

Andreas Cardeneo

Modellierung und Optimierung des B2C-Tourenplanungsproblems mit alternativen Lieferorten und -zeiten



Andreas Cardeneo

Modellierung und Optimierung des B2C-Tourenplanungsproblems mit alternativen Lieferorten und -zeiten

Wissenschaftliche Berichte des
Institutes für Fördertechnik und Logistiksysteme
der Universität Karlsruhe (TH)
Band 66

Modellierung und Optimierung des B2C-Tourenplanungsproblems mit alternativen Lieferorten und -zeiten

von
Andreas Cardeneo



universitätsverlag karlsruhe

Dissertation, Universität Karlsruhe (TH)

Fakultät für Maschinenbau, 2005

Referenten: Prof. Dr.-Ing. Dr. h.c. Dieter Arnold, Prof. Dr.-Ing. Kai Furmans

Impressum

Universitätsverlag Karlsruhe
c/o Universitätsbibliothek
Straße am Forum 2
D-76131 Karlsruhe
www.uvka.de



Dieses Werk ist unter folgender Creative Commons-Lizenz
lizenziert: <http://creativecommons.org/licenses/by-nc-nd/2.0/de/>

Universitätsverlag Karlsruhe 2005
Print on Demand

ISSN 0171-2772

ISBN 3-937300-75-9

**Modellierung und Optimierung
des B2C-Tourenplanungsproblems mit
alternativen Lieferorten und -zeiten**

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der Fakultät für Maschinenbau
der Universität Karlsruhe (TH)
genehmigte

Dissertation

von

Dipl.-Wi.-Ing. Andreas Cardeneo

aus Voorburg (Niederlande)

Tag der mündlichen Prüfung:
Hauptreferent:
Korreferent:

17. Juni 2005
Prof. Dr.-Ing. Dr. h.c. D. Arnold
Prof. Dr.-Ing. K. Furmans

Vorwort

*Die Technik von heute ist das Brot von morgen –
die Wissenschaft von heute ist die Technik von morgen.*
Richard von Weizsäcker

Idealerweise gelingt es in der angewandten Forschung, Forschungsinhalte aus praktischen Problemen abzuleiten und durch wissenschaftlichen Fortschritt zu neuen innovativen und umsetzbaren Lösungen zu gelangen. Dieses Ideal zu erreichen ist der Anspruch des Instituts für Fördertechnik und Logistiksysteme der Universität Karlsruhe (TH), an dem die vorliegende Dissertationsschrift entstand.

Ich bin sehr dankbar dafür, eine von Freiheit einerseits und damit einhergehender Verantwortung andererseits geprägte Arbeitsatmosphäre erlebt haben zu dürfen, die es mir erlaubt hat, eine Arbeit anzufertigen, von der ich hoffe, daß sie dem oben formulierten Ideal nahekkommt.

Für dieses von Vertrauen geprägte Klima zeichnet in erster Linie mein akademischer Lehrer Prof. Dr.-Ing. Dr. h.c. Dieter Arnold verantwortlich, bei dem ich mich ganz herzlich für die Übernahme des Referats bedanken möchte. Mein Dank gilt weiterhin Prof. Dr.-Ing. Kai Furmans für die Übernahme des Korreferats sowie Prof. Dr.-Ing. Jürgen Fleischer für die Übernahme des Vorsitzes der mündlichen Prüfung.

Natürlich ist eine Dissertation in erster Linie eine wissenschaftliche Leistung des jeweiligen Verfassers. Dennoch ist sie auch das Werk einer Gemeinschaft, bestehend aus der Familie, Freunden und Kollegen. Auch ich möchte mich an dieser Stelle bei denjenigen bedanken, die mich – jeweils auf ihre Weise – während der verschiedenen Phasen dieser Arbeit unterstützt haben.

An erster Stelle möchte ich mich bei meiner Familie und besonders meiner Lebensgefährtin Beate für den Rückhalt und das fortwährend in mich gesetzte Vertrauen bedanken. Ich bin zuversichtlich, daß das Ergebnis für die zahlreichen Abende und Wochenenden, an denen ich abwesend war, entschädigt. Auch meine Kinder Patrick und Sanja haben mich immer unterstützt, indem sie für ausreichend Ablenkung gesorgt und mir damit mühelos über das eine oder andere „lokale Optimum“ hinweggeholfen haben.

Mein besonderer Dank gilt meinen Eltern, die stets am Fortschritt der Arbeit interessiert waren und an mich geglaubt haben. Neben meinen Eltern haben auch Beates

Eltern Hans und Angela mich tatkräftig unterstützt, in dem sie während einiger Aufenthalte die Ablenkung und Versorgung der Kinder dann übernommen haben, wenn der Fortschrittsgradient nach oben zeigte und ich den Arbeitsfluß erhalten wollte.

Während der Elternzeit für Patrick und später für Sanja habe ich viele liebe Menschen kennengelernt, deren ganz gewöhnliche Probleme bei der Kindererziehung und den Dingen des täglichen Lebens immer für die eigene Erdung gesorgt haben und verhindert, daß ich mich in wissenschaftlichen Sphären verliere. Besonders möchte ich mich an dieser Stelle bei Anne bedanken, die es immer wieder geschafft hat, auch noch zu später Stunde Rohfassungen dieser Arbeit akribisch Korrektur zu lesen und mir wertvolle Hinweise zu geben.

Zu guter Letzt bedanke ich mich bei meinen Kollegen am IFL für die stets vorhandene Diskussionsbereitschaft, die Aufgeschlossenheit für meine Anliegen und die wertvolle konstruktive Kritik.

Karlsruhe, im Juli 2005

Andreas Cardeneo

Kurzfassung

Andreas Cardeneo

Modellierung und Optimierung des B2C-Tourenplanungsproblems mit alternativen Lieferorten und -zeiten

In dieser Arbeit wird ein Optimierungsverfahren für das Tourenplanungsproblem mit alternativen Lieferorten und -zeiten entwickelt und bewertet. Ausgangspunkt der Entwicklung ist eine Untersuchung der logistischen Herausforderungen der B2C-Logistik, der Zustellung von Gütern des täglichen Bedarfs an private Endkunden. Zustelldienste stehen hier oftmals vor dem Problem erfolgloser Zustellversuche infolge abwesender Empfänger.

Den kostenorientierten existierenden Konzepten, bei denen durch den Dienstleister größere Sendungsmengen an einem Ort zur späteren Abholung durch die einzelnen Empfänger hinterlegt werden, wird ein serviceorientierter Ansatz zur Seite gestellt. Grundgedanke des Ansatzes ist das Angebot alternativer Lieferorte und -zeiten durch die Empfänger und die Auswahl daraus durch den Dienstleister. Eine Umsetzung dieses Ansatzes in Form eines webbasierten Informationssystems wird gezeigt.

Für den Paketdienstleister stellt sich in der Folge ein Tourenplanungsproblem mit alternativen Lieferorten und -zeiten. Dieses Problem wird als gemischt-ganzzahliges lineares Programm modelliert und ein Lösungsverfahren wird entwickelt. Hierbei handelt es sich um einen *Branch-and-Price*-Ansatz, der auf einer Zerlegung des Planungsproblems in einen Teil zur optimalen Tourenausswahl und einen Teil zur Routenbestimmung basiert. Für die Tourenausswahl wird Lineare Programmierung verwendet, während die Routenbestimmung mittels Constraint Programming modelliert und durchgeführt wird. Weiterhin kommen heuristische Lösungsverfahren und verschiedene algorithmische Techniken zur Suchraumreduktion zum Einsatz.

Die Bewertung des Verfahrens betrachtet den Einfluß verschiedener algorithmischer Varianten und struktureller Faktoren auf die Laufzeit. Aufbauend auf modifizierten Testinstanzen aus der Literatur werden Empfehlungen für die Gestaltung und Beschleu-

nigung der Routenbestimmung gegeben. Es kann u.a. gezeigt werden, daß die entwickelten Suchschemata sich im Vergleich zu bekannten Ansätzen bewähren.

Für die Untersuchung des logistischen Potentials werden realitätsnahe künstliche Szenarien erzeugt, mit deren Hilfe das Kostensenkungspotential alternativer Lieferorte aus Dienstleistersicht bestimmt wird. Es kann empirisch gezeigt werden, daß die Einführung von alternativen Lieferorten und -zeiten zu deutlichen Kostensenkungen und kleineren Flottengrößen führt. Um Planungsinstanzen im Hinblick auf das Kostensenkungspotential schnell miteinander vergleichen zu können, wird ein Verbesserungsindex vorgestellt.

Gemeinsam mit der Ergebnisvorstellung werden Empfehlungen für zukünftige Forschungsarbeiten gegeben.

Abstract

Andreas Cardeneo

Modelling and optimization of the B2C vehicle routing problem with alternative delivery locations and time windows

Scope of this doctoral thesis is the development and evaluation of an optimization method for the vehicle routing problem with alternative delivery locations and time windows. Based on an assessment of B2C logistics operations, the absence of customers at the time of delivery is identified as a major challenge to express and parcel delivery services.

Current solution concepts rely on depositing shipments at locations wherefrom they are picked up by customers. This thesis proposes an additional service-oriented approach based on the mutual agreement between customers and service providers on delivery locations and time windows. Customers may offer alternative locations and time windows among which service providers make a selection. A web-based information system implementing this concept is presented.

The resulting vehicle routing problem with alternative delivery locations and time windows is modelled as a mixed-integer program (MIP). A branch-and-price approach is proposed, based on a decomposition scheme. While linear programming is used for determining optimal route selection, constraint programming is the method of choice for modelling and calculating optimal vehicle routes with side constraints. In addition, heuristic solution methods and various algorithmic techniques for search space reduction are used.

Computational results based on modified benchmark instances taken from literature are presented. In particular, the influence of particular algorithm elements and problem structure on runtime performance are analyzed. Recommendations on the configuration of the full optimization algorithm are given. Among others, it can be shown that the newly developed search schemes perform favourably compared to known approaches. The potential for cost reduction from a service providers perspective is analyzed using a set of artificial but realistically structured set of scenarios. The empirical results show

that a considerable reduction in cost and smaller fleet sizes can be achieved. A method for rapid comparison of different instances with respect to the achievable cost reduction is presented and future research directions are suggested.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ziel und wissenschaftlicher Beitrag	2
1.2	Aufbau der Arbeit	3
2	B2C-Logistik	5
2.1	Charakterisierung der B2C-Logistik	6
2.1.1	Akteure	6
2.1.2	Artikelspektrum der B2C-Logistik	6
2.1.3	B2C-Logistik und eCommerce	7
2.1.4	B2C-Logistik und Verkehr	9
2.2	Herausforderungen der B2C-Logistik	10
2.2.1	Kommissionierung	11
2.2.2	Transport	12
2.2.3	Logistik als Wettbewerbsinstrument	13
2.2.4	Die Problematik der Feinverteilung	14
2.3	Lösungskonzepte für die Letzte Meile	16
2.3.1	Entkopplungskonzepte	17
2.3.2	Abstimmungskonzepte	23
2.4	Personal Logistics Assistant – ein Integrationsansatz	25
2.5	Tourenplanungsproblem mit alternativen Liefermöglichkeiten	28
3	Methodische Grundlagen	31
3.1	Grundlegende Konzepte	31
3.2	Branch-and-Price und Column Generation	33
3.2.1	Prinzipskizze des Verfahrens	34
3.2.2	Die Dantzig-Wolfe-Dekomposition	36
3.2.3	Column Generation	38
3.2.4	Column Generation und Modellierung	40
3.2.5	Lösung des Pricing-Problems	40
3.2.6	Branch-and-Price – Berechnung ganzzahliger Lösungen	41
3.2.7	Branch-and-Price-and-Cut	44
3.2.8	Herstellung einer Initiallösung	45
3.3	Constraint Programming	46

3.3.1	Grundlegende Konzepte: Variablen, Domains, Constraints . .	47
3.3.2	Lösung von Constraint Programmen	49
3.3.3	Variableninstanziierung	51
3.3.4	Lösung von Optimierungsproblemen	53
3.3.5	Gestaltung von CP-Lösungsverfahren	55
3.4	Hybride Verfahren: Constraint Programming-basiertes Column Generation	56
3.5	Heuristiken und Metaheuristiken	57
3.5.1	Eröffnungsverfahren	58
3.5.2	Lokale Suche	60
3.5.3	Metaheuristiken	63
4	Aktuelle Arbeiten zu verwandten Problemen – ein Literaturüberblick	69
4.1	Anwendung von Column Generation auf Tourenplanungsprobleme . .	70
4.2	Anwendung des Column Generation auf weitere logistische Problemstellungen	72
4.3	Weitere MIP-Ansätze zur Lösung des VRPTW	73
4.4	Lösung des Handlungsreisendenproblems mittels Constraint Programming	74
4.5	Hybride MIP/CP Ansätze	79
4.5.1	Vergleichende Arbeiten	79
4.5.2	Integrierte Modellbildung	80
4.5.3	Kombination von OR und CP	82
4.6	Heuristische Verfahren	86
4.6.1	Kombination von Lokaler Suche und Constraint Programming	87
4.7	Bilanz des Literaturüberblicks	88
5	Modellierung des Tourenplanungsproblems mit alternativen Liefermöglichkeiten	91
5.1	Das Standardtoursplanungsproblem mit Kapazitäts- und Zeitfensterrestriktionen	92
5.1.1	Entscheidungsvariablen	93
5.1.2	Strukturbedingungen	93
5.1.3	Fahrzeugrestriktionen	94
5.1.4	Zeitfensterrestriktionen	95
5.1.5	MIP Formulierung des VRPTW	96
5.2	Das VRPTW als allgemeines Routingproblem	97
5.3	Erweiterung um alternative Liefermöglichkeiten	98
5.3.1	MIP Formulierung des Tourenplanungsproblems mit alternativen Liefermöglichkeiten	99
5.4	Das Dekompositionsmodell	100
5.4.1	Das Master-Problem	101
5.4.2	Das Pricing-Problem – Ein Constraint Programming-Modell .	103

5.5	Weitere Anwendungsfälle des Modells	107
5.5.1	Mehrfachzeitfenster	108
5.5.2	Einfache intermodale Transporte: Abbildung von Fahrplänen für Shuttle-Züge	108
5.5.3	Tageszeitabhängige Fahrtkosten	109
6	Lösungsverfahren für das VRPA	111
6.1	Master-Problem	112
6.1.1	Set Partitioning- und Set Covering-Formulierungen	112
6.1.2	Berechnung einer Initiallösung	114
6.2	Pricing-Problem	115
6.2.1	Strukturelle Constraints	115
6.2.2	Behandlung von Zeitfenster- und Kapazitätsrestriktionen	117
6.2.3	Optimierungsconstraints	118
6.2.4	Zielfunktion	125
6.2.5	Variableninstanziierung	126
6.3	Herstellung ganzzahliger Lösungen: Branching-Schema	127
6.4	Behandlung von Spalten	130
6.5	Verfahrensvarianten	131
6.6	Implementierungsaspekte	131
7	Algorithmische und logistische Bewertung	133
7.1	Algorithmische Bewertung	134
7.1.1	Testumfeld	134
7.1.2	Experimentelle Untersuchung	141
7.1.3	Lösungsgüte in der Initialphase	141
7.1.4	Einfluß der Initialphase auf die Gesamtlaufzeit	145
7.1.5	Wirkung der Bounding und Pruning-Techniken	147
7.1.6	Zielkriterium des Pricing-Problems	149
7.1.7	Beitrag des Spaltenmanagements	151
7.1.8	Vergleich von Instanziierungsstrategien	154
7.1.9	Analyse des Branching-Verhaltens	155
7.1.10	Zusammenhang zwischen Problemstruktur und Laufzeit	157
7.1.11	Schlußfolgerungen aus der algorithmischen Bewertung	158
7.2	Untersuchung des logistischen Potentials alternativer Liefermöglichkeiten	160
7.2.1	Anwendungsszenario	160
7.2.2	Erzeugung der Szenarien	166
7.2.3	Szenarienübersicht	168
7.2.4	Wegersparnis durch alternative Liefermöglichkeiten	169
7.2.5	Reduktion der Fahrzeugzahl	171
7.2.6	Strukturelle Ursachen der Verbesserung	172
7.2.7	Kombinierbarkeit von Liefermöglichkeiten	174

7.2.8 Bilanz und Ausblick der logistischen Bewertung	176
8 Zusammenfassung	179
Literaturverzeichnis	182

1 Einleitung

*Ratlosigkeit und Unzufriedenheit
sind die ersten Vorbedingungen des Fortschritts*
Thomas Alva Edison

Bei seiner Geburt ist ein Mensch im Vergleich zu anderen Säugetieren erstaunlich unreif. Es ist ihm – im Gegensatz zu vielen Tieren – unmöglich, allein zu stehen oder zu gehen. Erst Zug um Zug lernt der Mensch zu kommunizieren, sich fortzubewegen und sich sozial zu verhalten. Jeder Lernschritt, jede neue Fähigkeit erweitert den Aktionsraum und bietet neue Möglichkeiten. Lernprozesse haben im Laufe der Evolution zur Erschaffung und zum Gebrauch von Werkzeugen geführt, getrieben von Bedürfnissen, die sich ihrerseits auch entsprechend der Möglichkeiten und des Umfeldes veränderten und auch heute noch verändern. Ökonomisches Verhalten scheint dabei natürlich zu sein, und logistische Verhaltensmuster entwickeln sich evolutionär: Werden beim Spiel mit Bausteinen diese zunächst einzeln von Raum zu Raum getragen, nimmt ein Kind nach kurzer Zeit einen Behälter zu Hilfe, um diese Aufgabe effizienter durchzuführen – wie der Autor dieser Arbeit bei seinem damals knapp zweijährigen Sohn beobachten konnte.

Das Bestreben nach effizienter Logistik ist folglich dem Anschein nach ein natürliches Verhalten. Wie können jedoch Logistikleistungen jenseits der kindlichen Intuition effizient erbracht werden? Nach gängiger Definition ist das Verhältnis aus Ertrag und dem dafür erbrachten Aufwand ein Maß für die Effizienz. Eine Steigerung der Effizienz ist entsprechend des Ökonomischen Prinzips entweder durch eine Maximierung des Ertrags bei gleichbleibendem Aufwand oder durch eine Minimierung des Aufwands bei gleichbleibendem Ertrag erreichbar. Die Entscheidung darüber, welcher dieser Wege eingeschlagen wird, ist abhängig vom Zielsystem des Handelnden und unterliegt zugleich einem System von Restriktionen. Es liegt demnach ein Optimierungsproblem vor.

In der Praxis stellt sich die Frage, welcher Grad der Effizienz anzustreben ist. Im wirtschaftlichen Wettbewerb wird die Antwort darauf von der Konkurrenzsituation abhängen und so ausfallen, daß die Wettbewerbsfähigkeit gesichert ist. Das theoretische Ideal der maximalen Effizienz wird in der Praxis dann durch ein erreichbares und als ausreichend angesehenes Effizienzniveau ersetzt. Die Dynamik des Wettbewerbs wird dieses anvisierte Effizienzniveau jedoch kontinuierlich verschieben, so daß sich für die

unternehmerische Entscheidungsfindung die naheliegende Frage nach dem bestmöglichen Effizienzniveau – dem Optimum – ergibt. Der besondere Wert des Optimums liegt dann in seiner Kenntnis und in seiner Verwendung als Maßstab: Die eigene Position kann eingeordnet, eigene Handlungen können zielgerichtet gelenkt werden.

Eine festgestellte Abweichung zwischen aktuellem Zustand und Optimum ist jedoch nur von strategischem Wert und muß in konkrete ausführbare Handlungen umgesetzt werden. Auf der operativen Ebene werden Rahmenbedingungen zumeist als nicht kurzfristig veränderbar angesehen, so daß nach Optimallösungen innerhalb der scheinbar gegebenen Grenzen gesucht wird. Bei diesen Grenzen handelt es sich jedoch oftmals um gewachsene Strukturen, die nur deshalb existieren, weil zum Zeitpunkt ihrer Entstehung andere Alternativen unbekannt oder mangels Wissen, geeigneter Methoden oder Werkzeuge nicht realisierbar waren.

An diesem Punkt kann anwendungsorientierte Forschung ansetzen und über die wissenschaftliche Erarbeitung von Methoden den Grundstein für praktisch einsetzbare Werkzeuge legen. Neue Werkzeuge können es ermöglichen, gegebene Strukturen zu überwinden und ein größeres Optimierungspotential zu erschließen. Unternehmen werden durch den Gebrauch innovativer Werkzeuge in die Lage versetzt, verbesserte Leistungen am Markt anzubieten oder kostengünstiger zu agieren. Im Idealfall kann beides erreicht werden.

