. In: (Zitzler et al., 2001), S. 359–372
- [Jansen und Wegener 2002] JANSEN, T. ; WEGENER, I.: On the analysis of a dynamic evolutionary algorithm. In: *Journal of Discrete Algorithms* (2002). – eingereicht
- [Janson et al. 2002] JANSON, S. ; MERKLE, D. ; MIDDENDORF, M. ; ELGIN-DY, H. ; SCHMECK, H.: On enforced convergence of aco and its implementation on the reconfigurable mesh architecture using size reduction tasks. In: *Proceedings of the Second International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA'02)*, CSREA Press, 2002, S. 3–9
- [Kallel et al. 2000] KALLEL, L. ; NAUDTS, B. ; ROGERS, A. (Herausgeber): *Theoretical Aspects of Evolutionary Computing*. Berlin : Springer-Verlag, 2000

- [Karlson und Butenandt 1959] KARLSON, P. ; BUTENANDT, A.: Pheromones (ectohormones) in insects. In: *Annual Review of Entomology* 4 (1959), S. 39–58
- [Klein 2000] KLEIN, R.: Bidirectional planning: improving priority rule-based heuristics for scheduling resource-constrained projects. In: *European Journal of Operations Research* 127 (2000), S. 619–638
- [Kolisch 1996a] KOLISCH, R.: Efficient priority rules for the resource-constrained project scheduling problem. In: *Journal of Operations Management* 14 (1996), Nr. 3, S. 179–192
- [Kolisch 1996b] KOLISCH, R.: Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. In: *European Journal of Operational Research* 90 (1996), Nr. 2, S. 320–333
- [Kolisch und Drexel 1996] KOLISCH, R. ; DREXL, A.: Adaptive search for solving hard project scheduling problems. In: *Naval Research Logistics* 43 (1996), Nr. 1, S. 23–40
- [Kolisch und Hartmann 1999] KOLISCH, R. ; HARTMANN, S.: Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis. In: (Weglarz, 1999), S. 197–212
- [Kolisch et al. 1999] KOLISCH, R. ; SCHWINDT, C. ; SPRECHER, A.: Benchmark instances for project scheduling problems. In: (Weglarz, 1999), S. 147–178
- [Kolisch und Sprecher 1996] KOLISCH, R. ; SPRECHER, A.: PSPLIB - A project scheduling problem library. In: *European Journal of Operational Research* 96 (1996), S. 205–216
- [Kolisch 1995] KOLISCH, Rainer: *Project scheduling under resource constraints - Efficient heuristics for several problem classes*. Heidelberg : Physica-Verlag, 1995
- [Koopmans und Beckmann 1957] KOOPMANS, T. C. ; BECKMANN, M.: Assignment problems and the location of economic activities. In: *Econometrica* 25 (1957), S. 53–76
- [Kuhn 1955] KUHN, H.W.: The hungarian method for the assignment problem. In: *Naval Research Logistics Quarterly* 2 (1955), S. 83–97