Dieser Argumentation folgend, legt die vorliegende Arbeit die methodischen und konzeptionellen Grundlagen für ein Werkzeug — dem *Personal Logistics Assistant* (PLA) – zur Optimierung der *Business-to-Consumer*-Logistik (B2C-Logistik), der Logistik zur Zustellung von Gütern des täglichen Bedarfs an private Endkunden.

Bei dem PLA handelt es sich um eine informationstechnische Plattform, mit deren Hilfe Transportdienstleister und Kunden die Zustellung oder Abholung von Waren kooperativ organisieren können. Der Einsatz eines solchen Werkzeugs stellt neue Anforderungen an die Planungssysteme der Dienstleister und ist nur möglich, wenn eine dafür geeignete Planungsmethode verfügbar ist. Eine solche Methode wird in dieser Arbeit in Form eines exakten Lösungsverfahrens für das Tourenplanungsproblem mit alternativen Lieferorten und -zeiten – kurz als alternative Liefermöglichkeiten bezeichnet – entwickelt.

1.1 Ziel und wissenschaftlicher Beitrag

Der PLA erlaubt es Transportdienstleistern, durch Kooperation mit den Kunden die eigenen Kosten bei gleichzeitiger Erhöhung der Servicequalität zu senken. Die vorliegende Arbeit erweitert die Literatur zur B2C-Logistik um diese kooperativen Distributionskonzepte, die bislang nicht betrachtet worden sind.

Das Tourenplanungsproblem mit alternativen Liefermöglichkeiten ist grundlegend für das Konzept und das Werkzeug. Nach Kenntnis des Autors wird die Problemstellung

hier zum ersten Mal wissenschaftlich untersucht. Das Problem wird definiert, modelliert und ein exaktes Lösungsverfahren wird konzipiert, implementiert und bewertet.

Das entwickelte Verfahren zeichnet sich durch die Integration von gemischt-ganzzahliger Programmierung, Constraint Programming, Heuristiken und Graphalgorithmen aus und erweitert bestehende Ansätze um neue methodische Aspekte, die an geeigneter Stelle ausgeführt werden.

Die Bewertung umfasst neben algorithmischen Kriterien auch die Untersuchung des Konzepts alternativer Liefermöglichkeiten anhand von Distributionsszenarien. Die Erzeugung der Szenarien erfolgt dabei durch einen innovativen Ansatz, bei dem Constraint Programming für die Deklaration der Szenarienspezifikation und zugleich für die Instanzenerzeugung verwendet wird.

1.2 Aufbau der Arbeit

Die Arbeit führt zunächst in die Thematik der Business-to-Consumer-Logistik ein. Es wird begründet, weshalb diesem Zweig der Logistik besondere Aufmerksamkeit gewidmet werden sollte, indem die charakteristischen Aspekte dargestellt und die spezifischen Anforderungen daraus abgeleitet werden.

Die methodischen Grundlagen des entwickelten Optimierungsverfahrens werden in Kapitel 3 erarbeitet. Nach derzeitigem Erkenntnisstand ist bei Vorliegen einer geeigneten Struktur *Branch-and-Price* die Methode der Wahl zur Lösung großer gemischt-ganzzahliger linearer Programme. *Constraint Programming* wiederum bietet sich an zur Modellierung komplexer Sachverhalte. Tatsächlich verspricht Constraint Programming hier einen Vorteil nicht allein zum Zeitpunkt der erstmaligen Erstellung des Modells, sondern während dessen gesamtem Lebenszyklus: Die Anpassung des Modells an neue Gegebenheiten ohne die Notwendigkeit, zugleich den Lösungsalgorithmus zu modifizieren, erhöht die in der Praxis wichtige Flexibilität. Diese Möglichkeit kann zudem zu einer weiteren Verbreitung von Optimierungswerkzeugen beitragen, weil anwendungsnahe Modellierer nun nicht mehr zugleich auch Algorithmenentwickler sein müssen.

Kapitel 4 gibt einen umfassenden Literaturüberblick über relevante und thematisch verbundene wissenschaftliche Arbeiten. Zunächst jeweils für sich, werden gemischt-ganzzahlige und Constraint Programming-basierte Verfahrensansätze zur Optimierung von Routingproblemen vorgestellt. Anschließend konzentriert sich die Betrachtung auf hybride Ansätze, in denen beide Methoden auf verschiedene Art und Weise miteinander kombiniert werden. Mit der Bilanz des Literaturüberblicks wird die Auswahl des Optimierungsverfahrens, wie sie in dieser Arbeit getroffen wird, begründet.

Grundlage des Optimierungsverfahrens ist ein eindeutig formuliertes mathematisches Modell. Kompakte MIP-Modelle und extensive Dekompositionsmodelle für das Tourenplanungsproblem mit alternativen Liefermöglichkeiten werden in Kapitel 5 vorge-

stellt. Dabei zeigt sich, daß aus Effizienzgründen Gesamtmodell und Lösungsverfahren aufeinander abgestimmt werden müssen. Die Modelle für die Teilprobleme des Dekompositionsansatzes werden erstellt und die problemspezifischen Erweiterungen um Alternativen formuliert.

Die Mächtigkeit mathematischer Modelle zeigt sich im Transfer der gleichen Modellstruktur auf intuitiv verschieden erscheinende Problemstellungen. Für einige ausgewählte Beispiele aus der Logistik wird dieser Transfer geleistet.

Die Entwicklung eines Optimierungsverfahrens auf Basis der Branch-and-Price-Methode bietet im Detail zahlreiche Freiheitsgrade für die Gestaltung. Auf die in dieser Arbeit entstandenen Varianten des Verfahrens wird in Kapitel 6 eingegangen. Die Verwendung des Dekompositionsansatzes setzt sich in der Architektur fort. Der komponentenorientierte Aufbau erlaubt es, einzelne Bestandteile zu modifizieren, ohne das Zusammenspiel zu beeinträchtigen. Von dieser Möglichkeit wird in Kapitel 7 Gebrauch gemacht, indem verschiedene Kombinationen von Verfahrensbestandteilen einer Leistungsanalyse unterzogen werden. Der Vergleich von Verfahrensvarianten erfolgt anhand modifizierter Instanzen von Standard-Benchmarkproblemen. Die Bewertung des logistischen Konzepts alternativer Liefermöglichkeiten bedient sich einer Modellstadt, deren Population und Struktur von einem auf Constraint Programming basierendem Szenariengenerator erzeugt wird. Ebenfalls in diesem Kapitel finden sich Ausblicke auf mögliche weitere wissenschaftliche Fragestellungen.

Zum Abschluß der Arbeit resümiert die Zusammenfassung in Kapitel 8 nochmals Problemstellung, Lösungsansatz und Ergebnisse.

2 B2C-Logistik

*[...] daß alles Unglück der Menschen einem entstammt,
nämlich daß sie unfähig sind, in Ruhe
allein in ihrem Zimmer bleiben zu können*
Blaise Pascal

Das britische Wirtschaftsmagazin *The Economist* beschrieb 2000 in einem Artikel mit dem Titel *Distribution dilemmas*, wie amerikanische Unternehmen im Weihnachtsgeschäft 1999 feststellen mußten, daß Distributionssysteme, die dazu gestaltet waren, Filialen mit palettierter Ware zu beliefern, nicht ohne Weiteres auch Sendungen für den Endkunden in gleicher Qualität abwickeln konnten. Für die Unternehmen zeigte sich, daß trotz aufwendig gestalteter Webseiten und Shopsysteme letztlich die funktionierende Sendungsauslieferung über den Geschäftserfolg entscheidet.

Die aufgetretenen Schwierigkeiten erwiesen deutlich, daß die *Business to Consumer-Logistik* (B2C-Logistik), d.h. die Zustellung von Warensendungen an private Endkunden, die Logistiksysteme vor besondere Herausforderungen stellt und darauf abgestimmte Maßnahmen erfordert. Die Unternehmen können dabei allerdings nicht isoliert von der Außenwelt agieren und lediglich interne Prozesse verbessern – vielmehr können effektive (und nach Möglichkeit auch effiziente) Maßnahmen nur dann erfolgreich wirken, wenn die Gestaltung des Distributionssystems das wettbewerbliche Umfeld umfassend einbezieht. Von besonderem Stellenwert ist dabei die Beachtung der Kundenwünsche. Diese Wünsche werden bislang zumeist als Kostentreiber gesehen, bieten aber bei kooperativer Abstimmung auch die Chance der wettbewerblichen Differenzierung und Leistungsverbesserung.

In diesem Kapitel werden zunächst die B2C-Logistik anhand verschiedener Kriterien charakterisiert und die logistischen Herausforderungen dargestellt. Dabei liegt der Schwerpunkt auf der Transportlogistik und dem noch näher zu definierenden Problem der *Letzten Meile*. Zu diesem Themenbereich werden Lösungskonzepte erläutert und strukturiert. Als neues kooperatives Abstimmungsinstrument für die B2C-Logistik wird das Konzept des *Personal Logistics Assistant* präsentiert. Daraus resultieren neue Anforderungen an Planungssysteme, wobei das Tourenplanungsproblem mit alternativen Liefermöglichkeiten hierbei eine Schlüsselstellung einnimmt. Das Problem wird aus Anwenderperspektive präsentiert und der Bedarf für die Neuentwicklung eines Optimierungsverfahrens wird aufgezeigt.

2.1 Charakterisierung der B2C-Logistik

Unter B2C-Logistik werden die mit der physischen Distribution von Waren des täglichen Bedarfs von Unternehmen an Privathaushalte zusammenhängenden Prozesse der Logistik verstanden. Die B2C-Logistik ist demnach ein Teilgebiet der Distributionslogistik und umfaßt neben organisatorischen und dispositiven Prozessen insbesondere Materialflußprozesse des Transports und des Lagerns.

2.1.1 Akteure

In Bezug auf die Akteure kann die B2C-Logistik von der B2B- und der C2C-Logistik abgegrenzt werden. Während die „Business to Business“-Logistik die Abwicklung der Warenströme zwischen den verschiedenen Stufen der Wertschöpfungskette bestehend aus Lieferanten, Produzenten und Händlern abdeckt, steht C2C für „Consumer to Consumer“ und bezieht sich in logistischer Hinsicht zumeist auf Postdienste. In der B2C-Logistik finden sich auf Unternehmensseite sowohl die Versender als auch zwischen-geschaltete Dienstleister für die Abwicklung von Finanztransaktionen und Logistik. Die Endkunden sind je nach Handelsbeziehung zu einem unterschiedlichen Grad in die B2C-Logistik eingebunden.

Bereits der alltägliche Einkauf in einem Supermarkt ist eine der B2C-Logistik zuzurechnende Tätigkeit, da der Kunde im Laden Artikel entsprechend seiner Einkaufsliste entnimmt und somit zunächst als Kommissionierer tätig ist. Anschließend übernimmt der Kunde ebenfalls den Transport der Ware nach Hause. Während also für Lebensmittel der Kunde die Distributionslogistik übernimmt, tritt bei großvolumigen oder schweren Gütern (bspw. Möbel oder Elektrogroßgeräte) ein anderer Fall ein. Hier sucht sich der Kunde die Ware häufig in einem Ladenlokal aus und veranlaßt die Auslieferung durch den Einzelhändler. Da es sich bei der ausgestellten Ware zumeist um Ansichtsexemplare handelt, löst der Kauf einen Beschaffungs- und einen anschließenden Distributionsprozess aus. Neben dem Transport sind im Rahmen der Auslieferung dann häufig noch zusätzliche Dienstleistungen wie Aufbau oder Inbetriebnahme durchzuführen. Paradebeispiel für B2C-Logistik ist jedoch der Versandhandel: Der Kunde wählt die zu bestellende Ware aus einem in digitaler oder gedruckter Form vorliegenden Katalog aus und kann anschließend über verschiedene Kommunikationskanäle die Bestellung aufgeben. Der Versandhändler wickelt den Auftrag ab, d.h. er kommissioniert die Positionen des Auftrags, konfektioniert und konfiguriert die Artikel gegebenenfalls, verpackt und versendet sie. Den Transport zum Endkunden übernehmen (s. (Daduna 2000)) zumeist spezialisierte Dienstleister.

2.1.2 Artikelspektrum der B2C-Logistik

Neben den Akteuren und ihrem Zusammenspiel im Markt ist die Art und Beschaffenheit der physisch bewegten Güter eine weitere charakterisierende Dimension. Insbe-

sondere prägen Geometrie und Abmessungen der Waren des Sortiments Struktur und Abläufe des Logistiksystems.

Nach (Retail Logistics Task Force 2001) und (Rahn 2004) werden in der B2C-Logistik typischerweise Güter des täglichen Bedarfs transportiert, wobei Rahn unter Abschätzung des üblichen Sortiments im Versandhandel von mindestens 250.000 verschiedenen Artikeln aus den Bereichen Lebensmittel, Bekleidung und Hausrat ausgeht. Hauptsächlich umgesetzt werden nach (TNS 2002) Produkte, die unabhängig vom Händler überall in gleicher Form und Funktionalität erhältlich sind. Dazu gehören Bücher, Musikträger, Markenkleidung, Unterhaltungselektronik und Telekommunikationsgeräte, Computer und Zubehör oder Lebensmittel. Dieses Sortiment weist eine große Vielfalt bezüglich Abmessungen und Gewicht, Wert und Handhabbarkeit auf. Beispielsweise stellen Lebensmittel während Lagerung und Transport gänzlich andere Anforderungen an Temperatur und Hygiene als Kleidung. Diese kann wiederum im Vergleich zu anderen Konsumgütern besonders wertvoll sein. Nicht zum Spektrum der B2C-Logistik werden digitalisierbare Produkte wie Musikstücke, elektronische Bücher, Reisetickets und Konzertkarten gezählt, die über Netzwerke wie dem Internet oder Mobilfunknetze vertrieben werden oder bei denen es sich um reine Buchungsvorgänge handelt. Diese Produkte können beliebig vervielfältigt, über elektronische Netzwerke transportiert und zu nahezu vernachlässigbaren Grenzkosten gelagert werden¹. Ausgeschlossen sind ebenfalls Versorgungsleistungen wie Gas, Wasser, Heizöl oder Strom, die über Rohrleitungen oder Kabel an die Haushalte geliefert werden.

Die Distribution von Nahrungsmitteln und insbesondere die von Frischware, stellt die B2C-Logistik vor ganz besondere Herausforderungen, die sich aus den Kundenanforderungen einer schnellen und pünktlichen Zustellung und den benötigten Ressourcen ergeben. Die Kunden stellen in diesem Marktsegment enge Zeitvorgaben, weil sie die Zutaten zu Hause für die Zubereitung von Mahlzeiten benötigen. Gerade dieser Aspekt der vergleichsweise stark ortsgebundenen Verwendbarkeit der Produkte, läßt jedoch wenig Spielraum für alternative Lieferorte und rückt die Zustellung von Frischware in der Konsequenz außerhalb des Blickfelds dieser Arbeit. In der Folge soll daher dieser Teil des Artikelspektrums nicht weiter betrachtet werden.

2.1.3 B2C-Logistik und eCommerce

Nach (BTA 2002) kann eCommerce als Handelsgeschäft verstanden werden, das auf öffentlich zugänglichen Märkten unter Verwendung interaktiver elektronischer Medien abgeschlossen wird. Der Aspekt der Interaktivität ist hierbei wichtig und schließt damit Bestellungen per Call-Center, Telefax oder Bestellkarte aus. Unzweifelhaft hat eCommerce eine große Auswirkung auf den B2C-Handel, indem Angebote aktuell gehalten, Preise und Verfügbarkeitsinformationen dynamisch bestimmt und Leistungen

¹Die Lagerhaltung dieser Produkte besteht in der Speicherung der Produkte bzw. Buchungsdaten in Datenbanken auf digitalen Speichersystemen.

über Techniken wie Data Mining oder einfache Kaufhistorien kundenindividuell angepaßt werden können. Auch gänzlich neue Dienste, wie das Internet-Auktionshaus *ebay*, wurden in dieser Form erst durch eCommerce möglich.

Auch wenn die eCommerce-Umsatzzahlen sich im B2C-Handel nicht so stürmisch entwickelt haben, wie dies noch Ende der 1990er Jahre prognostiziert wurde (vgl. (Danzas 2002)), deutet die Entwicklung der Nutzerzahlen des Internet und der Zahl der online-Einkaufenden auf ein Wachstum des B2C-eCommerce hin: Bei einer immer noch wachsenden Zahl von Internetnutzern stellt die Studie (TNS 2002) einen nahezu gleichbleibenden Anteil von Menschen fest, die das Internet für Einkäufe einsetzen. Derzeit haben nach einer Haushaltsbefragung im Rahmen einer EU-Studie ((EU 2004)) bis zum Jahr 2004 21% der Bevölkerung in Deutschland mindestens einmal etwas über das Internet gekauft. Auch auf Seiten der Anbieter ist mit einem Wachstum zu rechnen, da das Internet nicht nur etablierten Versandhändlern einen zusätzlichen Kommunikations- und Vertriebskanal bietet, sondern sich insbesondere der Facheinzelhandel neue überregionale oder sogar globale Kundenkreise erschließen kann.

In der Literatur (siehe bspw. (Oberhuber 2003; Laseter, Houston, Chung, Byrne, Turner und Devendran 2000; Gregori 2003; Prockl 2001; Hahne 2001)) wird zuweilen der Eindruck erweckt, B2C-Logistik sei erst mit dem Aufkommen des eCommerce Mitte der 1990er Jahre relevant geworden. Die einleitenden Beispiele des Abschnitts 2.1.1 zeigen jedoch, daß B2C-Logistik zum einen verschiedene Formen annimmt und zum anderen alltäglich ist. Tatsächlich handelt es sich im Prinzip schon um ein sehr viel älteres Problem: Es kann auf die ersten rudimentären Ansätze des Versandhandels und damit auf das Jahr 1498 zurückdatiert werden kann, als (vgl. (Plattform Versandhandel 2003)) Aldus Manutius² einen Katalog von 15 seiner im eigenen Verlag herausgegebenen Bücher auflegte. Daß es seinerzeit gerade Bücher waren, die per Katalog vertrieben werden sollten, erscheint angesichts der aktuellen Entwicklung des Internethandels durchaus bemerkenswert.

Der quantifizierbare Einfluß des eCommerce auf die B2C-Logistik ist derzeit noch vergleichsweise gering, da gemessen am Umsatz der Anteil des eCommerce am gesamten Versandhandel (nach (Bundesverband des Deutschen Versandhandels e.V. 2004)) 22% beträgt, bezogen auf den Einzelhandelsumsatz lediglich 1,5%. Diese Zahlen berücksichtigen ein Artikelspektrum, wie es typischerweise im Versandhandel zu finden ist und nehmen digitale Produkte und Buchungen von Reisetickets oder Eintrittskarten von der Betrachtung aus.

Der Effekt des eCommerce ist weniger in veränderten physischen Abläufen oder in stark gewachsenen Sendungsvolumina zu sehen. Vielmehr müssen die Logistiksysteme der Versender auf infolge des eCommerce veränderte Kundenerwartungen reagie-

²Aldus Manutius lebte von 1449/50 bis 1515 und war ein wegen der ästhetischen Qualität seiner Bücher auch heute noch geschätzter venezianischer Buchdrucker und Herausgeber. Ihm wird das erste jemals in Kursivschrift gedruckte Buch zugeschrieben. Dies geschah jedoch weniger aus ästhetischen, denn aus wirtschaftlichen Gründen, da die kursive Schrifttype einen geringeren Platzbedarf hatte.

ren und sich auf ein höheres Anspruchsniveau einstellen.

Für den Kunden stellt sich der Einkauf im Internet als eine bequeme und zumeist schnelle und sichere Möglichkeit dar. Diese Einfachheit weckt allerdings gleichzeitig auch eine Erwartungshaltung bezüglich der Erfüllung der Bestellung (s. (Retail Logistics Task Force 2001)). Wird beispielsweise im Internet-Shop der Eindruck sofortiger Lieferbarkeit erweckt, erwartet der Kunde auch eine Zustellung innerhalb von einem oder zwei Tagen. Dabei ist es wichtig festzustellen, daß aus Sicht des Kunden erst der tatsächliche Empfang der Ware eine erfolgreiche Bestellung abschließt, was Lasch und Lemke folgendermaßen formulieren:

„Es reicht nicht aus, beim Kauf die schnelle und bequeme Lieferung in die Wohnung zu versprechen und den Kunden dann mit der Benachrichtigungskarte im Hausbriefkasten über den fehlgeschlagenen Zustellversuch zu konfrontieren.“

2.1.4 B2C-Logistik und Verkehr

Die vorangegangene Darstellung charakteristischer Elemente der B2C-Logistik zeigt das Umfeld auf, innerhalb dessen Versender und Dienstleister ihre logistischen Leistungen erbringen. Neben dieser mikroökonomischen Betrachtung aus der Perspektive einzelner Unternehmen sind auch soziale, umweltbezogene und gesamtwirtschaftliche Betrachtungen interessant. Der Zusammenhang zwischen B2C-Logistik und Verkehr spielt hierfür eine herausragende Rolle, da besonders Städte und Kommunen Aspekte des Umweltschutzes und der urbanen Lebensqualität in der Stadtplanung und -entwicklung berücksichtigen müssen. Während rein intuitiv erwartet werden kann, daß eine wachsende Zahl von Heimbeförderungen zu steigendem Verkehrsaufkommen mit entsprechenden Folgen in den Innenstädten führt, zeigt die Auswertung mehrerer Studien keine eindeutige Bestätigung dieser Annahme.

Optimistische Schätzungen gehen davon aus, daß eine Beförderungsfahrt mehrere private Einkaufsfahrten ersetzt, während pessimistische Erwartungen von einer Substitution von Einkaufsfahrten durch andere Fahrten (bspw. Freizeitverkehr-) und damit von einem wachsenden Verkehrsaufkommen ausgehen. Nach (Siebel und ten Hompel 2001) ist der Effekt des zunehmenden Warenbezugs über online-Medien in seiner Gesamtwirkung noch nicht abzuschätzen. Einerseits wird eine Reduzierung der Transportzahl mit LKW zu Einzelhandelsfilialen und ein Wachstum der Transportzahl mit kleineren Verteilfahrzeugen zu den Kunden gerechnet, woraus insgesamt nach (Siebel und ten Hompel 2001) ein Anstieg der Transportzahlen resultiert. Andererseits reduzieren sich der Parksuchverkehr und die Zahl der Einkaufsfahrten. Inwiefern sich insgesamt eine Erhöhung oder eine Verringerung der Transportzahlen ergibt, ist noch ungeklärt.

In Bezug auf den verkehrlichen Effekt der B2C-Logistik kommt (Verra 2004) zu dem Schluß, daß sich insgesamt die Fahrleistung – d.h. die mit der Entfernung bewerte-

te Anzahl von Fahrten – im Einkaufsverkehr reduziert. Demgegenüber wird zwar ein Wachstum der Fahrleistung im Lieferverkehr erwartet, das aber die Reduktion im Einkaufsverkehr nicht kompensiert.

Für eine Untersuchung zur Haustürzustellung, die auch umweltbezogene Zielkriterien enthält, wird auf (Punakivi 2003) verwiesen. Die Darstellung in dieser Arbeit konzentriert sich auf die operativen Aspekte der B2C-Logistik. Im nächsten Abschnitt werden in diesem Zusammenhang die besonderen Herausforderungen an die Logistik erläutert.

2.2 Herausforderungen der B2C-Logistik

Neben dem Anpassungsdruck, der von den veränderten Kundenerwartungen ausgeht, befinden sich Versender und Dienstleister im Wettbewerb miteinander und stehen vor der ständigen Aufgabe der Prozessverbesserung und Kostensenkung. Kundenerwartungen, Wettbewerber und die eigenen unternehmerischen Ziele bilden das Spannungsfeld für die Logistikstrategie der versendenden Unternehmen. Im Rahmen der Strategiefindung entscheiden Unternehmen über die grundsätzlichen Unternehmensziele, die einzusetzenden Ressourcen und die zu etablierenden Strukturen. Grundsätzlich lassen sich aus betriebswirtschaftlicher Sicht (vgl. bspw. (Kummer 2002)) zwei Grundstrategien unterscheiden, die sich aus dem Ökonomischen Prinzip ergeben. Durch die Strategie der Differenzierung versuchen Unternehmen eine herausgehobene Position auf bestimmten – für die Abnehmer entscheidenden – Feldern zu erreichen. Demgegenüber verfolgt die Strategie der Kostenführerschaft das Ziel minimaler Gesamtkosten bei mit dem Wettbewerb vergleichbaren Leistungen. Für die Logistik im B2C-Umfeld ergeben sich nach (de Koster 2002) u.a. folgende strategische Fragestellungen:

- Wie viele bestandsführende Stufen hat das Distributionssystem?
- Wie viele Lagerstandorte werden errichtet und wo?
- Welches Sortiment wird angeboten?
- Wie groß ist das abgedeckte Liefergebiet?
- Welche Struktur hat das Distributionssystem: Wird direkt ab einem Zentrallager geliefert oder wird ein mehrstufiges System verwendet?
- Werden die Lagerhaltung, der Transport, beide oder keines von beiden extern vergeben (*Outsourcing*)?
- Welche Art von Belieferung wird angeboten: Haustürzustellung oder ein fester oder mit dem Kunden vereinbarter Abgabeort?

Ergänzt werden diese strategischen Fragen um taktische Aspekte, die die konkrete Ausgestaltung der Strategieelemente zum Ziel haben. Zu den Entscheidungen zählen nach (de Koster 2002):

- Welche Partner übernehmen im Falle des Outsourcing die Aufgaben?
- Welche Produkte bzw. Kundenaufträge werden aus dem Lager bedient, welche direkt vom Lieferanten und für welche wird Crossdocking (s. z.B. (Alicke 2003)) eingesetzt?
- Welche Zeitfenster werden für die Belieferung angeboten? Kann der Kunde die Zeitfenster wählen?
- Muß der Kunde für die erfolgreiche Auslieferung anwesend sein?
- Werden die Logistikkosten den Kunden offen oder verdeckt in Rechnung gestellt?

Die strategischen Fragestellungen adressieren logistische Entscheidungen, die hauptsächlich mit zwei Prozesskategorien in Verbindung stehen: Zum einen dem Kommissionierprozess innerhalb der Distributionsstandorte, zum anderen dem Transportprozess zum Endkunden. Diese Schlüsselprozesse betreffen die physischen Materialflüsse und werden von weiteren ergänzenden Prozessen unterstützt.

2.2.1 Kommissionierung

Die Herausforderungen in der Kommissionierung liegen in einer gegenüber der B2B-Relation veränderten Struktur der Aufträge, die unter dem Schlagwort „Atomisierung“ bekannt ist (s. bspw. (Daduna 2000; Rahn 2004)). Nach (de Koster 2002) umfassen die Kundenaufträge üblicherweise nur wenige Positionen, so daß bei einer einstufigen Kommissionierung (s. (Gudehus 2002)) der Anteil der Weg-, Tot- und Basiszeiten pro Auftrag vergleichsweise hoch ist. Im Versandhandel finden sich daher zumeist zweistufige Kommissioniersysteme, bei denen in der ersten Stufe eine artikelorientierte Entnahme und in der zweiten Stufe eine Sortierung und Verteilung der Artikel auf Kundenaufträge stattfinden. Die zweite Kommissionierstufe sorgt im Vergleich zu Kommissioniersystemen im B2B-Handel, in denen üblicherweise ganze Gebinde (Paletten) aus dem Hochregallager des Distributionszentrums kommissioniert und verladen werden, für eine gesteigerte Komplexität. Nicht nur die gewachsene technische und organisatorische Komplexität, sondern auch die reine Zahl von (kleineren) Aufträgen stellt für auf B2B-Versand ausgelegte Kommissioniersysteme aufgrund von Beschränkungen von Puffern, Transport- oder Informationssystemen ein Problem dar.

Technische Einrichtungen und betriebliche Abläufe unterscheiden sich demnach stark zwischen B2C- und B2B-Kommissioniersystemen. Die enge Verbindung zwischen Kommissionierung, Lagerhaltung und Bestandsmanagement sorgen nach (Daduna 2000) dafür, daß der Lagerbetrieb als zum Kerngeschäft gehörig angesehen und daher nicht an externe Dienstleister abgegeben wird. Diese Feststellung wird von der empirischen Studie zur B2C-Logistik (Lasch und Lemke 2002) gestützt.

2.2.2 Transport

Im Gegensatz zur Kommissionierung wird der Transport der Sendungen zu den Endkunden gemäß (Daduna 2000) zumeist spezialisierten Dienstleistern überlassen. Hierbei handelt es sich überwiegend um Kurier-, Express- und Postdienste (KEP-Dienste), in Einzelfällen – wie im Fall des *Otto-Versands* – verfügt der Versender über (konzern)eigene Transportdienstleister.

KEP-Dienste greifen für den Sendungstransport von Absender zu Empfänger auf ein Transportnetzwerk bestehend aus Depots und Umschlageneinrichtungen (s. (Thomas 2002)) zurück. Häufig entspricht die Struktur einem mehrstufigen Netzwerk mit einem *Hub & Spoke*-Aufbau. Die Abwicklung der Transporte geschieht in den drei Phasen Vor-, Haupt- und Nachlauf. Im Vor- und Nachlauf bilden die Depots Ausgangs- und Endpunkte von regionalen Sammel- und Verteiltouren. Im Hauptlauf werden die Sendungen über einen oder mehrere *Hubs* geleitet, die als zentrale Umschlageneinrichtungen die Sortierung und Verteilung auf die Empfangsregionen übernehmen. Direktfahrten zwischen den Depots finden nur auf aufkommensstarken Relationen statt, die es erlauben, Transportmittel voll auszulasten. Im Nachlauf werden die Sendungen in den Empfangsdepots auf die Auslieferungsfahrzeuge verteilt und an die Empfänger ausgeliefert. Für weitere Ausführungen zum Aufbau von KEP-Netzen wird auf (Blunck 2005) und (Thomas 2002) verwiesen.

Die technischen Einrichtungen einschließlich der Identifikations- und Informationstechnik in Depots und Hubs erlauben eine effiziente und hochgradig automatisierte Abwicklung des Transports von Sendungen mit Abmessungen und Gewicht innerhalb bestimmter Grenzen³. Sendungen, die von diesen Maßen abweichen, werden gesondert und oftmals manuell behandelt. Für den deutschen Markt schätzt Rahm die Zusammensetzung des Sendungsspektrums gemessen am Umsatz auf 56% für Paketsendungen, 25% für großvolumige Sendungen (bspw. Möbel oder weiße Ware) und 19% für Lebensmittel. Nach (Retail Logistics Task Force 2001) handelt es sich bei etwa 93% aller Sendungen im B2C-Sektor in Großbritannien um Paketsendungen, bei weiteren 5% um großvolumige Sendungen und bei den verbleibenden 2% um Lebensmittelsendungen.

Gerade dieser hohe Anteil von Paketsendungen begründet die Zusammenarbeit mit darauf spezialisierten Dienstleistern. Durch die externe Vergabe müssen keine eigenen Transportkapazitäten aufgebaut und vorgehalten werden. Neu in den Markt eintretenden Unternehmen wird der Zugang erleichtert. Andererseits erschwert es den versendenden Unternehmen, den Kunden differenzierte Lieferoptionen anzubieten, da stets gewährleistet werden muß, daß der Dienstleister in der Lage ist, diese auch umzusetzen.

³Nach (Thomas 2002) liegen diese Maße für ein Paket bei der Deutschen Post AG (heute DHL) bei 1200mm × 600mm × 600mm und einem Gewicht von 31,5kg.

2.2.3 Logistik als Wettbewerbsinstrument

KEP-Dienste bieten heute für den Produktbereich der Expresszustellungen preislich abgestufte garantierte Liefertermine. Die Marktentwicklung wird (Daduna 2000) zufolge dazu führen, daß auch für den Bereich der Paketzustellung kundenindividuelle Transportleistungen angeboten werden. Mögliche Angebotsformen umfassen die kundenseitige Auswahl aus mehreren vorgegebenen oder gänzlich frei wählbaren Lieferzeitfenstern. Auch (Lasch und Lemke 2002) schlagen individualisierte oder auf Kundengruppen zugeschnittene Leistungen und Preise für die Distributionslogistik vor. Entsprechend (Retail Logistics Task Force 2001) können gegenüber heute stärker individualisierte Belieferungsoptionen sogar zu einem weiteren Wachstum des (eCommerce) Distanzhandels beitragen, d.h. die Logistik wandelt sich von einer Hilfsfunktion zu einem Gestaltungsinstrument.

Von (Lasch und Lemke 2002) wird das heutige Leistungsangebot von Online-Shops hinsichtlich der Logistik betrachtet. Es wird deutlich, daß üblicherweise nur eine Standardform der Belieferung angeboten wird: Eine schnellere Lieferung oder auch die Vereinbarung eines genauen Lieferdatums bieten 45% bzw. 44% der befragten Unternehmen an, die Vereinbarung eines Lieferzeitfensters erlauben 33%. Für die Zukunft streben 16% der Unternehmen verstärkt die Einbeziehung von sog. Abholpunkten an. Eine Reduktion der Lieferzeit⁴ und eine Vereinbarung von genauen Lieferdaten mit dem Kunden gehören zu den weiteren geplanten Maßnahmen. Die Möglichkeit, ein individuelles Lieferzeitfenster zu vereinbaren, wird von 9% der Unternehmen anvisiert. Kaum ausgebaut werden soll hingegen die Mehrfachlieferung bei Abwesenheit. Interessanterweise glauben allerdings nur 21% aller Unternehmen, daß online bestellende Kunden individuelle Lieferleistungen wünschen. Dies kollidiert mit der Einschätzung der Autoren der Umfrage, wonach sich der Trend im Kundenverhalten hin zum bequemen Einkauf von zu Hause aus verändert und demzufolge von den Anbietern angepaßte Angebote erstellt und umgesetzt werden müssen. Nach (Lasch und Lemke 2002) sowie (Wagener & Herbst Management Consultants GmbH 2002) erwarten die Kunden in Zukunft mit einer sehr hohen Wahrscheinlichkeit die Lieferung zu frei bestimmbar und exakt einzuhaltenden Zeiten bei gleichzeitig einfacher Abstimmung von Lieferzeiten und -modalitäten. Ebenso werden im Retourenfall Abholungen und die Bündelung mehrerer Bestellungen erwartet.

In Bezug auf die Logistikdienstleister berichten (Lasch und Lemke 2002) von einer selbst für Standardlieferungen niedrigen Qualität der Logistikleistungen in Bezug auf Lieferzeit und Pünktlichkeit. Bei den von den Autoren der Studie aufgenommenen Qualitätsdaten zeigt sich, daß spezialisierte Dienstleister am ehesten in der Lage sind, Standardlieferungen zu den versprochenen Bedingungen abzuwickeln. Bei vom Standard abweichenden Lieferungen, etwa der Zustellung an Abholpunkte, stellen Lasch und Lemke jedoch noch Verbesserungspotential fest und beziehen sich hierbei insbe-

⁴Eine Analyse der Zusammensetzung und Bedeutung der Lieferzeit innerhalb der Supply Chain liefert (Zillus 2003).

sondere auf die Pünktlichkeit der Zustellung.

2.2.4 Die Problematik der Feinverteilung

Die unbefriedigende Qualität der Zustellung resultiert aus der unzureichenden Planbarkeit der Feinzustellung durch die Transportdienstleister. Neben den Unsicherheiten bezüglich der Fahrzeiten, die sich infolge von Verkehrsstauungen ergeben und in der Praxis zumeist durch Erfahrungswerte und Ortskenntnis der Fahrer ausgeglichen werden, tritt im Fall der B2C-Zustellung das Problem abwesender Empfänger auf. In (Retail Logistics Task Force 2001) wird dieses Phänomen treffend beschrieben:

„What a co-incidence that as more and more goods are delivered to customers' homes, these customers are less and less likely to be at home to receive them!“

Mangels Absprache und Vorabinformation treffen die Paketzusteller nicht auf den Empfänger der Sendung und können nur eine Nachricht hinterlassen oder versuchen, die Sendung in der Nachbarschaft abzugeben. Letzteres ist zwar gängige Praxis, im Hinblick auf Fragen der Haftung, Zustellsicherheit, Diskretion und Servicequalität jedoch mindestens unbefriedigend (s. (Retail Logistics Task Force 2001)). Im Speziellen liegt das Problem bei abwesenden Empfängern darin, daß der Zusteller die Situation erst vergleichsweise spät erkennen kann. Dies wird deutlich bei Betrachtung der üblichen Abläufe eines Zustellversuchs:

1. Parken des Zustellfahrzeugs,
2. Heraussuchen der Sendung,
3. Fahrzeug abschließen,
4. Gebäudeeingang und Klingel suchen,
5. klingeln und warten auf den Kunden,
6. evtl. Treppen laufen oder Aufzugsfahrt in die richtige Etage,
7. Sendung aushändigen und quittieren lassen,
8. Weg zurück zum Fahrzeug.

Mindestens sechs dieser acht Punkte sind auch im Fall eines vergeblichen Zustellversuchs erforderlich, so daß sich die Standzeiten bei erfolgloser bzw. erfolgreicher Aushändigung nicht deutlich unterscheiden. Für eine erfolgreiche Abgabe bei einem Nachbarn muß erst ein dazu bereiter Hausbewohner gefunden werden, was die Standzeit zusätzlich verlängert ohne eine Erfolgsgarantie zu bieten.

Die Einschätzungen der Tragweite des Problems sind unterschiedlich. Nach (Retail Logistics Task Force 2001) sind bis zu 60% der Erstzustellversuche nicht erfolgreich, (Verkehrsforum 2004) geht von einer Erreichbarkeitsquote bei Privathaushalten von 70–80% aus, d.h. ca. ein Viertel der Empfänger wird vom Fahrer im ersten Zustellversuch nicht angetroffen. Die Kosten werden von (Verkehrsforum 2004) auf mindestens 2,50 €, die bei ungünstigen Lieferumständen (Treppengang in anderes Stockwerk) noch höher liegen können, quantifiziert. Damit werden mehr als zwei Zustellversuche von KEP-Dienstleistern als unwirtschaftlich angesehen, wobei es allerdings am Markt nicht durchsetzbar ist, dem Kunden weitere Zustellversuche gesondert zu berechnen.

Während eines Stops wird zumeist nur eine Sendung ausgeliefert. Dieser sog. niedrige *Stopfaktor* und die vergleichsweise lange Standzeit bedeuten in der Konsequenz einen hohen Kostenanteil der Feinverteilung an den gesamten Distributionskosten. Laut (Verkehrsforum 2004) liegt der Anteil der auf die *Letzte Meile* entfallenden Distributionskosten an den gesamten KEP-Distributionskosten zwischen 50 und 75%. Würden vergebliche Zustellversuche und Zweitzustellungen durch neue Konzepte wegfallen, könnten nach (Oberhuber 2003) die Paketdienste zwischen 10–30% ihrer Distributionskosten einsparen.

Der Begriff „Letzte Meile“ ist bislang nicht einheitlich definiert. In Anlehnung an (Verkehrsforum 2004) wird hier darunter der letzte Abschnitt einer Transportkette zwischen dem Ende des Hauptlaufs und dem Erreichen des Empfängers verstanden. Innerhalb des Transportnetzwerks eines KEP-Dienstleisters bezeichnet die Letzte Meile damit die Distribution im Nachlauf zuzüglich der Sortierung, Verteilung und Verladung der Sendungen auf die Zustellfahrzeuge im Empfangsdepot.

Aus den vorangegangenen Ausführungen lassen sich in operativer Hinsicht zwei Handlungsfelder für eine Steigerung der Effizienz in der B2C-Logistik ableiten. Auf Seiten der Versender sind auf den B2C-Markt zugeschnittene Kommissioniersysteme zu entwickeln und zu betreiben. Transportdienstleister benötigen Konzepte und unterstützende Werkzeuge für die Abwicklung der Distribution auf der Letzten Meile. Auf beiden Handlungsfeldern müssen die Lösungen den Kundennutzen, d.h. insbesondere Qualität, Flexibilität und Lieferzeit, einbeziehen und bei gegebenen Kostenstrukturen wirtschaftlich sein.

Im Rahmen dieser Arbeit wird die logistische Herausforderung der Letzten Meile untersucht. Im folgenden Abschnitt werden darauf zugeschnittene Lösungskonzepte betrachtet. Für die Behandlung des Aspekts der Kommissionierung wird auf die Arbeit (Kämäräinen 2003) und den Artikel (Kämäräinen, Småros, Jaakola und Helmström 2001) verwiesen.