- [Langdon et al. 2002] LANGDON, W.B. ; CANTU-PAZ, E. ; K.MATHIAS ; ROY, R. ; D.DAVIS ; POLI, R. ; BALAKRISHNAN, K. ; HONAVAR, V. ; RUDOLPH, G. ; WEGENER, J. ; BULL, L. ; POTTER, M.A. ; SCHULTZ, A.C. ; MILLER, J.F. ; BURKE, E. ; JONOSKA, N. (Herausgeber): *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*. San Francisco, CA, USA : Morgan Kaufmann, 2002
- [Langrock und Jahn 1979] LANGROCK, P. ; JAHN, W.: *Einführung in die Theorie der Markovschen Ketten und ihre Anwendung*. Leipzig : B.G. Teubner, 1979
- [Lawler et al. 1985] LAWLER, E. L. ; LENSTRA, J. K. ; RINNOOY KAN, A. H. G. ; SHMOYS, D. B.: *The traveling salesman problem*. Wiley, 1985
- [Lawler 1977] LAWLER, E.L.: A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness. In: *Annals of Discrete Mathematics* (1977), S. 331–342
- [Leon und Ramamoorthy 1995] LEON, V.J. ; RAMAMOORTHY, B.: Strength and adaptability of problem-space based neighborhoods for resource-constrained scheduling. In: *OR Spektrum* 17 (1995), S. 173–182
- [M. Zlochín und Dorigo 2001] M. ZLOCHIN, N. M. ; DORIGO, M.: Model-base search for combinatorial optimization / IRIDIA, Université Libre de Bruxelles, Belgien. 2001 (IRIDIA/2001-15). – Forschungsbericht
- [Maniezzo 1999] MANIEZZO, V.: Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. In: *INFORMS Journal on Computing* 11 (1999), Nr. 4, S. 358–369
- [Mariano und Morales 1999] MARIANO, Carlos E. ; MORALES, Eduardo: MOAQ an Ant-Q algorithm for multiple objective optimization problems. In: (Banzhaf et al., 1999), S. 894–901
- [Merkle und Middendorf 2000] MERKLE, D. ; MIDDENDORF, M.: An ant algorithm with a new pheromone evaluation rule for total tardiness problems. In: (Cagnoni et al., 2000), S. 287–296
- [Merkle und Middendorf 2001a] MERKLE, D. ; MIDDENDORF, M.: Collective optimization: The artificial ants way. In: *Abstracts of the Workshop From Worker to Colony: Understanding the Organisation of Insect Societies*. Cambridge, UK, Dezember 2001

- [Merkle und Middendorf 2001b] MERKLE, D. ; MIDDENDORF, M.: Fast ant colony optimization on reconfigurable processor arrays. In: *Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS-01)*, IEEE Computer Society Press, April 2001
- [Merkle und Middendorf 2001c] MERKLE, D. ; MIDDENDORF, M.: A new approach to solve permutation scheduling problems with ant colony optimization. In: (Boers et al., 2001), S. 484–493
- [Merkle und Middendorf 2001d] MERKLE, D. ; MIDDENDORF, M.: On the behaviour of ant algorithms: studies on simple problems. In: *Proceedings of the 4th Metaheuristics International Conference (MIC' 2001)*, Porto, Portugal, Juli 2001, S. 573–577
- [Merkle und Middendorf 2001e] MERKLE, D. ; MIDDENDORF, M.: Prospects for dynamic algorithm control: Lessons from the phase structure of ant scheduling algorithms. In: HECKENDORN, R.B. (Herausgeber): *Proceedings of the 2001 Genetic and Evolutionary Computation Conference – Workshop Program*. San Francisco, CA, USA : Morgan Kaufmann, 2001, S. 121–126
- [Merkle und Middendorf 2002a] MERKLE, D. ; MIDDENDORF, M.: An ant algorithm with global pheromone evaluation for scheduling a single machine. In: *Applied Intelligence*. 18 (2002), S. 105–111
- [Merkle und Middendorf 2002b] MERKLE, D. ; MIDDENDORF, M.: Ant colony optimization with the relative pheromone evaluation method. In: (Cagnoni et al., 2002), S. 325–333
- [Merkle und Middendorf 2002c] MERKLE, D. ; MIDDENDORF, M.: Fast ant colony optimization on runtime reconfigurable processor arrays. In: *Genetic Programming and Evolvable Machines* 3 (2002), Nr. 4, S. 345–361
- [Merkle und Middendorf 2002d] MERKLE, D. ; MIDDENDORF, M.: Modelling ACO: Composed permutation problems. In: (Dorigo et al., 2002), S. 149–162
- [Merkle und Middendorf 2002e] MERKLE, D. ; MIDDENDORF, M.: Modelling the dynamics of ant colony optimization. In: *Evolutionary Computation* 10 (2002), Nr. 3, S. 235–262. – vorläufige Version auch als Forschungsbericht Nr. 412, Institut AIFB, Universität Karlsruhe (TH)

- [Merkle und Middendorf 2002f] MERKLE, D. ; MIDDENDORF, M.: On Solving Permutation Scheduling Problems with Ant Colony Optimization / Institut AIFB, Universität Karlsruhe. 2002 (415). – Forschungsbericht. eingereicht beim *International Journal of Systems Science*
- [Merkle und Middendorf 2002g] MERKLE, D. ; MIDDENDORF, M.: Studies on the dynamics of ant colony optimization algorithms. In: (Langdon et al., 2002), S. 105–112
- [Merkle et al. 2000a] MERKLE, D. ; MIDDENDORF, M. ; SCHMECK, H.: Ant colony optimization for resource-constrained project scheduling. In: (Whitley et al., 2000), S. 893–900
- [Merkle et al. 2000b] MERKLE, D. ; MIDDENDORF, M. ; SCHMECK, H.: Ant colony optimization for the RCPSP using weighted summation evaluation. In: (Dorigo et al., 2000), S. 84–87
- [Merkle et al. 2000c] MERKLE, D. ; MIDDENDORF, M. ; SCHMECK, H.: Pheromone evaluation in ant colony optimization. In: *Proceedings of the 26th Annual Conference of the IEEE Electronics Society IECON-2000 (2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation), Third Asia-Pacific Conference on Simulated Evolution and Learning (SEAL2000)*. Piscataway, USA : IEEE Press, Oktober 2000, S. 2726–2731
- [Merkle et al. 2002] MERKLE, D. ; MIDDENDORF, M. ; SCHMECK, H.: Ant colony optimization for resource-constrained project scheduling. In: *IEEE Transactions on Evolutionary Computation* 6 (2002), Nr. 4, S. 333–346. – vorläufige Version auch als Forschungsbericht Nr. 409, Institut AIFB, Universität Karlsruhe (TH)
- [Michalewicz 1996] MICHALEWICZ, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1996. – Contains introductory chapter on LCS
- [Michels und Middendorf 1999] MICHELS, R. ; MIDDENDORF, M.: An ant system for the shortest common supersequence problem. In: (Corne et al., 1999), S. 51–61
- [Middendorf et al. 2002] MIDDENDORF, M. ; REISCHLE, F. ; SCHMECK, H.: Multi Colony Ant Algorithms. In: *Journal of Heuristics* 8 (2002), Nr. 3, S. 305–320

- [Mitchell und Forrest 1997] MITCHELL, M. ; FORREST, S.: Royal road functions. In: (Bäck et al., 1997), S. 20–23
- [Möhring et al. 2000] MÖHRING, R.H. ; SCHULZ, A.S. ; STORK, F. ; UETZ, M.: Solving project scheduling problems by minimum cut computations / Dept. of Mathematics, TU Berlin, Germany. 2000 (680-2000). – Forschungsbericht
- [van Nimwegen et al. 1999] NIMWEGEN, E. van ; CRUTCHFIELD, J.P. ; MITCHELL, M.: Statistical dynamics of the royal road genetic algorithm. In: *Theoretical Computer Science* 229 (1999), S. 41–102
- [Nonobe und Ibaraki 1999] NONOBE, K. ; IBARAKI, T.: Formulation and tabu search algorithm for the resource constrained project scheduling problem (RCPSP) / Department of Applied Mathematics and Physics, Kyoto University, Japan. 1999. – Forschungsbericht
- [Ow und Morton 1998] OW, P.S. ; MORTON, T.E.: The single machine early/tardy problem. In: *Management Science* 35 (1998), S. 177–191
- [Prügel-Bennett und Rogers 2000] PRÜGEL-BENNETT, A. ; ROGERS, A.: Modelling GA dynamics. In: (Kallel et al., 2000), S. 59–86
- [Prügel-Bennett und Shapiro 1997] PRÜGEL-BENNETT, A. ; SHAPIRO, J.L.: Analysis of genetic algorithms using statistical mechanics. In: *Physica D* 104 (1997), S. 75–114
- [Rajendran und Ziegler 2001] RAJENDRAN, C. ; ZIEGLER, H.: Ant-Colony Algorithms for Flowshop Scheduling. In: *European Journal of Operational Research* (2001). – eingereicht
- [Rubinstein 1999] RUBINSTEIN, R. Y.: The cross-entropy method for combinatorial and continuous optimization. In: *Methodology and Computing in Applied Probability* 1 (1999), S. 127–190
- [Schirmer 2000] SCHIRMER, A.: Case-based reasoning and improved adaptive search for project scheduling. In: *Naval Research Logistics* 47 (2000), S. 201–222
- [Schoonderwoerd et al. 1996] SCHOONDERWOERD, R. ; HOLLAND, O.E. ; BRUTEN, J.L. ; ROTHKRANTZ, L.J.M.: Ant-based load balancing in telecommunications networks. In: *Adaptive Behavior* 5 (1996), Nr. 2, S. 169–207