2.3 Lösungskonzepte für die Letzte Meile

In diesem Abschnitt sollen Lösungskonzepte für das Distributionsproblem der Letzten Meile vorgestellt werden. Zu diesem Zweck werden die den Konzepten zugrunde liegenden Ansätze und Ideen herausgearbeitet und beurteilt, wobei die Kundenperspektive im Vordergrund steht.

Grundsätzlich lassen sich zwei verschiedene Ansätze identifizieren:

Entkopplungskonzepte: Entkopplung der Zustellung von der Anwesenheit des Empfängers,

Abstimmungskonzepte: Abstimmung der Zustellung zwischen Empfänger und Transportdienstleister.

Konzepte, die den ersten Ansatz verfolgen, sind primär kosten- und anbieterorientiert, während die dem zweiten Ansatz entsprechenden eine stärkere Serviceorientierung aufweisen. Das zentrale Gestaltungselement aller Ansätze besteht in der Wahl des Übergabepunkts der Sendungen vom Transportdienstleister an den Empfänger. Die persönliche Übergabe an den Empfänger und die Abholung am Depot des Dienstleisters bilden dabei die Extreme zwischen denen sich eine Vielzahl von Konzepten mit wachsendem Bündelungsgrad – dem bereits zuvor erwähnten Stop-Faktor – ansiedelt. Mit steigendem Bündelungsgrad übernehmen die Kunden einen größeren Anteil der Distribution und eröffnen dem Transportdienstleister damit ein entsprechendes Kostensenkungspotential.

Wird eine Strategie der Leistungsdifferenzierung anstelle der Kostenführerschaft verfolgt, können Abstimmungskonzepte, bei denen Empfänger und Dienstleister durch Informationsaustausch und über Anreize miteinander kooperieren, für den Anbieter interessant sein. In den folgenden Abschnitten werden Entkopplungskonzepte und Abstimmungskonzepte vorgestellt. Für die Entkopplungskonzepte existieren eine Reihe von empirischen Studien zu Kundenerwartungen und -akzeptanz, die für eine Beurteilung herangezogen werden. Abstimmungskonzepte sind noch überwiegend theoretischer Natur und Gegenstand aktueller Forschungen, eine Bewertung ist daher größtenteils derzeit noch nicht möglich. Für alternative Liefermöglichkeiten als Bestandteil von Abstimmungskonzepten liefert diese Arbeit eine erste Bewertung in Kapitel 7.

Neben diesen explizit auf die Übergabe abzielenden Konzepten wurden auch Lösungen für vorgelagerte Prozesse entwickelt. Von (Rahn 2004) wurde ein Modulsystem zur Verbesserung der Beladung von KEP-Fahrzeugen vorgestellt. Es basiert auf der Idee, Sendungen im Depot in ein bewegliches Regalsystem zu kommissionieren, das dann als Ganzes in das Fahrzeug gerollt und dort befestigt wird. Das System ist modular aufgebaut und erlaubt somit eine einfache Anpaßbarkeit an unterschiedliche Fahrzeugtypen, wie sie im KEP-Bereich eingesetzt werden. Im Vergleich zum herkömmlichen Ansatz ist die Prozesskette von der Sendungskommissionierung bis zur Fahrzeugbeladung damit effizienter gelöst.

2.3.1 Entkopplungskonzepte

Entkopplungskonzepte sehen vor, daß der Dienstleister die Sendungen an einem Ort hinterlegt, an dem die Kunden sie nach erfolgreicher Legitimation abholen können. Für die Kunden kann diese Vorgehensweise ohne bzw. mit nur geringen Einschränkungen der Servicequalität verbunden sein, wenn die Wege zum Abholungsort sich mit anderen Wegen, bspw. zur Arbeitsstelle, ohne größere Umwege kombinieren lassen. Die Abholung durch den Kunden ist für Dienstleister besonders in ländlichen Regionen interessant, in denen die durchschnittliche Fahrzeit zwischen zwei Stops größer als in der innerstädtischen Zustellung ist, d.h. in denen die Stopdichte niedriger ist.

Im Laufe der letzten Jahre sind verschiedene Entkopplungskonzepte entwickelt und teilweise auch umgesetzt worden, deren Implementierungsgrad von prototypischen Demonstrationseinsätzen bis hin zu voll operativen Systemen reicht. Im Folgenden sollen verschiedene Entkopplungskonzepte kurz vorgestellt werden, wobei zwischen Systemen mit hohem, mittlerem und niedrigem Bündelungsgrad differenziert wird. Nicht eingegangen wird auf die Möglichkeiten der Bezahlung und Benachrichtigung, weil sämtliche Konzepte, bei denen die Sendungen zur Abholung hinterlegt werden, eine automatische Benachrichtigung über die üblichen Kommunikationskanäle wie Telefon, SMS oder eMail und die bargeldlose Bezahlung anbieten.

Entkopplungskonzepte mit hohem Bündelungsgrad

Geht man davon aus, daß einzelne Empfänger nur eine oder zumindest wenige Sendungen erhalten, läßt sich ein hoher Bündelungsgrad offensichtlich nur dann erreichen, wenn eine große Sendungsmenge für eine Vielzahl von Empfängern am Abholort hinterlegt wird. Dies impliziert zum einen, daß der Abholort ein relativ großes Einzugsgebiet haben muß und zum anderen, daß das System vom Durchsatz her leistungsfähig genug sein muß, um die Sendungen übergeben zu können. Als Aufstellorte bieten sich daher stark frequentierte und gut per PKW oder ÖPNV erreichbare Orte an. Um die erforderlichen Durchsätze zu erreichen und für die Kunden und Einlieferer nur geringe Wartezeiten zu gewährleisten, sind automatisierte Lösungen gestaltet worden. Bekannte Beispiele sind der vom Fraunhofer-Institut für Materialfluss und Logistik (IML) in Dortmund entwickelte *Tower24* sowie die *PackStation* von DHL.

Der Tower24 (s. Abbildung 2.1) wird zwar aus wirtschaftlichen Gründen nicht weiterverfolgt, ist jedoch konzeptionell so interessant, daß er hier kurz beschrieben werden soll. Es handelt sich um ein zylindrisches automatisches Kleinteilelager, das in zwei Temperaturbereiche aufgeteilt ist und auch Kühlware aufnehmen kann. Bei 10m Gesamthöhe, davon 4m unter der Oberfläche, und einem Durchmesser von 4m waren Einkaufszentren und größere Parkplätze als Aufstellungsorte gedacht. Das Aufnahmevermögen beträgt 200 bis 400⁵ ringförmig angeordnete Behälter standardisierter

⁵Bei doppeltiefer Lagerung kleiner Behälter, s. auch (Lippolt 2003) zum Thema doppeltiefer Lagerung.



Abbildung 2.1: Das automatische Lagersystem Tower24 (Quelle: Fraunhofer Institut Materialfluss und Logistik)

Größe. Der Zugriff erfolgt über ein Regalbediengerät (RBG) mit zwei Lastaufnahmemitteln. Angaben zu Durchsätzen sind dem Verfasser nicht bekannt, allerdings gehen (Siebel und ten Hompel 2001) von einer Wartezeit von 3 bis 5 Minuten aus und der Hersteller SSI Schäfer (s. (SSI Schäfer 2003)) gibt an, daß mehrere Kunden parallel bedient werden können. Die Einlieferung einer Sendung soll laut (Verkehrsforum 2004) weniger als eine Minute dauern, was angesichts der angestrebten Sendungsmenge unangemessen lang erscheint⁶.

Die Investitionskosten belaufen sich nach (Oberhuber 2003) auf ca. 300.000 €, nach (Verkehrsforum 2004) muß der Betreiber angesichts Investitions- und Betriebskosten zwischen 0,80 € und 1,00 € pro Paket Erlösen, um die Rentabilitätsschwelle zu erreichen, die nach (Oberhuber 2003) bei rund 100 Paketen pro Tag liegt. Das System ist als offenes System ausgelegt und erlaubt es somit prinzipiell jedem Dienstleister, es mit Sendungen zu beschicken. Das Geschäftsmodell ist dadurch allerdings komplexer und nach (Bundesverband Deutscher Postdienstleister e.V. 2004) sind es gerade Wirtschaftlichkeitsüberlegungen, die dazu geführt haben, daß das Konzept herstellerseitig nicht weiterverfolgt werden soll. Bis zu jenem Zeitpunkt war eine Demonstrationsanlage in Dortmund errichtet worden.

In der Praxis erfolgreicher ist das PackStation-System (s. Abbildung 2.2), das zum Zeitpunkt der Erstellung dieser Arbeit über knapp 100 PackStationen in Deutschland verfügte und weiter ausgebaut werden soll. Es handelt sich dabei um ein Schließfach-

⁶Dies ist eine persönliche Einschätzung des Verfassers. Zwar sind strenggenommen wenige Sekunden auch „weniger als eine Minute“, allerdings kann man annehmen, daß in diesem Fall die Formulierung in der Quelle auch „in wenigen Sekunden“ gelautet hätte.



Abbildung 2.2: Das Schließfachsystem PackStation (Quelle: Deutsche Post World Net)

system, das derzeit als geschlossenes System ausschließlich von DHL genutzt und in der Nähe stark frequentierter Orte, bspw. Bahnhöfe, Universitäten oder großer Bürohäuser (bspw. beim Softwareunternehmen SAP in Walldorf, s. (mylogistics.net 2004)), installiert wird. Die Packstationen werden in drei Größen mit 56, 76 bzw. 180 Fächern ((Rindfleisch 2005)) aufgestellt und haben Fächer in vier unterschiedlichen Größen. Die technische Ausführung der Einlieferung erlaubt es nach Herstellerangaben, ein Paket in ca. 10 Sekunden einzuliefern.

Entkopplungskonzepte mit mittlerem Bündelungsgrad

Neben diesen automatisierten Lösungen sind auch Konzepte für mittlere Durchsätze entwickelt und umgesetzt worden, bei denen die Sendungen den Empfängern persönlich bei der Abholung übergeben werden. Typischerweise finden sich die Abholorte in Einrichtungen wie Tankstellen, Kiosken, Videotheken, Sonnenstudios und ähnlichen Lokalitäten mit langen Öffnungszeiten und guten Parkmöglichkeiten. Die Einzugsbereiche sind deutlich kleiner gegenüber automatisierten Lösungen und somit ist auch die pro Abholort zu bewältigende Sendungsmenge geringer. Anders als bei den Konzepten für einen hohen Bündelungsgrad ist die logistische Dienstleistung für die (zumeist kleinen) Partnergeschäfte ein reines Zusatzgeschäft und nicht der primäre Geschäftszweck.

Beispiele für diese Konzepte sind die *PickPoints* der PickPoint AG und die *Paketshops* des Hermes Versand Service. Während es sich bei den Letzteren wiederum um ein geschlossenes System handelt, sind die PickPoints jedem Dienstleister zugänglich. Dank

im Vergleich zum Tower24 deutlich niedrigerer Investitionskosten konnte sich dieses offene System, bei dem die Kunden eine Gebühr pro Lieferung (s. (Mucha, Spille und Wader 2002)) entrichten, bereits seit Herbst 2000 am Markt etablieren. Nach Unternehmensangaben ((PickPoint AG 2005)) wurden zu Beginn des Jahres 2005 über 2000 PickPoints in Deutschland betrieben.

Entkopplungskonzepte mit niedrigem Bündelungsgrad

Während die vorgenannten Entkopplungskonzepte eine Bündelung der Sendungsmengen gezielt anstreben, finden sich auf der anderen Seite solche mit einem sehr niedrigen Stop-Faktor bzw. gänzlich ohne Bündelung. In der Literatur werden diese Konzepte häufig als „Boxsysteme“ bezeichnet. Es handelt sich im Wesentlichen um Paketbriefkästen, die einzeln entweder unmittelbar am Haus oder in größerer Anzahl für ganze Gebäudeanlagen aufgebaut werden. Technisch können die Paketbriefkästen von unterschiedlicher Komplexität sein und bspw. auch die kurzzeitige Lagerung von kühlbedürftiger oder sogar tiefgekühlter Ware erlauben. Sicherungseinrichtungen über Codeschlösser mit PIN-Eingabe und evtl. vorhandene Kühlaggregate benötigen eine Energieversorgung, was den nachträglichen Einbau aus baulichen oder baurechtlichen Gründen erschwert oder sogar verhindert. Weitere Probleme ergeben sich für den Fall von Mehrfachbelieferungen, da die Boxen nach der Beschickung bis zur Entnahme gesperrt sind, und durch die beschränkten Abmessungen der Paketbriefkästen, die entsprechend der üblichen Maße in der KEP-Branche dimensioniert sind.

Das bei weitem größte Problem dieser Konzepte stellt jedoch die Wirtschaftlichkeit dar. Einerseits sind die Empfangsboxen je nach technischer Ausstattung sehr teuer in der Produktion – (Oberhuber 2003) gibt ein Beispiel einer von der Firma Electrolux entwickelten Kühlbox, deren Herstellkosten mit 3000 € angegeben werden – andererseits ist mangels Bündelung das Kostensenkungspotential auf die Zustellung in Abwesenheit beschränkt. Am Markt angebotene Konzepte (s. (Mucha, Spille und Wader 2002; Punakivi 2003)) sehen daher eine finanzielle Beteiligung durch den Empfänger in Form von Kauf oder Miete vor. Wie in Abschnitt 2.3.1 näher erläutert wird, weisen die Preisvorstellungen von Anbietern und Kunden jedoch erhebliche Diskrepanzen auf. Im Fall der erwähnten Kühlbox ist bspw. bei einer monatlichen Miete von 5 € mit einer außerhalb jeder Diskussion stehenden Amortisationszeit von 50 Jahren zu rechnen – ohne bereits Wartungs- und Reparaturkosten zu berücksichtigen.

Kundenerwartungen und -akzeptanz

In den vorangegangenen Abschnitten sind einige Entkopplungskonzepte im Hinblick auf Ansatz und praktische Umsetzung erläutert worden. Dabei sind Aspekte der Wirtschaftlichkeit, wie sie sich aus Anbietersicht darstellen, bereits aufgenommen worden. Kunden haben naturgemäß eine andere Perspektive bzgl. der Konzepte und entscheiden sich zwischen diesen vor dem Hintergrund ihrer Erwartungen und dem individuell

empfundene Grad der Eignung. An dieser Stelle werden einige Umfrageergebnisse aus der Literatur zusammengefaßt, um ein Bild dieser Kundenperspektive zu vermitteln. In die Betrachtung fließt neben den beschriebenen Entkopplungskonzepten auch die Haustürzustellung als Normalfall zum Vergleich ein.

Kunden entscheiden sich für bestimmte Konzepte aus Kosten- und Leistungsgründen. Selbst bei idealer Lage der Abholorte sind mit der Abholung keine Kostenersparnisse für die Kunden verbunden, sieht man einmal von möglichen finanziellen Anreizen ab. Aus Kundenperspektive zählen daher vor allem der Leistungsaspekt und die Servicequalität der Zustellung.

In der Tat sind die Kriterien Bequemlichkeit und Zeitersparnis entsprechend der Umfrage (Siebel und ten Hompel 2001) für die Kunden die ausschlaggebenden Faktoren bei einer Online-Bestellung und es zeigt sich, daß die Gruppe, die den größten Umsatz im Internet tätigt, auch die Heimbeförderung vorzieht. Servicequalität insgesamt geht darüber hinaus und umfaßt auch die Aspekte der Zahlungsabwicklung, Identitätsprüfung, Rücknahme, Entsorgung oder den Austausch von Gütern. Einige Konzepte sind nicht auf die Abwicklung von Retouren eingerichtet, Empfangsboxen erlauben keine Zahlungsabwicklung.

Eine bekannte Zusatzleistung ist die Verfolgung des Sendungsstatus, *Tracking & Tracing* genannt. Hierbei kann der Absender anhand einer Identifikationsnummer der Sendung deren Weg durch das Transportnetzwerk verfolgen. Auf Versenderseite kann die Mitteilung über die erfolgreiche Zustellung Auslöser für nachgelagerte Prozesse (Freischaltungen, Bezahlvorgänge usw.) sein und bildet ein definiertes Ende der Warentransaktion. Auch wenn kundenseitig keine Eingriffsmöglichkeiten bestehen, mag die Information in manchen Fällen für die Planung und Vorbereitung von mit der Lieferung zusammenhängenden Tätigkeiten nützlich sein. Zumindest kann Tracking & Tracing in der Kommunikation mit dem Kunden ein sinnvolles Instrument darstellen, um den Fortschritt der Auftragsabwicklung zu dokumentieren.

Die individuelle Lebenssituation der Kunden ist erwartungsgemäß maßgeblich für die Bevorzugung von Zustellsystemen. Nach der von (Fichtner und Engelsleben 2001) durchgeführten Umfrage ist die Haustürzustellung bei knapp der Hälfte aller Versandhandelskunden (44,3%) die bevorzugte Form der Zustellung. 29,8% bevorzugten ein Boxsystem, 25,8% eine Abholstation. Dabei zeigt sich das erwartete Ergebnis, wonach insbesondere ältere (über 40 Jahre) und unter 25jährige Kunden die Haustürzustellung bevorzugen (83,3%). In der mittleren Altersgruppe zwischen 25 und 40 Jahren lagen alle vorgeschlagenen Zustellmöglichkeiten (Boxsystem, Abholstation und Haustürlieferung) nahezu gleichauf. Auch der Wohnort bestimmt das Bild. In der Studie zum BMBF-Projekt B2C-VERRA (s. (Verra 2004)) zeigt sich, daß Abholkonzepte vorwiegend von Einwohnern der Innenstadt bevorzugt werden, während in ländlichen Gegenden die Haustürzustellung und Boxensysteme favorisiert werden.

Neben dem Ort ist auch der Zeitpunkt der Zustellung im Hinblick auf die Abwesenheit der Kunden wichtig. Heute stellt sich diese Situation so dar, daß zu den Auslieferzei-

bevorzugter Zeitraum	Häufigkeit
vor 8:00	4%
8:00 – 12:00	18%
12:00 – 14:00	6%
14:00 – 16:00	3%
16:00 – 18:00	9%
18:00 – 20:00	34%
20:00 – 22:00	4%
nach Vereinbarung	17%
keine Angabe	6%

Tabelle 2.1: Bevorzugter Lieferzeitraum

ten der KEP-Dienste die Empfänger oftmals außer Haus sind. Die von der Retail Logistics Task Force ermittelte Aufstellung der bevorzugten Zustellzeitpunkte (s. Tabelle 2.1) zeigt, daß eine Zustellung entweder vormittags oder abends nach Feierabend erfolgen sollte.

Bezüglich des Standorts von Schließfachanlagen zur Abholung geben die Studien ebenfalls Hinweise. Gemäß (Siebel und ten Hompel 2001) wird ein Standort in der Nähe des Arbeitsplatzes für die Abholung von Sendungen deutlich gegenüber anderen Standorten bevorzugt. Dies scheint im Widerspruch zur Feststellung von Fichtner und Engelsleben zu liegen, wonach die Verbraucher Standorte in Wohngebieten gegenüber Citylagen bevorzugen. Der Widerspruch löst sich auf, wenn bei etwas freierer Interpretation der Antworten angenommen wird, daß der Begriff „Citylage“ bei den Befragten Assoziationen mit Parkplatzsuche und Hektik auslöst, und die Antwortmöglichkeit „Wohngebiet“ stellvertretend für einen vertrauten und gut erreichbaren Ort steht. Daß neben harten Kriterien wie der Erreichbarkeit auch die Einstellung der Empfänger zum Abholort wichtig ist, hat ebenfalls die Konsumentenbefragung von (Fichtner und Engelsleben 2001) gezeigt. Demnach sind das Image des Abholortes und die Sicherheit bei der Abholung wichtige Aspekte. In der Folge werden Tankstellen bevorzugt, wohingegen Bahnhöfe und Kioske eher abgelehnt werden.

Entkopplungskonzepte weisen für die Transportdienstleister mit abnehmendem Bündelungsgrad ein sinkendes Kostensenkungspotential auf. Umgekehrt wächst die Eigenleistung der Kunden, da sie einen wachsenden Teil der Letzten Meile selber übernehmen. Es liegt daher nahe zu fragen, inwiefern monetäre Anreize die Wahl der Zustellform durch die Kunden beeinflussen können. Es zeigt sich deutlich, daß Kunden bereit sind, für eine größere Bequemlichkeit einen Aufschlag zu zahlen. Nach der Befragung von (Siebel und ten Hompel 2001) sind fast die Hälfte (46%) der Kunden mit einem Aufschlag für die Haustürzustellung einverstanden, während diese Zahl für automatisierte Schließfachsysteme bei 29% liegt. Die mittlere akzeptierte Aufschlaghöhe wird

mit 2,91 €⁷ für die Haustürzustellung bzw. 2,19 € für die Abholung angegeben.

Wie bereits zuvor dargestellt, ist die Wirtschaftlichkeit gerade von Boxsystemen nicht gesichert. Die Übernahme eines Teils der Kosten durch die Kunden könnte – falls am Markt durchsetzbar – eine wirtschaftliche Grundlage schaffen. Fichtner und Engelsleben haben in ihrer Umfrage nach der Zahlungsbereitschaft gefragt: Die Bereitschaft, Boxen zu mieten, ist niedrig und ein monatlicher Mietpreis von 5,11 € wurde als Höchstpreis angegeben. Zwischen den von Boxsystemanbietern angegebenen Verkaufspreisen von 127,82 € und dem von Kunden als Preisobergrenze genannten 35,79 € gibt es eine sehr deutliche Diskrepanz. Obwohl Boxsysteme von Menschen, die häufig im Versandhandel bestellen, durchaus bevorzugt würden, ist ein rentabler Betrieb angesichts der Umfrageergebnisse wahrscheinlich nicht möglich.

Die Umfragen zeigen eine eindeutige Bevorzugung der Haustürzustellung vor anderen Konzepten. Dies ist durch mehrere Faktoren beeinflusst. Zum einen ist die Haustürzustellung zweifelsohne die für die Kunden bequemste und einfachste Form der Zustellung. Zum anderen werden alternative Konzepte häufig nicht aufgrund mangelnder Eignung abgelehnt, sondern sind, wie (Retail Logistics Task Force 2001) und (Verra 2004) erkannt haben, schlichtweg unbekannt.

Solange dem Kunden die Kosten der Haustürzustellung nicht bewußt sind, wird es für ihn auch wenig Anlaß geben, davon abzuweichen. Versendern und Dienstleistern ist zu empfehlen – wollen sie verstärkt die Kosten senken – dem Kunden die Logistikkosten, gestaffelt entsprechend alternativer Zustellformen, bereits bei der Bestellung transparent zu machen und nicht durch pauschalierte Angaben zu verdecken. Die vom Kunden getroffene Auswahl wird dabei höchstwahrscheinlich fallweise unterschiedlich sein. Letztlich erwarten Kunden individuell und einfach abzustimmende Lieferungen mittels einfach zu benutzender physischer und informatorischer Schnittstellen (s. (Lasch und Lemke 2002)). Die im Folgenden beschriebenen Abstimmungskonzepte bieten Wege zur flexiblen Zustellorganisation.

2.3.2 Abstimmungskonzepte

In eine andere Richtung als Entkopplungskonzepte, die die unmittelbare Warenübergabe adressieren, gehen Konzepte einer zeitlichen und organisatorischen Abstimmung von Dienstleistern und Kunden. Die Koordination kann dabei in Form einfacher Terminabsprachen geschehen oder sich bestimmter Anreize – üblicherweise Rabatte – bedienen. Ziel ist es dabei jeweils, die Anwesenheit des Kunden zum Zustellzeitpunkt zu gewährleisten. Eine Sendungsbündelung wird nicht angestrebt.

Ein in der Praxis bereits umgesetztes Konzept stammt vom Unternehmen *BI-LOG* und wird unter dem Namen *BI-LOG direkt*⁸ angeboten. Vereinbart werden können

⁷Für die Darstellung in dieser Arbeit wurden die in der Quelle verwendeten DM-Angaben in € umgerechnet

⁸Dieses Angebot findet sich in der Literatur (vgl. (Mucha u. a. 2002)) auch unter dem Namen *Leos* für

in ganz Deutschland zwei-Stunden-Lieferzeitfenster und die Dienstleistungen gehen über die reine Belieferung hinaus, da sie auch Installations-, Legitimations- und Geräteaustauschdienste umfassen. Genutzt wird das Angebot insbesondere von Händlern technischer Geräte, bspw. Mobiltelefonen, und die Erfahrungen zeigen, daß durch die Terminvereinbarung die Kunden auch fast immer erreicht werden. Nach Angaben des Unternehmens (s. (BI-LOG 2004)) bewirkt der gebotene Service ein positives Unternehmensimage und eine gestiegene Kundenbindung.

Zwei Arbeiten von Campbell und Savelsbergh befassen sich theoretisch mit Optimierungsansätzen für die Haustürzustellung. Betrachtet wird jeweils ein dynamisches Problem, bei dem Kunden zwecks Vereinbarung eines Liefertermins in einer Dispositionszentrale anrufen. Der Disponent muß in kürzester Zeit unter Berücksichtigung bereits zugeteilter und noch zu erwartender Aufträge mit dem Kunden ein Zeitfenster vereinbaren. Gegenüber (Campbell und Savelsbergh 2002) wird das Problem in (Campbell und Savelsbergh 2003) dadurch erweitert, daß der Disponent die Entscheidungen des Kunden für bestimmte Zeitfenster durch Anbieten einer Auszahlung zu lenken versucht. Ebenfalls untersucht wird das Angebot von Prämien für die Wahl größerer Lieferzeitfenster, konkret zur Wahl von 2h-Zeitfenstern anstelle von 1h-Zeitfenstern.

Zur Lösung der Probleme werden jeweils verschiedene Varianten einer Einfügeheuristik entwickelt und experimentell verglichen. In der Ergebnisanalyse kommen die Autoren zu einigen interessanten Schlußfolgerungen. Der Einfluß der Länge von Zeitfenstern wird unabhängig von der Kundendichte als sehr stark angesehen: Längere Zeitfenster erhöhen die Flexibilität und senken damit die Transportkosten. Ebenfalls zu erhöhter Flexibilität führen kundenseitig angegebene alternative Zeitfenster (bei gleichem Lieferort). Bezüglich der Zeitfensterbreite stellen die Autoren fest, daß es effektiver ist, den Kunden für eine Vergrößerung des Lieferzeitfensters zu gewinnen, als eine konkrete Auswahl zu treffen. Der Grund hierfür liegt wahrscheinlich darin, daß im angenommenen Szenario mit sequentiell eintreffenden Terminanfragen, größere Zeitfenster eine höhere Flexibilität im Hinblick auf zukünftige Anfragen sichern. Weiterhin wird ein Zusammenhang zu der verwendeten Fahrzeugflotte beobachtet. Demnach ist es tendenziell besser, eine größere Flotte kleinerer Fahrzeuge als eine kleinere Flotte größerer Fahrzeuge einzusetzen, da insbesondere zu stark nachgefragten Zeiten auf diese Weise mehr Kunden gleichzeitig mit verschiedenen Fahrzeugen bedient werden können.

Auch (Kunze 2005) betrachtet ein dynamisches Dispositionsproblem. In seinem Anwendungsszenario müssen mit den a priori unbekanntem Kunden zeitgleich zur Bestellannahme Lieferzeitfenster für die Lebensmittelzustellung vereinbart werden. Die Kunden zeigen dabei eine unterschiedliche zeitliche Flexibilität und Toleranz gegenüber unpünktlicher Zustellung. Das Optimierungsziel besteht in der Maximierung der Anzahl von Auslieferungen. Es wird ein Verfahren entwickelt, daß Aufträge nach räum-

„Leichter einkaufen ohne Stress“.

lichen und zeitlichen Gesichtspunkten gruppiert und neu eintreffende Aufträge in das existierende Gerüst einsortiert. Bei Vorliegen einer großen Zahl von Aufträgen gegenüber dem Transportkapazitätsangebot und einem im Vergleich zum Bedienzeitanteil niedrigen Fahrzeitanteil kann Kunze beispielhaft zeigen, daß die üblicherweise feststellbare Abhängigkeit der Transportkosten von der Zeitfensterbreite nicht mehr besteht. Der Autor weist jedoch daraufhin, daß für eine weitergehende Bewertung ausgiebigere Untersuchungsreihen erforderlich sind.

Heute vereinbaren Paketdienste keine Liefertermine mit ihren Kunden in der Befürchtung, dadurch in ihrer Flexibilität zu stark eingeschränkt zu werden. Außerdem wird die Administration der Belieferungswünsche bei einer großen Zahl von Kunden als organisatorische Schwierigkeit gesehen. Es fehlt demnach an Werkzeugen, die den Dienstleistern eine Unterstützung anbieten und es dadurch ermöglichen, individualisierte Belieferungen zu vereinbaren. Der nächste Abschnitt stellt ein solches vom Verfasser konzipiertes Werkzeug vor.

2.4 Personal Logistics Assistant – ein Integrationsansatz

In den vorherigen Abschnitten wurden die logistischen Herausforderungen und existierende Lösungskonzepte auf der Letzten Meile dargestellt. Dabei ist es auffallend, daß die meisten Ansätze für die Effizienzsteigerung der Zustellung auf der Letzten Meile bei der physischen Abwicklung ansetzen und üblicherweise nicht kooperativ sind. Stattdessen müssen sich die KEP-Dienstleister nach Vorgaben der Kunden richten und tragen die wirtschaftlichen Risiken. Gerade moderne Ansätze der Gestaltung von Logistiksystemen und der Logistiksteuerung – üblicherweise unter dem Begriff *Supply Chain Management* subsumiert – zeigen jedoch die Vorteilhaftigkeit der Kooperation im Hinblick auf die Reduktion von Kosten und die Erhöhung der Kundenzufriedenheit (s. bspw. (Alicke 2003; Faißt 2003; Zillus 2003)). Kernelemente dieser Ansätze sind die Bereitstellung und der Austausch von Informationen, die es allen Partnern in der Kette erlauben, ihre Aktivitäten aufeinander abzustimmen. Gerade vor dem Hintergrund, daß B2C-Logistik in weiten Teilen durch eCommerce getrieben wird und damit per Definition ein interaktives Medium Verwendung findet, liegt der Gedanke nahe, zur Effizienzsteigerung der B2C-Logistik auch diese Interaktionsmöglichkeiten zu nutzen.

Im Rahmen des von der Europäischen Union im IST-Programm geförderten Projekts MOSCA (Decision support system for integrated door-to-door delivery: planning and control in logistic chains, IST 2000-29557) wurde mit MOSCANET ein webbasiertes Portal entwickelt, das als Abstimmungsplattform zwischen Versendern, Logistikdienstleistern und Kunden zur Vereinbarung von Lieferort und -zeitrahmen konzipiert ist. Das in (Cardeneo und Mohr 2003) näher erläuterte System basiert auf der Defi-

tion von Erreichbarkeitsprofilen, einem Regelwerk zur Spezifikation der Gültigkeit alternativer Lieferadressen und -zeitfenster.

Erreichbarkeitsprofile können in ihrer zeitlichen Gültigkeit beschränkt werden oder sich speziell auf einzelne Versender oder Sendungen beziehen. Konzeptionell liegt den in den Profilen angegebenen alternativen Liefermöglichkeiten die Einschätzung zugrunde, daß Endkunden neben der Wohnung zu bestimmten Zeiten auch den Arbeitsplatz, eine Freizeiteinrichtung oder ein Schließfachsystem als Lieferort vorziehen. Zudem kennen die Kunden diese Zeiten und Orte in der Regel im Voraus. Ein Erreichbarkeitsprofil kann sich demnach über mehrere Tage erstrecken und mehrere alternative Liefermöglichkeiten pro Tag anbieten. Abbildung 2.4 zeigt beispielhaft die Darstellung eines Erreichbarkeitsprofils im System von MOSCANET.

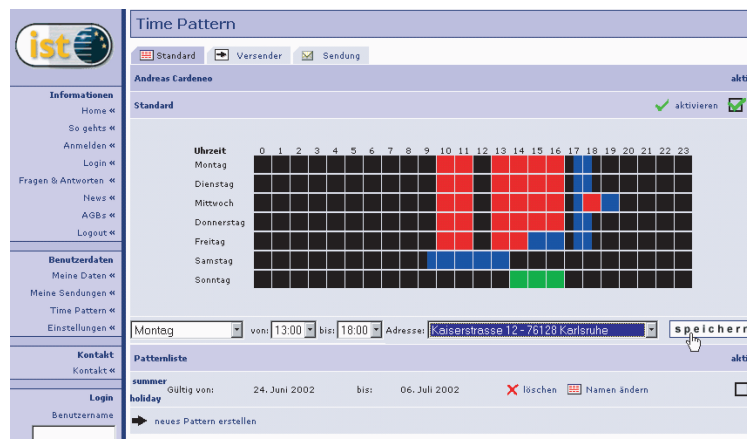


Abbildung 2.3: Eingabe von Erreichbarkeitsprofilen in MOSCANET

Unter Berücksichtigung der Anforderungen an den Datenschutz und die Vertraulichkeit der hinterlegten Informationen können Händler Sendungen über entsprechende Schnittstellen dem Portal avisieren und dem automatisch benachrichtigten Endkunden somit die Möglichkeit geben, angepaßte Liefervorgaben zu definieren. Auf diese Weise können Endkunden bei Abwesenheitszeiten (Urlaub, Dienstreisen) oder speziellen Warengruppen (Möbel, sperrige Artikel) noch rechtzeitig reagieren. Sobald die Waren durch den Händler lieferbar sind, wird dem Logistikdienstleister für die Organisation der Zustellung ein Verweis auf die Kundeninformation im Portal übergeben. Die für die Planung der Lieferung relevanten Informationen – im wesentlichen Anschrift und Zeitfenster – können vom Dienstleister abgerufen werden. Das System findet anhand verschiedener Merkmale das zutreffende Erreichbarkeitsprofil heraus und gibt dem Dienstleister eine Liste alternativer Liefermöglichkeiten zurück. Ausgestattet mit einem entsprechenden Planungssystem wird die Auslieferung unter Berücksichtigung der Alternativen und weiterer Restriktionen geplant. Nach Fertigstellung der Planung erhält der Kunde über das Portal vom Dienstleister ein Zustellungsavis. Abbildung 2.4 gibt einen Überblick über das Konzept von MOSCANET.

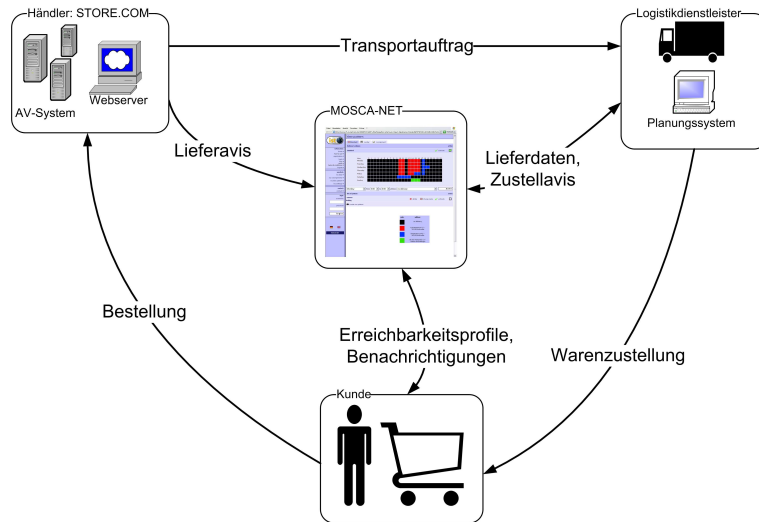


Abbildung 2.4: Akteure und Zusammenspiel in MOSCANET

Der *Personal Logistics Assistant* (PLA) ist ein sich gegenwärtig in der Konzeptionsphase befindliches erweitertes System (s. (Cardeneo 2004)), das neben einer Lauffähigkeit auf einer Vielzahl von Endgeräten (bspw. Mobiltelefonen, PDAs, PCs, digitalen TV-Geräten mit Rückkanal) insbesondere die agentengestützte Verhandlung von Zustelloptionen zwischen Logistikdienstleister und Endkunden vorsieht. Letztere erlaubt es dem Dienstleister, seinerseits Zustelloptionen vorzuschlagen, um auf diese Weise bspw. die Hinterlegung der Sendung in einem Schließfachsystem gegen Rabatt oder Zugaben⁹ zu erreichen. Gegenüber dem MOSCANET-System verfügt der Kunde über stark erweiterte Möglichkeiten der Beschreibung von Zustelloptionen, die auch Bedingungen für die Akzeptanz von dienstleisterseitigen Alternativvorschlägen umfassen können und vom PLA-Backendsystem auf elektronischem Weg mit dem Dienstleister verhandelt werden (vgl. (Ströbel 2003)).

Hinsichtlich des Effekts auf die Logistik ist zu erwarten, daß die Servicequalität als Resultat der Terminvereinbarungen zwischen Kunden und Logistikdienstleister steigt und die Logistikkosten sinken. Auf der einen Seite wird der Kunde die Wahl von Lieferorten und -zeitfenstern als Entgegenkommen durch den Dienstleister werten, auf der anderen Seite sorgt die gestiegene Anwesenheitswahrscheinlichkeit für sinkende Kosten und erhöhte Planungssicherheit beim Dienstleister.

Der Ansatzpunkt des PLA ist ein völlig anderer als der von bereits zuvor beschriebenen Konzepten, da er nicht die Wahl des Zustellortes, sondern die Organisation der Zustellung im Blickpunkt hat. Konzeptionell gibt es keine Einschränkungen für die

⁹Für die Gestaltung eines solchen Rabatt- und Zugabensystems gibt es viele teilweise bereits heute aus dem Handel bekannte Möglichkeiten, bspw. die Gutschrift von Punkten auf Kundenkonten.

Wahl des Zustellortes, vielmehr wird dieser zu einer verhandelbaren Größe. Die Komplexität der Organisation von Bestellungen im eCommerce muß dabei trotzdem nicht zwingenderweise steigen, weil – wie bereits im MOSCANET-System vorgesehen – der Kunde standardmäßige Vorgaben erstellen kann, durch die Sendungen stets gleichartig geleitet werden.

Für eine Darstellung der Implikationen des PLA-Konzepts auf Geschäftsmodelle, Integration in bestehende IT-Landschaften sowie die mögliche technische Gestaltung, die an dieser Stelle nicht gegeben werden soll, wird auf (Cardeneo 2004) verwiesen.

Auf Seiten des Dienstleisters muß für die durchgängige und automatisierte Anwendbarkeit der Möglichkeiten des PLA ein Planungssystem vorhanden sein, daß die vom Kunden gegebenen Optionen auch verarbeiten kann. Auf die Planungssysteme – und die besonderen Herausforderungen an sie durch Konzepte wie den PLA – wird im nächsten Abschnitt eingegangen.

2.5 Tourenplanungsproblem mit alternativen Liefermöglichkeiten

Die für die operative Distributionsplanung eingesetzten entscheidungsunterstützenden Systeme haben nach (Fleischmann und Gietz 2002) die Aufgabe, Transportaufträge nach wirtschaftlichen Kriterien zu Touren zusammenzufassen. In der operativen Tourenplanung (engl. *Vehicle Routing Problem*, VRP) werden üblicherweise die innerhalb eines Planungshorizonts von einem Tag durchzuführenden Abholungen oder Auslieferungen von Waren bei wechselnden Kunden geplant. Die Planung berücksichtigt dabei Daten zu:

- der geographischen Lage der Kundenstandorte,
- den Abmessungen der Sendungen (bspw. Gewicht, Volumen, Anzahl der Packstücke),
- der verfügbaren Fahrzeugflotte (Anzahl und Typ der Fahrzeuge),
- weiteren kunden-, sendungs- oder fahrzeugbezogenen Restriktionen sowie Kombinationen davon

und findet Tourenpläne, deren Güte gemäß anwenderseitig spezifizierten Zielkriterien und Kostenmodellen bewertet wird. In der Praxis findet sich eine Vielzahl verschiedener Restriktionen und Zielkriterien. Zu den verbreitetsten Einschränkungen gehören Beschränkungen der Fahrzeugkapazität und kundenseitige Zeitfenster, durch die Kunden einen gewünschten Liefer- oder Abholzeitraum angeben. In diesem letzten Fall bezeichnet man das Planungsproblem als VRPTW, wobei *TW* für *time windows* steht.