- [Schwefel et al. 2000] SCHWEFEL, H.-P. ; SCHOENAUER, M. ; DEB, K. ; RUDOLPH, G. ; YAO, X. ; LUTTON, E. ; MERELO, J.J. (Herausgeber): *Proceedings of Parallel Problem Solving from Nature VI (PPSN-VI)*. Berlin : Springer, 2000. (Lecture Notes in Computer Science (LNCS) 1917)
- [Shannon 1948] SHANNON, C. E.: A mathematical theory of communication. In: *Bell System Technical Journal* 27 (1948), S. 379–423, 623–656
- [Spector et al. 2001] SPECTOR, L. ; WHITLEY, D. ; GOLDBERG, D. ; CANTU-PAZ, E. ; PARMEE, I. ; BEYER, H.-G. (Herausgeber): *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*. San Francisco, CA, USA : Morgan Kaufmann, 2001
- [Sprecher et al. 1995] SPRECHER, A. ; KOLISCH, R. ; DREXL, A.: Semi-active, active and non-delay schedules for the resource-constrained project scheduling problem. In: *European Journal of Operations Research* 80 (1995), S. 94–102
- [Stinson et al. 1978] STINSON, J.P. ; DAVIS, E.W. ; KHUMAWALA, B.M.: Multiple resource-constrained scheduling using branch and bound. In: *AIIE Transactions* 10 (1978), S. 252–259
- [Stützle 1995] STÜTZLE, T.: Parallelization strategies for ant colony optimization. In: (Eiben et al., 1995), S. 722–731
- [Stützle 1998] STÜTZLE, T.: An Ant Approach to the Flow Shop Problem. In: *Proceedings of the European Congress on Intelligent Techniques and Soft Computing (EUFIT)*, 1998, S. 1560–1564
- [Stützle 1999] STÜTZLE, T.: *Local Search Algorithms for Combinatorial Problems – Analysis, Improvements, and New Applications*, Technische Universität Darmstadt, Fachbereich Informatik, Dissertation, 1999
- [Stützle und Dorigo 2001] STÜTZLE, T. ; DORIGO, M.: An experimental study of the simple ant colony optimization algorithm. In: MASTORAKIS, N. (Herausgeber): *Proceedings of the 2001 WSES International Conference on Evolutionary Computation (EC'01)*, WSES-Press International, 2001, S. 253–258
- [Stützle und Dorigo 2002a] STÜTZLE, T. ; DORIGO, M.: The ant colony optimization metaheuristic: algorithms, applications, and advances. In: (Glover und Kochenberger, 2002)