Andere Beispiele sind kundenseitige platzbedingte Größenbeschränkungen für die Fahrzeuge, warenbedingte technische Anforderungen an die Fahrzeuge (temperaturgeführte Transporte) oder zeitliche Einschränkungen in Form von beschränkten Tourdauern, Pausenzeiten oder durch Kommunen festgelegte Zeiten für die Anlieferung in ansonsten verkehrsberuhigten Zonen. Typische Zielkriterien sind die zu minimierende Gesamtstrecke bzw. -dauer aller Touren, die ebenfalls zu minimierende Fahrzeugzahl oder die gleichmäßige Auslastung der Fahrzeuge. Für die Darstellung weiterer Ziele und Nebenbedingungen wird auf die Literatur, bspw. (Fleischmann und Gietz 2002) oder (Toth und Vigo 2002), verwiesen.

Wie bereits dargestellt, setzen KEP-Dienste zumeist auf eine dreigeteilte Transportkette bestehend aus Vor-, Haupt- und Nachlauf. Während die Organisation und Struktur des Hauptlaufs Gegenstand der mittelfristig orientierten Netzwerkplanung sind und Fahrten im Linienverkehr nach einem Fahrplan organisiert stattfinden, liegen im Vor- und Nachlauf bei täglich wechselnden Kunden Tourenplanungsprobleme vor. Für den Bereich der B2C-Logistik relevant sind dabei aufgrund von Retouren sowohl die Sammeltouren des Vorlaufs als auch die Verteiltouren des Nachlaufs für die Zustellung.

In der Tat sind Vor- und Nachlauf im B2B-Sektor wesentlich stabiler, da es oftmals Quellen und Senken mit großem Sendungsaufkommen gibt, die täglich wiederkehrend besucht werden. Ein höherer Anteil von oder gar eine vollständige Beschränkung auf B2C-Transporte erhöht somit durch die stärkere Veränderlichkeit der täglichen Planungsaufgabe den Bedarf nach einem Planungssystem.

Das Tourenplanungsproblem der KEP-Dienste stellt sich wie folgt dar:

- Es wird eine homogene Fahrzeugflotte eingesetzt, weil dies in wirtschaftlicher Hinsicht vorteilhaft ist. Die Vorteile zeigen sich sowohl bei Anschaffung und Unterhalt als auch durch gleiche Abmessungen bei der Be- und Entladung im Betrieb.
- Zeitbezogene Restriktionen haben den größten Stellenwert. Während heute im B2B-Bereich Produkte nach dem spätesten Zustellzeitpunkt differenziert werden, ist nach den in den vorangegangenen Ausführungen erwähnten Studien im B2C-Bereich mit der zunehmenden und von den Kunden ausgehenden Einforderung von Zustellzeitfenstern zu rechnen.
- Weniger kritisch – aber dennoch zu beachten – sind Kapazitätsrestriktionen, die sich im Paketbereich jedoch aufgrund des üblichen Sendungsspektrums nicht auf die Masse oder das Volumen, sondern auf die Anzahl der Packstücke beziehen.

Herkömmliche Planungssysteme (eine Übersicht über Systeme am Markt geben bspw. (Vogt 1998) oder (Hall 2004)) können diese Typen von Restriktionen abbilden und algorithmisch handhaben. Ein anderes Bild ergibt sich, wenn über ein System wie den Personal Logistics Assistant weitere Belieferungsoptionen durch den Kunden an den Logistikdienstleister weitergegeben werden.

In der Konzeption des PLA sind relativ komplexe Möglichkeiten für die Spezifikation von Zustellungen vorgesehen, ein Schlüsselproblem ist jedoch die Berücksichtigung von alternativen Liefermöglichkeiten. Eine Liefermöglichkeit stellt sich dabei als Kombination von Lieferort und -zeitfenster dar, wobei Ort und Zeitfenster im Fall alternativer Liefermöglichkeiten meistens verschieden sein werden. Bietet ein Kunde bspw. zwei alternative Liefermöglichkeiten an, können auf diese Weise

- verschiedene Zeitfenster an einem Ort (bspw. für eine vor- oder nachmittägliche Zustellung in der Wohnung),
- verschiedene Orte zu gleicher Zeit (bspw. für die Zustellung an die Wohnung bzw. den Arbeitsplatz von Lebenspartnern), oder
- verschiedene Orte und disjunkte Zeitfenster (bspw. Zustellung an den Arbeitsplatz tagsüber und abends an die Wohnungsadresse oder ein Schließfachsystem)

beschrieben werden. Die Angabe von alternativen Liefermöglichkeiten vergrößert die Schwierigkeit des Planungsproblems augenscheinlich, da gegenüber der Planungsaufgabe ohne Alternativen noch zusätzlich eine Auswahlentscheidung aus den angebotenen Alternativen getroffen werden muß. Sieht man die Anzahl der in der Planung zu berücksichtigenden Lieferorte als Indikator für die Komplexität der Planungsaufgabe, vergrößert sich diese bei p Kunden mit jeweils zwei zueinander alternativen Liefermöglichkeiten rein mathematisch um den Faktor 2^p .

Transportdienstleister benötigen Planungswerkzeuge, um komplexe Tourenplanungsprobleme wiederholt in hoher und gleichbleibender Qualität lösen zu können. Das Tourenplanungsproblem mit alternativen Liefermöglichkeiten stellt, wie geschildert, eine Schlüsselkomponente im Rahmen von Abstimmungskonzepten und individualisierten Zustelllösungen dar. Um das Potential des Ansatzes auszuloten, muß ein darauf zugeschnittenes Lösungsverfahren entwickelt werden, da dies bislang in der Literatur nicht zu finden ist. Die folgenden Kapitel beschreiben Modelle, Grundlagen, Methoden und das entwickelte Verfahren in seiner Gesamtheit. Die abschließende Bewertung umfaßt einerseits algorithmische und andererseits logistische Aspekte.

3 Methodische Grundlagen

Die Lösung ist immer einfach, man muss sie nur finden
Alexander Solschenizyn

Nach der Einführung in die Problemstellung sollen in diesem Kapitel die notwendigen Grundlagen der in dieser Arbeit verwendeten Methoden dargestellt werden.

Idealerweise ist die Modellierung eines Problems unabhängig von seiner algorithmischen Lösung. Es zeigt sich jedoch gerade für den Bereich der gemischt-ganzzahligen Programmierung, daß die Art der Formulierung starken Einfluß auf die Lösungseffizienz haben kann (s. bspw. (Williams 1999)). Aus diesem Grund wird in dieser Arbeit zunächst die methodische Basis gelegt, auf der in Kapitel 5 für die Modellierung und Kapitel 6 für die konkrete Umsetzung des Verfahrens aufgebaut wird. Zugleich dient die Darstellung der methodischen Grundlagen dem Verständnis der Literaturübersicht im folgenden Kapitel.

Der Schwerpunkt liegt auf den exakten Verfahren Branch-and-Price und Constraint Programming sowie ihrer Kombination in hybriden Ansätzen. Auf (meta)heuristische Lösungsverfahren wird ebenfalls eingegangen, da sie in der Initialphase des entwickelten Verfahrens verwendet werden.

Zunächst jedoch werden gemeinsame Konzepte der exakten Verfahren aufgezeigt. Die Vorstellung erfolgt teilweise im Vorgriff auf die spätere, detailliertere Darstellung und erleichtert eine Einordnung der Verfahren, indem die verbindenden Ideen verdeutlicht werden.

3.1 Grundlegende Konzepte

Die Lösung von kombinatorischen Optimierungsproblemen ist in den wissenschaftlichen Gemeinschaften des Operations Research (OR) und Constraint Programming (CP) in der Vergangenheit unterschiedlich angegangen worden. Während im OR bereits während und nach dem Zweiten Weltkrieg mit der Linearen Programmierung die Grundlage für die spätere Entwicklung der gemischt-ganzzahligen Programmierung (engl. *Mixed Integer Programming* (MIP)) gelegt wurde (siehe (Wolsey 1998)), sind die Entwicklungen im Constraint Programming, das sich in den 1980er und 1990er

Jahren aus der Künstlichen Intelligenz heraus entwickelt hat (siehe (Marriott und Stuckey 1998), (van Hentenryck 1989)), deutlich jünger.

Trotz der unterschiedlichen Herkunft lassen sich nach (Otto 2000) drei wesentliche miteinander verknüpfte Konzepte identifizieren, die beiden Disziplinen zugrunde liegen:

1. Suche,
2. Relaxation und
3. Inferenz.

Unter einer Suche versteht man dabei die systematische Erforschung des Lösungsraumes. Ziel der Suche ist eine Zuordnung von Werten zu Variablen, die sowohl eine gegebene Menge von Restriktionen erfüllt als auch – bei Optimierungsproblemen¹ – eine Zielfunktion minimiert. Ein Suchalgorithmus ist *vollständig* wenn er in endlicher Zeit eine existierende Lösung findet oder anderenfalls das Fehlen einer solchen erkennt. Im Gegensatz dazu findet eine *unvollständige* Suche eine Lösung nicht unbedingt. Insbesondere kann eine unvollständige Suche im Kontext von Optimierungsproblemen terminieren, ohne die Optimallösung gefunden zu haben. Die Suche wird üblicherweise so ausgeführt, daß implizit ein Suchbaum aufgebaut wird, dessen Knoten Teillösungen und dessen Kanten (Äste) Wahlmöglichkeiten entsprechen, durch die der Suchraum aufgeteilt (partitioniert) wird. Vollständige Lösungen entsprechen den Blättern des Suchbaums.

Das *Branch-and-Bound*-Verfahren (siehe (Land und Doig 1960)) ist ein vollständiges Suchverfahren, das sowohl beim CP als auch bei MIP eingesetzt wird. Der Grundgedanke des Branch-and-Bound besteht darin, das Gesamtproblem im Verlauf der Suche schrittweise in Teilprobleme zu zerlegen und möglichst frühzeitig zu erkennen, wann ein Teilproblem von der weiteren Betrachtung ausgeschlossen werden kann. Beim CP wird Branching in Verbindung mit logischen Schlüssen eingesetzt, um die Zahl der Suchknoten zu begrenzen. Bei der Lösung von MIP wird Branch-and-Bound dazu verwendet, unzulässige oder garantiert nicht untersuchenswerte Teile des Suchbaums abzuschneiden. Dies geschieht über eine optimistische Abschätzung des Zielfunktionswertes anhand einer *Relaxation*.

Bei einer Relaxation handelt es sich im wesentlichen um eine vereinfachte Problemformulierung mit einer größeren Lösungsmenge als das Ursprungsproblem. Formal ausgedrückt: Ist P das Ursprungsproblem

$$P := \min_{x \in \mathcal{X}} f(x) \tag{3.1}$$

¹In dieser Arbeit werden stets Minimierungsprobleme angenommen. Dies stellt keine Einschränkung dar, da sich Maximierungsprobleme einfach in Minimierungsprobleme überführen lassen, indem die Zielfunktion mit -1 multipliziert wird.

mit der zulässigen Menge \mathcal{X} und der Zielfunktion $f(x)$ und \tilde{P} ein Problem

$$\tilde{P} := \min_{x \in \tilde{\mathcal{X}}} \tilde{f}(x), \quad (3.2)$$

dann ist \tilde{P} eine Relaxation von P genau dann wenn

1. $\mathcal{X} \subseteq \tilde{\mathcal{X}}$ und
2. $\tilde{f}(x) \leq f(x) \forall x \in \mathcal{X}$

ist. Demnach liefert eine Relaxation eine untere Schranke für das Ursprungsproblem, da die Optimallösung des letzteren nie besser als die der Relaxation sein kann.

Von Wert ist eine Relaxation dann, wenn sie erstens schneller zu lösen ist als das Ursprungsproblem, und sie zweitens dessen Struktur möglichst gut erhält. Der zweite Aspekt ist im Hinblick auf das Branching wichtig, weil dann davon ausgegangen werden kann, daß die untere Schranke gut ist, d.h. daß die Optimallösung des Ursprungsproblems sich „in der Nähe“ der Lösung der Relaxation befindet.

Eine einfache Möglichkeit, für ein MIP eine Relaxation zu erhalten, ist das Streichen von Nebenbedingungen. Eine verbreitete Relaxation besteht darin, die Ganzzahligkeitsforderungen zu vernachlässigen und das so entstandene LP zu lösen. Beim CP hingegen, wird in (Ottosson 2000) berichtet, ist der Einsatz von Relaxationen für die Wertzuweisung an Variablen und damit zur Steuerung der Suche nicht sehr verbreitet, wengleich Relaxationen zum Bounding innerhalb von Constraints eingesetzt werden.

Das dritte grundlegende Konzept der Inferenz, d.h. der logischen Schlußfolgerung, zielt darauf ab, aus einer Menge von Constraints weitere Informationen in Form zusätzlicher Constraints abzuleiten, um damit den Suchraum verkleinern zu können. Beim CP geschieht dies durch den Wertebereiche verkleinernden *Propagationsmechanismus* (s. Abschnitt 3.3.2), bei MIP durch das Abschneiden fraktionaler Lösungen mittels Schnittebenen. Ein weiteres Beispiel für logische Schlußfolgerungen im MIP sind Preprocessing-Techniken und Variablenfixierung aufgrund reduzierter Kosten (siehe bspw. (Martin 1999)). Der wesentliche Unterschied zwischen CP und MIP in diesem Punkt besteht darin, daß die im CP angewandten Techniken die Zulässigkeit und nicht die Optimalität als Ziel haben, d.h. daß zuvorderst unzulässige Variable/Wert-Kombinationen durch Propagation entfernt werden und nicht solche, die an keiner Optimallösung teilnehmen können.

3.2 Branch-and-Price und Column Generation

In diesem Abschnitt werden die methodischen Grundlagen zur Lösung großer gemischt-ganzzahliger Programme, wie sie in dieser Arbeit auftreten, erläutert. Vorausgesetzt wird eine gewisse Vertrautheit mit der Begriffswelt der Linearen Programmierung; ggf. ist (Martin 1999) eine gute Referenz für die Optimierung großer LPs und (Wolsey 1998) eine schöne Einführung in die ganzzahlige Programmierung.

Zentrale Methode ist die Technik der Spaltenerzeugung (engl. *Column Generation*), deren Funktionsweise zunächst in einem Überblick vorgestellt werden soll. Im Anschluß daran wird in den folgenden Abschnitten vertiefend auf die dahinterstehende Theorie der Dekomposition linearer Programme und der Ermittlung ganzzahliger Lösungen eingegangen.

3.2.1 Prinzipskizze des Verfahrens

Zur Erläuterung der Funktionsweise des Column Generation wird aus Gründen der Anschaulichkeit auf das Tourenplanungsproblem mit Kapazitätsrestriktionen zurückgegriffen. Es geht es darum, eine geographisch verteilte Kundengruppe kostenminimal mit Fahrzeugen zu besuchen, wobei die Fahrzeuge nur eine beschränkte Kapazität haben und jeder Kunde nur genau einmal besucht wird. Zwei Fragen stellen sich hierbei: Durch welches Fahrzeug werden die Kunden jeweils besucht und in welcher Reihenfolge geschieht dies? Allgemein kann man auch davon sprechen, einer Menge von Aufgaben (Kundenbesuche) Ressourcen (Fahrzeuge) zuzuweisen mit dem Ziel, ein bestimmtes Kriterium zu optimieren und der Einschränkung, daß verschiedene Restriktionen zu beachten sind.

Die zentrale Idee der Spaltenerzeugung besteht darin, die Lösung des Zuordnungs- und Reihenfolgeproblems auf zwei ineinandergreifende Komponenten zu verteilen: Den *Routengenerator* und die *Tourenzusammenstellung*². Der Routengenerator erzeugt gültige, d.h. den Kapazitätsrestriktionen gehorchende, Routen. Diese werden von der zweiten Komponente zu einem optimalen Tourenplan zusammengestellt, wobei bei der Zusammenstellung berücksichtigt werden muß, daß jeder Kunde genau einmal besucht wird³. Das Ergebnis der Tourenzusammenstellung ist bezüglich der betrachteten Routen optimal, d.h. es wird die bestmögliche Auswahl aus den gegebenen Routen erzeugt.

In jeglichem Optimierungsproblem ist das Optimierungspotential um so größer, je mehr Freiheitsgrade bei der Entscheidung bestehen. Das gesamte Optimierungspotential ließe sich demnach unmittelbar nutzen, wenn der Routengenerator zunächst alle möglichen Routen erzeugte und anschließend durch die Tourenzusammenstellung die gesamtoptimale Kombination von Touren bestimmt würde. Die Bestimmung dieser optimalen Auswahl allein ist allerdings bereits sehr komplex und wird durch die sehr große Anzahl möglicher Routen n_R noch verstärkt. Die Anzahl verschiedener Routen

²Die Begriffe *Route* und *Tour* bezeichnen beide eine Menge von Kunden, die vom selben Fahrzeug besucht werden. Im Unterschied zu einer Tour wird der Begriff der Route dann verwendet, wenn die Abfolge der Kunden im Zusammenhang relevant ist.

³Man geht zumeist davon aus, daß die Dreiecksungleichung für alle Ressourcen erfüllt ist. Damit ist eine Lösung, die einen Kunden mehrmals besucht, durch einfaches Entfernen der Mehrfachbesuche immer in eine Lösung mit geringeren Kosten und ohne Mehrfachbesuche überführbar.

bei n Kunden ergibt sich durch

$$n_R = \sum_{p=1}^n \binom{n}{p} p! \quad (3.3)$$

und wächst mit zunehmender Kundenzahl sehr schnell: Für $n = 1, 2, 3, 4$ beträgt die Zahl der Routen 1, 4, 15 bzw. 64. Für $n = 10$ wächst die Zahl bereits auf 9864100, für $n = 25$ auf ca. $42 \cdot 10^{24}$ (über 42 Quadrillionen).

Eine effizientere Vorgehensweise besteht in der Verzahnung von Routenerzeugung und Tourenzusammenstellung in einem mehrstufigen Verfahren. Zunächst wird eine kleine Menge von Routen durch den Routengenerator berechnet, um anschließend daraus einen bestmöglichen Tourenplan zusammenzustellen. Aus dem Ergebnis der Optimierung lassen sich Informationen ableiten, die für den folgenden Aufruf des Routengenerators benötigt werden. Es handelt sich dabei um Informationen über die Struktur „guter“ Routen, so daß der Routengenerator in den nächsten Schritten nur solche Routen zurückgibt, die dazu beitragen können, das bisherige Ergebnis zu verbessern.

Aus den bereits vorhandenen und den neuen zurückgegebenen Routen erstellt die Komponente zur Tourenzusammenstellung in jedem Schritt einen neuen optimalen Tourenplan. Der Zyklus aus Informationsweitergabe, Routenerzeugung und Tourenplanerstellung wiederholt sich, bis der Routengenerator keine verbessernden Routen mehr erzeugen kann. Der prinzipielle Ablauf des Verfahrens ist nochmals in Abbildung 3.1 dargestellt.

Der Name „Column Generation“ rührt daher, daß die erzeugten Routen in Form von neuen Spalten der Matrix angefügt werden, die von der Komponente zur Tourenzusammenstellung intern zur Speicherung der Touren verwendet wird.

Die Strukturinformation ist offensichtlich das Schlüsselement des Verfahrens. Diese Information beschreibt keine konkreten Routen, etwa über die Angabe von Kundensequenzen. Vielmehr handelt es sich dabei um ein Anreizsystem: Dem Routengenerator werden Anreize dafür gegeben, Lösungen von der Struktur zu erzeugen, die verbessernde Routen aufweisen müßten. Die Information ist allerdings bezüglich der Zulässigkeit nicht vollständig, d.h. Routen von der idealen Struktur sind möglicherweise unzulässig.

Das Konzept der Steuerung einer zuliefernden Komponente durch Informationsweitergabe und Anreizsysteme findet sich in aktuellen betriebswirtschaftlichen (s. (Lafont und Martimort 2001)) und logistischen Fragestellungen, etwa der Steuerung einer Supply Chain (s. (Kaluza, Dullnig und Malle 2003)), wieder. Die Mathematik hat dies interessanterweise bereits vor über 40 Jahren (s. (Dantzig und Wolfe 1961)) entdeckt, und in der Tat gibt es Interpretationen des Column Generation bzw. der dahinter stehenden mathematischen Techniken, die Analogien zu der Steuerung hierarchischer Unternehmensstrukturen aufzeigen (s. (Williams 1999)). Diese mathematischen

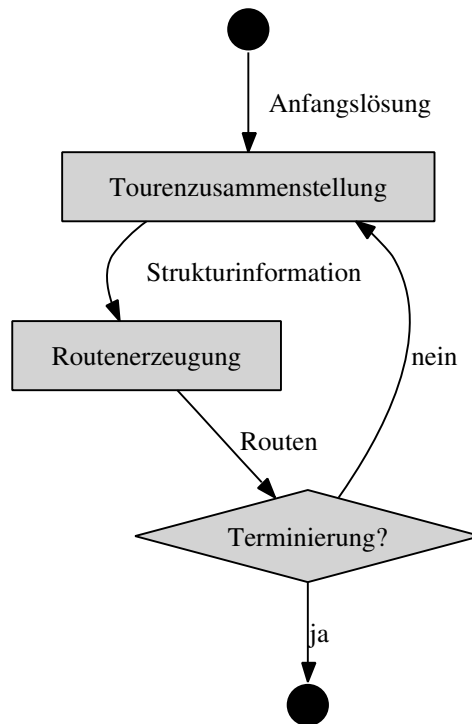


Abbildung 3.1: Prinzipieller Ablauf des Column Generation

Grundlagen werden in den folgenden Abschnitten im Hinblick auf die Anwendung in dieser Arbeit erläutert.

3.2.2 Die Dantzig-Wolfe-Dekomposition

Column Generation wird zur Lösung gut strukturierter gemischt-ganzzahliger Probleme mit einer großen Anzahl von Variablen eingesetzt. Es ist praktisch nicht möglich, diese Probleme durch einen direkten LP-Ansatz zu lösen, sei es, weil die verwendeten Werkzeuge z.B. aus Speicherplatzgründen keine ausreichend hohe Variablenzahl verarbeiten können oder sei es, weil allein die Erzeugung aller Variablen nicht in einem vernünftigen Zeitrahmen geschehen kann. An den Zahlen des obigen Beispiels der Routenerzeugung für ein kleines Tourenplanungsproblem mit $n = 25$ Kunden wurde bereits deutlich, daß die benötigte Datenmenge (Routenverlauf und -kosten) nicht von einem heutigen – und sehr wahrscheinlich auch nicht von einem in absehbarer Zeit zu erwartenden – Rechner bewältigt werden kann.

Die Idee besteht daher darin, mit einer kleinen Zahl von Variablen zu beginnen und dieses sog. *Restricted Master-Problem* (RMP) zu lösen. Ein Subproblem, *Pricing-Problem* genannt, wird zur Prüfung der Optimalität der Lösung des RMP verwendet

und liefert, falls möglich, neue Variablen, die in die Basis aufgenommen werden können.

Die Ursprünge gehen zurück auf die Arbeiten von (Dantzig und Wolfe 1961) und (Gilmore und Gomory 1961). Letztere untersuchten das Zuschnittproblem, bei dem sich das Subproblem als Rucksackproblem darstellt. Neuere Anwendungen umfassen verallgemeinerte Zuordnungsprobleme, Tourenplanungsprobleme und Crew Scheduling-Probleme. Eine Übersicht findet sich bspw. in (Desrosiers, Dumas, Solomon und Soumis 1995) und (Lübbecke 2001) sowie in Kapitel 4 dieser Arbeit.

Die mathematische Grundlage bildet die *Dantzig-Wolfe-Dekomposition* (s. (Dantzig und Wolfe 1961)), bei der ein lineares Programm in ein *Master-* und ein *Pricing-*Problem zerlegt wird. Zur Illustration sei das folgende LP gegeben:

$$\min c^T x \tag{3.4}$$

$$(CF) \quad Ax = b \tag{3.5}$$

$$Bx \leq d \tag{3.6}$$

$$x \geq 0. \tag{3.7}$$

Die Nebenbedingungen sind dabei so angeordnet, daß in (3.5) die „schweren“ und in (3.6) die „leichten“ Restriktionen zusammengefaßt werden. Nach (Nash und Sofer 1996) können Nebenbedingungen als schwer angesehen werden, wenn das lineare Programm ohne sie eine Struktur aufweist, die eine effiziente Lösung erlaubt. Dies kann bspw. eine Netzwerkfluß- oder eine Zuordnungsproblemstruktur sein.

Da beide Sätze von Nebenbedingungen (3.5) und (3.6) eingehalten werden müssen, ist jede zulässige Lösung x Element der Menge

$$\mathcal{X} = \{x | Bx \leq d, x \geq 0\}. \tag{3.8}$$

Zur Vereinfachung der Darstellung wird die Menge \mathcal{X} als beschränkt angenommen, d.h. es handelt sich dabei um ein beschränktes Polyeder, dessen Elemente $x \in \mathcal{X}$ sich nach dem Satz von Minkowski und Weyl (siehe z.B. (Martin 1999)) als Konvexkombination der Ecken (den Extrempunkten) $E(\mathcal{X}) = \{x_1, x_2, \dots, x_k\}$ darstellen lassen⁴. Es gilt daher:

$$x = \sum_{i=1}^k \lambda_i x_i \quad \text{mit} \quad \lambda_1, \lambda_2, \dots, \lambda_k \geq 0, \sum_{i=1}^k \lambda_i = 1. \tag{3.9}$$

⁴Für den Fall eines unbeschränkten Polyeders besagt der Satz von Minkowski und Weyl, daß zusätzlich zu der Konvexkombination der Extrempunkte noch eine nicht-negative Linearkombination der Extrempunkte zur Beschreibung eines Elements $x \in \mathcal{X}$ benötigt wird. Details finden sich z.B. in (Martin 1999), aus Gründen der Prägnanz wird hier von beschränkten Polyedern ausgegangen.

Ersetzt man x durch $\sum_{i=1}^k \lambda_i x_i$ und führt die neuen Bezeichnungen $\tilde{c}_i = c^T x_i$ und $\tilde{p}_i = Ax_i$ ein, kann das Problem (CF) umformuliert werden zu

$$\min \sum_{i=1}^k \tilde{c}_i \lambda_i \quad (3.10)$$

$$(CF^*) \quad \sum_{i=1}^k \tilde{p}_i \lambda_i = b \quad (3.11)$$

$$\sum_{i=1}^k \lambda_i = 1 \quad (3.12)$$

$$\lambda_i \geq 0 \quad \forall i \in \{1, \dots, k\}. \quad (3.13)$$

Es handelt sich demnach weiterhin um ein lineares Problem, allerdings mit den Entscheidungsvariablen λ_i , das weniger Zeilen als das Problem (CF) besitzt, weil die Matrix B nicht mehr explizit auftritt. Allerdings weist es üblicherweise eine sehr viel größere Spaltenzahl (entsprechend der Zahl von Extrempunkten) auf. Die Bedingung (3.12) wird *Konvexitätsbedingung* genannt und stellt die Verbindung zwischen den Variablen x und λ_i entsprechend Gleichung (3.9) dar.

3.2.3 Column Generation

Die Umformulierung allein verspricht jedoch noch keine Verbesserung gegenüber der ursprünglichen Formulierung (CF). Diese tritt erst ein, falls zu Beginn nur mit einer kleinen Anzahl von Spalten gerechnet wird und somit nur die zulässigen Punkte betrachtet werden, die sich als Konvexkombination der aktuellen Teilmenge der Extrempunkte darstellen lassen. Die Optimallösung ist darin nicht notwendigerweise enthalten, weshalb ein Weg gesucht wird zu entscheiden, welche zusätzlichen Extrempunkte in das lineare Programm aufgenommen werden müssen. Die Vorgehensweise dazu ist im Prinzip identisch mit der Vorgehensweise beim Simplex-Algorithmus, bei dem die vergleichbare Entscheidung darin besteht, eine in die Basis aufzunehmende Nicht-Basisvariable zu bestimmen. Die Entscheidung wird anhand der *reduzierten Kosten*

$$\hat{c}_i = \tilde{c}_i - \tilde{p}_i^T u - v \quad (3.14)$$

$$= c^T x_i - u^T Ax_i - v \quad (3.15)$$

$$= (c - A^T u)^T x_i - v \quad \text{mit } x_i \in E(X) \quad (3.16)$$

für die Variable λ_i vorgenommen, wobei hier $u = (u_1, \dots, u_m)^T$ der Vektor der Dualvariablen für die Bedingungen (3.11) und v die Dualvariable der Konvexitätsbedingung (3.12) sind.

Aus der Dualitätstheorie ist bekannt, daß eine Verbesserung⁵ des Zielfunktionswertes nur dann erreicht werden kann, wenn eine Spalte (Variable) mit negativen reduzierten Kosten in die Basis aufgenommen wird. Eine Verringerung des Zielfunktionswertes kann sich deshalb nur dann ergeben, wenn ein Extrempunkt $x_i \in E(X)$ gefunden wird, für den der Term

$$(c - A^T u)^T x_i - v \tag{3.17}$$

den kleinstmöglichen Wert annimmt und negativ ist. Dieser Extrempunkt läßt sich als Lösung des Optimierungsproblems (PP)

$$\begin{aligned} \text{(PP)} \quad & \min (c - A^T u)^T x - v & (3.18) \\ & x \in X. & (3.19) \end{aligned}$$

bestimmen. Bei dem Optimierungsproblem (PP) handelt es sich um das bereits zuvor informell eingeführte Pricing-Problem.

Das Problem der Auswahl einer aufzunehmenden Variable wird demnach auf ein Optimierungsproblem zurückgeführt, bei dem nach Konstruktion bekannt ist, daß die Optimallösung ein Extrempunkt mit minimalen reduzierten Kosten ist. Solange diese Kosten negativ sind, gibt es weitere Variablen, die zu einer Zielfunktionswertverringern von (CF') beitragen können, wenn sie nach einer Reoptimierung von (CF') in die Basis aufgenommen werden. Dadurch werden neue Dualvariablen berechnet, die anschließend für die Lösung des nächsten Pricing-Problems benötigt werden. Das Alternieren zwischen Optimierung des Restricted Master-Problems und Lösung des Pricing-Problems wird dann solange fortgesetzt, bis keine Variablen mit negativen reduzierten Kosten mehr bestimmt werden können. Da es nur eine endliche Zahl von Extrempunkten gibt, ist zudem sichergestellt, daß das Verfahren nach einer endlichen, wenn auch u.U. sehr großen, Zahl von Iterationen terminiert. In der Tat kann es aus theoretischer Sicht erforderlich sein, alle Spalten der Matrix A des Master-Problems zu erzeugen, allerdings spricht die empirische Erfahrung dagegen (s. (Fahle, Junker, Karisch, Kohl, Sellmann und Vaaben 2002)).

Es ist wichtig sich zu vergegenwärtigen, daß Column Generation per se in keiner Weise mit gemischt-ganzzahliger Optimierung verbunden ist. Insbesondere wird die Lösung des Master-Problems nur selten ganzzahlig sein, weshalb die Ganzzahligkeit auf andere Weise sichergestellt werden muß. Zum einen kann die erhaltene Lösung durch Heuristiken in eine ganzzahlige Lösung überführt („repariert“) werden. Dieses Vorgehen bietet sich immer dann an, wenn bekannt ist, daß die Lücke zwischen ganzzahliger Optimallösung und LP-Lösung klein ist. Alternativ kann der Column Generation-Prozess

⁵Im Kontext von Minimierungsproblemen bedeutet dies eine Verringerung

in ein Branch-and-Bound-Schema eingebunden und die optimale ganzzahlige Lösung in einer Baumsuche bestimmt werden.

3.2.4 Column Generation und Modellierung

Als Motivation für Column Generation wurde eingangs die Handhabung von Problemen mit einer großen Anzahl von Variablen gesehen. Allerdings ist als Erkenntnis der vorangegangenen Abschnitte festzuhalten, daß diese *extensive* Formulierung gerade zu einer großen Anzahl von Variablen führt. Der entscheidende Unterschied besteht darin, daß diese Variablen jedoch in Abgrenzung zu einem kompakten Modell dynamisch bei Bedarf generiert werden.

In (Barnhart, Johnson, Nemhauser, Savelsbergh und Vance 1998) wird begründet, weshalb eine Formulierung von (insbesondere gemischt-ganzzahligen) linearen Programmen mit einer großen Zahl von Variablen vorteilhaft sein kann:

1. Eine hierarchische Zerlegung, wie sie sich durch die Dantzig-Wolfe-Dekomposition und Column Generation ergibt, kann sich unmittelbar aus dem realen Problem ergeben und kommt damit dem (menschlichen) Wunsch nach einem verständlichen Modell entgegen,
2. die Relaxation der extensiven Formulierung kann besser als diejenige der kompakten Formulierung sein, d.h. die konvexe Hülle der ganzzahligen Lösungen besser approximieren,
3. eine kompakte Formulierung weist manchmal eine stark symmetrische Struktur auf, die besonders in Kombination mit Branch-and-Bound-Algorithmen zu langsamer Konvergenz führt, weil ein Branching-Schritt das Problem unter Umständen kaum verändert.

In der Konsequenz folgt daraus, daß Modellierung und Lösungsverfahren im Zusammenhang gesehen werden müssen. Dies gilt – tendenziell stärker als im RMP – auch für das Pricing-Problem, das üblicherweise die komplexeren Aspekte des Modells beinhaltet.

3.2.5 Lösung des Pricing-Problems

In den vorangegangenen Abschnitten wurde stillschweigend angenommen, daß das Pricing-Problem per Linearer Programmierung gelöst wird. Dies ist jedoch nicht zwingend erforderlich, vielmehr kann das Subproblem als eine Art *Black Box* betrachtet werden, von der lediglich gefordert wird, daß sie Spalten mit negativen reduzierten Kosten liefert, solange es solche gibt. Die Gestaltung des Pricing-Problems weist einige Freiheitsgrade auf und garantiert damit die Terminierung des Verfahrens unter relativ schwachen Bedingungen.

In der Tat ist es aus theoretischer Sicht völlig ausreichend, lediglich eine neue Spalte (Variable) zu berechnen und an das RMP zurückzuliefern, sofern sie existiert. Andererseits kann die Lösung des Pricing-Problems sehr aufwendig sein und in diesem Fall ist es empfehlenswert, eine Menge von Variablen mit negativen reduzierten Kosten zurückzugeben, wenn dies bei der Berechnung des Pricing-Problems keinen oder nur geringen Zusatzaufwand erfordert.

Demgegenüber steht ein erhöhter Aufwand für die Reoptimierung des RMP bei einer größeren Spaltenzahl, weshalb sich die Entscheidung über die Anzahl der zurückzugebenden Variablen aus der Abwägung beider Aufwände ergibt. Für den Fall der Erzeugung mehrerer Variablen kann die Auswahl der tatsächlich in das RMP aufzunehmenden Spalten ebenfalls nach bestimmten Kriterien erfolgen. In (Irnich 2002) wird auf diese Thematik der *Column Selection* näher eingegangen.

Weiterhin ist es nicht zwingend notwendig, das Pricing-Problem optimal zu lösen. Eine heuristische Lösung oder eine unvollständige Berechnung sind ebenfalls denkbar, solange sichergestellt ist, daß der Lösungsraum alle möglichen Lösungen mit negativen reduzierten Kosten enthält und (zumindest implizit) vollständig durchsucht wird. In (Larsen 1999) werden diesbezüglich verschiedene Strategien untersucht.

3.2.6 Branch-and-Price – Berechnung ganzzahliger Lösungen

In vielen praktischen Fällen ist das vorliegende Problem ganzzahlig, d.h. die kompakte Formulierung ist von der Form

$$\min c^T x \tag{3.20}$$

$$(CFI) \quad Ax = b \tag{3.21}$$

$$Bx \leq d \tag{3.22}$$

$$x \in \mathbb{Z}_+^n. \tag{3.23}$$

und die Dekomposition von der Form

$$\min \sum_{i=1}^k \tilde{c}_i \lambda_i \tag{3.24}$$

$$(CFI') \quad \sum_{i=1}^k \tilde{p}_i \lambda_i = b \tag{3.25}$$

$$\sum_{i=1}^k \lambda_i = 1 \tag{3.26}$$

$$\lambda_i \in \mathbb{Z}_+ \quad \forall i \in \{1, \dots, k\} \tag{3.27}$$

Für die Bestimmung der Dualvariablen und reduzierten Kosten wird im Column Generation die LP-Relaxation gelöst, weshalb die Optimallösung nicht notwendigerweise ganzzahlig ist. Um ganzzahlige Lösungen zu erhalten, kann das Verfahren der Spalten-

erzeugung in einen Branch-and-Bound-Rahmen eingebettet werden. Ein solcher Ansatz wird *Branch-and-Price* genannt und stellt sich etwas komplexer als ein klassischer Branch-and-Bound Ansatz dar.

Branch-and-Bound

Liegt eine nicht ganzzahlige Optimallösung für ein Problem P_0 vor, findet beim Branch-and-Bound eine Aufspaltung in eine Gruppe von Teilproblemen $P_k, k \in \mathcal{K}$ statt. Theoretisch kann diese Aufspaltung eine beliebige Anzahl von Teilproblemen umfassen, solange sichergestellt wird, daß die Vereinigung der Lösungsräume der Teilprobleme $\bigcup_{k \in \mathcal{K}} L(P_k)$ dem Lösungsraum $L(P_0)$ des ursprünglichen Problems entspricht, wobei aus Gründen der Effizienz außerdem noch $L(P_r) \cap L(P_s) = \emptyset$ für $r \neq s$ gelten sollte. Praktisch findet jedoch meist eine Zerlegung in zwei Teilprobleme P_L und P_R statt. Typischerweise erfolgt die Zerlegung anhand einer Ungleichung, durch die eine Teilmenge \mathcal{B} von Variablen in Relation zu einer ganzzahligen Konstante β gesetzt wird, d.h. es wird $\sum_{b \in \mathcal{B}} a_k \lambda_b \leq \beta$ bzw. $\sum_{b \in \mathcal{B}} a_k \lambda_b \geq \beta + 1$ durch eine der Problemformulierung hinzugefügte Nebenbedingung gefordert (wobei die Koeffizienten a_k entsprechend der konkret angewandten Branching-Regel bestimmte Koeffizienten sind). Durch die Aufspaltung von Problemen mit nicht ganzzahligen Lösungen in mindestens zwei Teilprobleme wird beim Branch-and-Bound ein Suchbaum aufgebaut, in dessen Knoten LP-Relaxationen gelöst werden. Jede optimale Lösung einer solchen Relaxation bildet eine untere Schranke für alle weiteren Knoten im zugehörigen Teilbaum. Bei diesen handelt es sich infolge der hinzugefügten Nebenbedingungen jeweils um Verschärfungen der übergeordneten Knoten; die Zielfunktionswerte sind daher nicht kleiner. Andererseits ergibt jede ganzzahlige Lösung eine obere Schranke \bar{z} und die kleinste obere Schranke \bar{z}^* entspricht dem bis dahin besten gefundenen Zielfunktionswert. Wird in einem Knoten eine ganzzahlige Lösung mit Zielfunktionswert $z' < \bar{z}^*$ gefunden, wird die bis dahin beste gefundene Lösung aktualisiert zu $\bar{z}^* = z'$. Die Effizienz des Branch-and-Bound Verfahrens beruht auf der Erkenntnis, daß, falls für den Zielfunktionswert z'' einer fraktionalen Lösung einer LP-Relaxation $z'' > \bar{z}^*$ gilt, die Untersuchung des dazugehörigen Teilbaums abgebrochen werden kann, weil darin keine bessere als die bis dahin beste gefundene Lösung erreicht werden kann. Anschaulich entspricht dieser Vorgang dem Abschneiden eines Astes des Suchbaums, weshalb dieser Schritt auch als *Pruning* bezeichnet wird. Die Argumentation über untere und obere Schranken (*Bounds*) gibt dem Verfahren seinen Namen.

Der Prozess des Branch-and-Bound kann vielfältig gestaltet werden. Im einzelnen wird durch eine Branching-Regel spezifiziert, welche Variablen für das Branching herangezogen werden und in welcher Reihenfolge die Knoten des Suchbaums abzuarbeiten sind. Ein Beispiel für die Auswahl des im nächsten Schritt auszulotenden Knotens ist die Entscheidung zugunsten des Knotens mit dem kleinsten Zielfunktionswert der LP-Relaxation. In (Martin 1999) und (Vanderbeck 1994) finden sich darüberhinaus noch weitere allgemeinere Branching-Techniken, die sich teilweise die spezielle Struktur

der Probleme zu Nutzen machen. Ein Beispiel hierfür ist die Branching-Regel von (Ryan und Foster 1981) für das Set-Partitioning-Problem (siehe auch (Barnhart, Johnson, Nemhauser, Savelsbergh und Vance 1998)).