- [Stützle und Dorigo 2002b] STÜTZLE, T. ; DORIGO, M.: A short convergence proof for a class of ACO algorithms. In: *IEEE Transaction on Evolutionary Computation* 6 (2002), Nr. 4, S. 358–365
- [Stützle und Hoos 1997] STÜTZLE, T. ; HOOS, H. H.: The MAX-MIN ant system and local search for the traveling salesman problem. In: (Bäck et al., 1997), S. 309–314
- [Stützle und Dorigo 2000] STÜTZLE, T. ; DORIGO, M.: A short convergence proof for a class of ACO algorithms / IRIDIA, Université Libre de Bruxelles, Belgien. 2000 (IRIDIA/2000-35). – Forschungsbericht
- [Stützle und Hoos 2000] STÜTZLE, T. ; HOOS, H.H.: MAX-MIN Ant System. In: *Future Generation Computer Systems* 16 (2000), S. 889–914
- [Talbi et al. 1999] TALBI, E.-G. ; ROUX, O. ; FONLUPT, C. ; ROBILLARD, D.: Parallel ant colonies for combinatorial optimization problems. In: J. ROLIM et al. (Herausgeber): *Proceedings of the Second Workshop on Biologically Inspired Solutions to Parallel Processing Problems (BioSP3)*, 1999 (Lecture Notes in Computer Science (LNCS) 1800), S. 239–247
- [Teich et al. 2001] TEICH, T. ; FISCHER, M. ; VOGEL, A. ; FISCHER, J.: A new Ant Colony Algorithm for the Job Shop Scheduling Problem. In: (Spector et al., 2001), S. 803
- [Valls et al. 2000] VALLS, V. ; QUINTANILLA, S. ; BALLESTIN, F.: *Resource-constrained project scheduling: a critical activity reordering heuristic*. 2000. – Manuskript
- [Veldhuizen und Lamont 2000] VELDHUIZEN, David A. V. ; LAMONT, Gary B.: Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. In: *Evolutionary Computation* 8 (2000), Nr. 2, S. 125–147
- [Voigt et al. 1996] VOIGT, W. ; EBELING, I. ; RECHENBERG, I. ; SCHWEFEL, H.-P. (Herausgeber): *Proceedings of Parallel Problem Solving from Nature IV (PPSN-IV)*. Berlin : Springer, 1996. (Lecture Notes in Computer Science (LNCS) 1141)
- [Vose und Liepins 1991] VOSE, M. D. ; LIEPINS, G. E.: Punctuated equilibria in genetic search. In: *Complex Systems* 5 (1991), S. 31–44
- [Vose 1999] VOSE, M.D.: *The Simple Genetic Algorithm: Foundations and Theory*. Cambridge, MA : MIT Press, 1999

- [Watkins 1989] WATKINS, C. J. C. H.: *Learning from delayed rewards*, King's College, Oxford, Dissertation, Mai 1989
- [Weglarz 1999] WEGLARZ, J. (Herausgeber): *Project Scheduling: Recent Models, Algorithms and Applications*. Boston : Kluwer Academic Publishers, 1999
- [Whitley et al. 2000] WHITLEY, D. ; GOLDBERG, D. ; CANTU-PAZ, E. ; SPECTOR, L. ; PARMEE, I. ; BEYER, H.-G. (Herausgeber): *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2000)*. San Francisco, CA, USA : Morgan Kaufmann, 2000
- [Wohlgemuth et al. 2001] WOHLGEMUTH, S. ; RONACHER, B. ; WEHNER, R.: Ant odometry in the third dimension. In: *Nature* 411 (2001), S. 795–798
- [Wu et al. 2000] WU, C.C. ; LEE, W.C. ; YOU, J.-M.: Trade-off solutions in a single-machine scheduling problem for total earliness and maximum tardiness. In: *International Journal of Systems Science* 31 (2000), S. 639–647
- [Zitzler et al. 2001] ZITZLER, E. ; DEB, K. ; THIELE, L. ; COELLO COELLO, C.A. ; CORNE, D. (Herausgeber): *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO'01)*. Berlin : Springer, 2001. (Lecture Notes in Computer Science (LNCS) 1993)

Index

- ACO-Algorithmus 4
- ACO-Modell 129
 - dynamisches Verhalten 160, 179
 - für Permutationsprobleme
 - eingeschränkte 137, 138
 - heterogene 138, 176
 - homogene 138
 - Fixpunktanalyse 154
 - Maskierung 184
 - Skalierung 184
 - Vergleich mit ACO-Algorithmus 170, 181
- ACS *siehe* Ant Colony System
- Aktivität (RCPSp) 78
- Aktualisierung *siehe*
 - Pheromonaktualisierung
- Ant Colony System 9
- Ant System 8
- Ant-Q 9
- AS *siehe* Ant System
- Auswahlwahrscheinlichkeiten 5, 19
 - ACO-Modell 131
 - Ameisenalgorithmus 5
 - Verzerrung 19, 20, 22, 28, 41, 55, 129, 142, 143, 154, 157, 159, 163, 164, 168, 170
- autokatalytischer Prozess 3
- Bikriterielle Optimierung 111, 116
- Dominanz 112
 - schwache 112
 - starke 113
 - von Zielfunktionswerten 113
 - von Permutationen 113
- Doppelbrückenexperiment 2
- Duftstoff *siehe* Pheromon
- Eigenwerte 156
- einfache Probleme 15
 - Abweichung (P_A) 16, 59
 - Definition 16
 - Diagonale (P_D) 31
 - mehrere Fälligkeitstermine (P_{MF}) 43, 59
 - modifizierte Abweichung (P_{MA}) .. 61
 - Parallele (P_{P_k}) 64
- eingeschränktes Permutationsproblem 137, 138
- elementares Teilproblem 138
- Elitestrategie 9, 87, 105
- Entropie 95
- Entscheidungsreihenfolge 53
 - Rückwärts-Ameisen 56, 90, 103
 - Vorwärts-Ameisen 56, 90, 103
 - zufällige 54
- eusoziale Insekten 1
- Fixpunktanalyse 154
- Fixpunktvektoren 155
 - Darstellung 157
- Front
 - nicht-dominierte 113
 - globale 118
 - lokale 118
 - Pareto-optimale 113
- Generation 5
- global beste Lösung 9
- globale Pheromonauswertung *siehe*
 - Pheromonauswertung

- heterogenes Permutationsproblem ... 138
- Heuristikeinfluss 6
 Veränderung zur Laufzeit ... 88, 104
- homogenes Permutationsproblem ... 138
- indirekte Kommunikation 2
- Insekten
 eusoziale 1
 indirekte Kommunikation 2
 soziale 1
 Staaten bildende 1
- Iteration 5
- Jacobimatrix 156
- Kodierung der Pheromonmatrix .. 6, 111,
 115, 126
- Kolonie 4, 10, 90, 111, 118
- Konkurrenz 139
- Kostenfunktion 5, 16
- Kostenmatrix 16
- Lösungspermutation 5
- lineares Zuordnungsproblem 16
- lokale Pheromonauswertung *siehe*
 Pheromonauswertung
- Malus 139
- Malussituation 139
 Wahrscheinlichkeit einer 140
- Markovkette 143
- Maskierung 184
- MAX-MIN Ant System 10
- Medianfront 113
- Metaheuristik 4
- Modellierung *siehe* ACO-Modell
- Multikriterielle Optimierung 111
- nicht-dominierte Front 113
- paralleles Prioritätsregelverfahren 81
- Pareto-Optimalität 113
- Permutationsproblem 4
 eingeschränktes 138
- heterogenes 138
- homogenes 138
- Phasenstruktur 91
- Pheromon 2, 6
- Pheromonaktualisierung 6
 angepasste 40
 globale 9
 lokale 9
- Pheromonauswertung 25
 gewichtete Summen- 27, 87
 globale 25
 lokale 25
 relative 38
 Summen- .. 26, 86, 94, 117, 159, 165,
 166
- Pheromoneinfluss 6
- Pheromonmatrix 6
 durchschnittliche 18
- Pheromonvektoren 154
 Darstellung 157
- Prioritätsregeln 84
- Prioritätsregelverfahren 79
- Projektplan 78
- Projektplanung *siehe* RCPSP
- Pseudo-random Proportional Rule . 9, 27
- PSPLIB 92
- RCPSP 78
 Aktivität 78
 Aktivitätenliste 78
 Ameisenalgorithmus für 82
 lokale Optimierung 89
 Präzedenzgraph 79
 Prioritätsregeln 84
 Prioritätsregelverfahren 79
 paralleles 81
 serielles 80
 Projektplan 78
 zulässiger 79
- Ressourcen 78
- Ressourcenkapazität 78
- Vorrangbeziehung 78

relative Pheromonauswertung.....	<i>siehe</i>
Pheromonauswertung	
Ressourcen	78
Schwarmintelligenz	1
serielles Prioritätsregelverfahren	80
Skalierung.....	184
SMTEMDP	47
SMTTP	34
mit Umrüstkosten	115
SMTWDP.....	68
SMTWTP.....	33
soziale Insekten	1
Staaten bildende Insekten	1
stigmergy.....	2
Summen-Pheromonauswertung	<i>siehe</i>
Pheromonauswertung	
Theorie.....	11, 129
TSP.....	6
Umrüstkosten	115
unendliche Population	129
Ungarische Methode.....	16
Verdunstungsrate.....	6
Veränderung zur Laufzeit ...	88, 104
Verzerrung	<i>siehe</i>
Auswahlwahrscheinlichkeiten	
Vorrangbeziehung	54