Branching für das Restricted Master-Problem

Zur ganzzahligen Lösung des RMP kann das Branch-and-Bound-Verfahren allerdings nicht unmittelbar angewandt werden, weil durch das Hinzufügen der Branching-Nebenbedingungen sich auch die Werte der Dualvariablen und damit die reduzierten Kosten ändern. Demzufolge kann es bislang noch nicht in das RMP aufgenommene (und auch noch gar nicht erzeugte) Variablen geben, deren Aufnahme nun vorteilhaft ist, weil sie unter den neuen Bedingungen negative reduzierte Kosten aufweisen.

Eine zweite Schwierigkeit zeigt sich an der Schnittstelle zwischen RMP und Pricing-Problem: Durch das Branching werden bestimmte Variablen bzw. deren Werte im RMP ungültig und es muß vermieden werden, daß im Pricing-Problem neue Variablen erzeugt werden, die ebenfalls ungültig – weil den Branching-Regeln widersprechend – sind. Wie in (Irnich 2002) erläutert, müssen Branching und Pricing zueinander kompatibel sein. Es zeigt sich, daß sich das Branching nicht an den Variablen der RMP-Formulierung, d.h. den Extrempunkten, orientieren sollte. Vielmehr sollten die Variablen der ursprünglichen (kompakten) Formulierung herangezogen werden. Dazu ist es notwendig, von den Werten der Variablen in der (fraktionalen) Lösung des RMP auf die entsprechenden Werte der Variablen in der kompakten Formulierung zurückzuschließen. Am Beispiel eines Routingproblems wird das Vorgehen hierzu deutlich. Routingprobleme – darunter auch das in dieser Arbeit behandelte Tourenplanungsproblem – werden in der extensiven Formulierung häufig als Set-Partitioning-Modell der Form

$$\min \sum_{r \in \mathcal{R}} c_r x_r \tag{3.28}$$

$$(RSP) \quad \sum_{r \in \mathcal{R}} a_{ir} x_r = 1 \quad \forall i \in \mathcal{C} \tag{3.29}$$

$$x_r \in \{0, 1\} \tag{3.30}$$

aufgestellt, wobei \mathcal{C} die Menge der Kunden und \mathcal{R} die Menge der aktuell erzeugten Routen ist. Der Koeffizient $a_{ir} \in \{0, 1\}$ beträgt 1 wenn der Kunde i auf Route r besucht wird und 0 sonst. Die Gleichheitsrestriktionen (3.29) gewährleisten, daß jeder Kunde genau einmal besucht wird. Offensichtlich kann eine Route r in Realität nur genau einmal oder gar nicht durchgeführt werden, die Lösung der LP-Relaxation von (RSP) kann aber einen fraktionalen Wert $x_r = f \in (0, 1)$ liefern. Es würde sich demnach anbieten, auf einen solchen fraktionalen Wert zu verzweigen, d.h. ein Problem P_L mit der zusätzlichen Bedingung $x_r = 0$ und ein Problem P_R mit $x_r = 1$ zu erzeugen. Während P_R in Bezug auf das Pricing unproblematisch ist, resultiert für P_L eine Schwierigkeit: Es muß verhindert werden, daß als Ergebnis eine mit der Route r iden-

tische Route erzeugt wird, weil ansonsten eine mit der Situation vor dem Branching identische Konstellation erzeugt würde und das Verfahren daher nicht fortschreiten kann. Das bedeutet aber, daß das Verfahren zur Lösung des Pricing-Problems eine zusätzliche Nebenbedingung der Form „erzeuge keine Lösung, die identisch zu r ist“ berücksichtigen muß und somit die Problemstruktur sehr wahrscheinlich zerstört wird. Bei Routingproblemen, bei denen das Subproblem meist aus der Lösung eines ressourcenbeschränkten Kürzeste-Wege-Problems besteht, könnten spezialisierte Algorithmen nicht mehr verwendet werden. Die Schwierigkeit verschärft sich noch, wenn (was sehr wahrscheinlich ist) mehrere Branching-Schritte notwendig sind.

Eine Lösung besteht beispielsweise darin (vgl. (Kohl 1995)), sich die Bedeutung der Variablen zu vergegenwärtigen: Der Wert der Variablen x_r kann als Maß dafür angesehen werden, wie sehr die in den Routen $r \in \mathcal{R}$ verwendeten Kanten $\langle i, j \rangle, i, j \in \mathcal{C}$ an der Gesamtlösung teilhaben sollten. Die Reihenfolgeinformation ist zwar nicht unmittelbar von den Routenvektoren abzuleiten, ist jedoch zur Berechnung der Routenkosten c_r im Pricing-Problem ermittelt worden und kann – eine Speicherung dieser Information vorausgesetzt – wiederverwendet werden. Das Branching kann dann darin bestehen, für eine nach bestimmten Regeln ausgewählte Kante $\langle i, j \rangle$ das Auftreten bzw. die Abwesenheit in der Gesamtlösung zu fordern. Dies entspricht tatsächlich den Variablen $x_{ijk} \in \{0, 1\}$ in der kompakten Formulierung, die festlegen, ob der Kunde j unmittelbar nach Kunde i mit dem Fahrzeug k besucht wird.

Während für andere Problemstellungen (s. (Irnich 2002)) unmittelbar die Variablen des RMP für das Branching verwendet werden können, wird am Beispiel des Routingproblems deutlich, daß die Wahl der Branching-Strategie eine ausgesprochene Schnittstellenproblematik ist. Die Betrachtung des Subproblems als „Black Box“ ist nicht mehr möglich, wenn für das Branching mehr Information über die Struktur des Problems bekannt sein muß. Weitere Ausführungen zum Einsatz des Branch-and-Price im Routing finden sich in (Kohl 1995), (Desrosiers, Dumas, Solomon und Soumis 1995), (Larsen 1999) oder auch (Toth und Vigo 2002).

3.2.7 Branch-and-Price-and-Cut

Eine weitere Möglichkeit, ganzzahlige Lösungen zu erhalten, besteht in der Anwendung von Schnittebenenverfahren. Die Idee hinter diesen Verfahren ist die Erzeugung zusätzlicher Nebenbedingungen (Hyperebenen), durch die fraktionale Lösungen unzulässig werden, aber keine ganzzahlige Lösungen verlorengehen. Ähnlich wie Spalten im Column Generation können die Schnittebenen bei Bedarf erzeugt werden. In der Tat entspricht die Spaltenerzeugung dem Hinzufügen von Schnittebenen im dualen Problem und umgekehrt. Das analog zum Pricing-Problem bestehende Problem der Schnittebenenerzeugung wird *Separation-Problem* genannt (s. (Wolsey 1998)).

Es wird zwischen lokal und global gültigen Schnitten unterschieden. Letztere können einer LP-Formulierung der Relaxation im Wurzelknoten des Suchbaums hinzugefügt werden und sind unabhängig von hinzugefügten Branching-Bedingungen im gesam-

ten Branch-and-Bound-Baum gültig. Lokale Schnitte werden in beliebigen Knoten des Suchbaums hinzugefügt und sind für den darunterliegenden Teilbaum gültig. Der Wert der Schnittebenen ist differenziert und im Zusammenspiel mit der Lösungsmethode für die Relaxation zu betrachten, weil die Gefahr besteht, daß eingefügte Schnittebenen eine vorteilhafte Struktur der Nebenbedingung zerstören: Beispielsweise kann die Relaxation aus einem Zuordnungsproblem bestehen, das mittels spezialisierter Algorithmen effizient gelöst wird. Das Hinzufügen von Schnittebenen zerstört diese Struktur und macht die Verwendung des spezialisierten Verfahrens unmöglich, so daß auf andere – weniger effiziente – Berechnungsmethoden ausgewichen werden muß.

In (Barnhart, Johnson, Nemhauser, Savelsbergh und Vance 1998) wird die Kombination von Column Generation und Schnittebenenverfahren empfohlen, weil sie zu sehr guten LP-Relaxationen führt. Das Hinzufügen von Nebenbedingungen bringt aber auch hier – wie im Fall des Branch-and-Bound – Schwierigkeiten mit sich, weil die Dualvariablen der noch nicht erzeugten Schnittebenen für das Pricing schon bekannt sein müssen. Da es sich hierbei jedoch, wie (Irnich 2002) ausführt, um ein nicht triviales Problem handelt, gibt es in der Literatur bis dato nur Arbeiten, bei denen die Separation sich dem Pricing anschließt. Beispielsweise werden in (Barnhart, Hane und Vance 2000) im Anschluß an die Lösung des Pricing-Problems, d.h. sobald keine Variablen mit negativen reduzierten Kosten mehr erzeugt werden können, Schnittebenen generiert und dem Problem hinzugefügt. Die anschließende Reoptimierung des LP erfordert die erneute Lösung des Pricing-Problems unter Berücksichtigung der neuen Dualvariablen für die Schnittebenen. Erst nachdem der Zyklus aus Pricing, Separation und Reoptimierung keine neuen Spalten und Schnitte mit ganzzahliger Lösung hervorbringt, findet ein Branching-Schritt statt.

3.2.8 Herstellung einer Initiallösung

Bei der bisherigen Beschreibung des Column Generation wurde stillschweigend angenommen, daß eine Ausgangslösung vorliegt, die durch das Hinzufügen neuer Spalten sukzessive verbessert werden kann. Diese Voraussetzung muß jedoch zunächst geschaffen werden, indem eine Initiallösung berechnet wird. Diese Initiallösung liefert zudem als Nebeneffekt eine obere Schranke mit.

Eine naheliegende Möglichkeit besteht darin, analog zur Phase I des Simplexverfahrens, über eine künstliche Zielfunktion und künstliche Variablen eine erste zulässige Lösung herzustellen. Die Kosten der künstlichen Variablen sind dabei so hoch gewählt, daß sie (falls eine zulässige Lösung existiert) nicht Bestandteil der Lösung sind.

Von (Vanderbeck 1994) wird vorgeschlagen, stattdessen zur Berechnung der Initiallösung eine problemspezifische Heuristik anzuwenden. Bei *Heuristiken* (aus dem Griechischen „heuristikein“, finden/entdecken) handelt es sich um Verfahren, die in akzeptabler Zeit eine Lösung von ausreichender Güte bestimmen. Nach Lübbecke sollte die Initiallösung eine ähnliche Struktur wie die spätere Optimallösung aufweisen. Insbesondere schlechte Ausgangslösungen, wie sie von schwachen Heuristiken oder sogar

zufälligen Startlösungen erzeugt werden, bewirken nach (Lübbecke 2004), daß das Column Generation zusätzliche Iterationen benötigt, um wieder erfolgversprechende Variablen zu generieren. Der Grund hierfür kann darin gesehen werden, daß schlechte Lösungen durch die Dualvariablen falsche Informationen über die Art der gesuchten Routen an das Subproblem übergeben. Nach (Rousseau 2004) entsprechen die Dualvariablen in der aktuellen Optimallösung den marginalen Besuchskosten eines Kunden. Bei Routen schlechter Qualität sind diese Kosteninformationen daher fehlleitend. Daher empfiehlt sich für die Ermittlung einer Initiaillösung die Bestimmung guter heuristischer Lösungen. Nach Möglichkeit sollten daher neben Eröffnungs- auch Verbesserungsverfahren eingesetzt und ggf. durch eine Metaheuristik gesteuert werden (s. Abschnitt 3.5).

3.3 Constraint Programming

Als noch relativ junge wissenschaftliche Disziplin – zumal im Vergleich mit der Linearen Programmierung – erschließt sich das Constraint Programming (CP), seit seiner Entstehung im Umfeld der Künstlichen Intelligenz, der Logik-Programmierung und der Entwicklung von Programmiersprachen, zunehmend das Feld der Optimierung, insbesondere der Kombinatorischen Optimierung.

In (Freuder 1996) wird CP folgendermaßen beschrieben:

„Constraint programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming: the user states the problem, the computer solves it.“

Auch wenn diese Beschreibung sehr euphorisch erscheint, erfaßt sie dennoch den für CP charakteristischen Aspekt der deklarativen Programmierung: Ein Problem wird beschrieben, ohne den Lösungsweg zu spezifizieren. Dies steht im Gegensatz zur prozeduralen Programmierung, bei der ein Algorithmus für die Lösung angegeben wird. CP bietet über die Deklaration hinausgehend die Möglichkeit, zusätzlich die Suche zu gestalten; daher beschreibt Hooker in (Hooker 2000) CP als die Verschmelzung von deklarativer und prozeduraler Programmierung.

Entwickelt hat sich Constraint Programming aus der Erweiterung der logischen Programmiersprache PROLOG (programmation en logique, s. (Colmerauer 1982)) um komplexere Constraints. Das Ergebnis wurde von Jaffar und Lassez *Constraint Logic Programming* genannt (s. (Jaffar und Lassez 1987)). Da die Behandlung von Constraint-Programmen je nach verwendeten Constraint- und Variablentypen unterschiedlich komplex ist, wurden jeweils spezialisierte Systeme entwickelt. Für kombinatorische Probleme besonders relevant sind *CP(FD)*-Systeme, wobei FD für den endlichen und üblicherweise nur ganzzahlige Werte umfassenden Wertebereich (*fini-*

te domain) der Variablen steht⁶. Weitere Entwicklungen auf dem Gebiet, etwa *Concurrent Constraint Programming* und *Constraint Handling Rules*, sowie ein Überblick über die geschichtliche Entwicklung beschreibt Hooker in dem sehr lesenswerten Band (Hooker 2000).

Neben logischen Programmiersprachen bieten sich auch objektorientierte Sprachen als Basis für die Integration von Constraints an, da sie über geeignete Abstraktionskonzepte verfügen, durch die Constraints gekapselt werden können. Das derzeit bekannteste Beispiel für diesen Ansatz ist die für die Sprachen C++ und Java entwickelte Bibliothek ILOG Solver (s. (ILOG 2001c)), die auch in dieser Arbeit verwendet wurde.

Constraint Programming bietet Möglichkeiten zur Modellierung von kombinatorischen Optimierungsproblemen, die weit über die vergleichsweise eingeschränkten Ausdrucksmöglichkeiten der gemischt-ganzzahligen Programmierung hinausgehen. In Verbindung mit der Integration in moderne Programmiersprachen hat dies zu einer Vielzahl von industriellen Anwendungen geführt. Simonis gibt in (Simonis 2001) eine Übersicht über Anwendungen aus den Bereichen Netzwerkplanung, Produktionsplanung und Scheduling, Transport und Personaleinsatzplanung. Kapitel 4 gibt ebenfalls eine Übersicht über in der Literatur beschriebene Anwendungen des CP auf Routingprobleme.

In den folgenden Abschnitten werden zunächst die grundlegenden Elemente des CP erläutert, um anschließend auf Lösungsmethoden – insbesondere für Optimierungsprobleme – einzugehen.

3.3.1 Grundlegende Konzepte: Variablen, Domains, Constraints

Grundsätzlich kann ein Constraint Programm als ein Problem verstanden werden, bei dem einer Menge von Variablen jeweils ein Wert aus dem zugehörigen Wertebereich so zugewiesen wird, so daß eine Menge von Nebenbedingungen eingehalten wird.

Formal ist ein Constraint Programming Problem definiert als ein Tripel (X, D, C) , wobei $X = \{x_1, \dots, x_n\}$ eine Menge von Variablen und $D = \{D_1, \dots, D_n\}$ eine Menge von endlichen Wertebereichen (Domains) mit $x_i \in D_i$ ist. Die Menge $C = \{c_1, \dots, c_n\}$ ist eine Menge von Constraints, d.h. von Relationen auf jeweils einer Teilmenge $X' \subseteq X$ von Variablen. Gesucht wird eine zulässige Lösung, also eine Zuweisung der Form $x_i = v_i$ für alle $x_i \in X$ mit $v_i \in D_i$, so daß alle Constraints c_i erfüllt sind. Das Tupel $v = (v_1, \dots, v_n)$ wird dann als Lösung des Problems bezeichnet. Ist der Wertebereich auf einen Wert reduziert, ist die entsprechende Variable *instanziiert*.

Nach (Milano 2004) unterscheidet man zwei Arten von Constraints: Mathematische und symbolische Constraints. Mathematische Constraints sind einfache Relationen

⁶Zur Vereinfachung wird in dieser Arbeit stets der Begriff Constraint Programming (CP) verwendet, auch wenn strenggenommen von CP(FD) die Rede sein müßte.

zwischen maximal zwei Variablen⁷, z.B. $x_i = x_j, x_i \leq x_j, x_i \geq x_j, x_i \neq x_j$. Symbolische Constraints, die häufig auch *globale* Constraints genannt werden, sind logische Ausdrücke über mehrere Variablen. Sie drücken üblicherweise vergleichsweise komplexe Sachverhalte aus, und mit ihnen sind spezialisierte *Filteralgorithmen* assoziiert. Diese Algorithmen entfernen aus den Wertebereichen der zugehörigen Variablen diejenigen Werte, die nicht Bestandteil einer zulässigen Lösung sein können.

Häufig lassen sich globale Constraints semantisch durch eine Gruppe einfacher Constraints nachbilden, allerdings ist die Wertebereichsreduktion bei globalen Constraints zuweilen deutlich mächtiger. Ein gutes Beispiel (s. bspw. (Bockmayr und Kasper 1998)) ist die *alldifferent* ($[x_1, x_2, \dots, x_n]$)-Constraint, die erfüllt ist, wenn alle Variablen x_i verschiedene Werte annehmen. Sie kann durch $\frac{n(n-1)}{2}$ einfache Constraints, die paarweise Ungleichheit der x_i sicherstellen, nachgebildet werden. Das folgende Beispiel zeigt aber, daß die globale Constraint Unzulässigkeit besser erkennt.

Beispiel 1 *Das folgende CP sei gegeben:*

- $X = \{x_1, x_2, x_3\}$,
- $D = \{\{0, 1\}, \{0, 1\}, \{0, 1\}\}$ und
- $C = \{x_1 \neq x_2, x_1 \neq x_3, x_2 \neq x_3\}$.

Das Problem ist nicht lösbar, allerdings wird dies nicht bemerkt, wenn die Constraints aus C jeweils einzeln betrachtet werden. Für eine globale alldifferent-Constraint ist jedoch unmittelbar erkennbar, daß bei 3 Variablen und zwei verschiedenen Werten keine zulässige Lösung gefunden werden kann. Ein Algorithmus zur Wertebereichsreduktion der alldifferent-Constraint, der auf der Berechnung eines Matchings in einem bipartiten Graph basiert, wird in (Régin 1994) vorgestellt.

Ein weiterer wichtiger Typ von Constraints sind *Meta-Constraints*, d.h. Constraints über Constraints. Hierbei wird ausgenutzt, daß Constraints entweder erfüllt oder unerfüllt sein können und ihnen somit ein Wahrheitswert zugewiesen werden kann, der seinerseits in logischen Ausdrücken weiterverwendet wird. Das folgende Beispiel zeigt, wie mittels Meta-Constraints aus Wahrheitswerten wieder numerische Werte erzeugt werden können.

Beispiel 2 *Sind $x \geq u$ und $x \leq o$ zwei Constraints, dann besagt die Meta-Constraint*

$$(x \geq u) \wedge (x \leq o) \implies z = 1, \tag{3.31}$$

daß z den Wert 1 annimmt, wenn x im Intervall $[u, o]$ liegt.

⁷Die Forschung beschränkt sich nach (Milano 2004) auf Constraints mit maximal zwei Variablen, wobei sich Constraints mit einer größeren Zahl stets in solche mit nur zwei Variablen umwandeln lassen.

3.3.2 Lösung von Constraint Programmen

Ziel eines Lösungsverfahrens für Constraint Programme ist es, Variablen Werte zuzuweisen und gleichzeitig alle Constraints zu erfüllen. Kernelement jeder Lösungsmethode sind daher Algorithmen zur Eliminierung von Werten aus den Wertebereichen der Variablen, die nicht Bestandteil einer Lösung sein können. Dieses Vorgehen wird *Domain Reduction* genannt, und die Änderung des Wertebereichs einer Variablen wird über den Mechanismus der *Propagation* anderen Constraints angezeigt. Dies bedeutet, daß Constraints, die von der Veränderung des Wertebereichs potentiell betroffen sind, (ggf. erneut) daraufhin geprüft werden, ob die Wertebereiche weiterer Variablen reduziert werden können. Werden infolge der Reduktion alle Werte aus dem Wertebereich einer Variablen entfernt, ist keine Lösung möglich.

Als Auslöser für die Propagation kommen neben der allgemeinen Veränderung des Wertebereichs noch weitere Ereignisse in Betracht, bei denen es sich um Sonderfälle der allgemeinen Wertebereichsveränderung handelt: Die Instanziierung und die Änderung der Wertebereichsgrenzen, d.h. des kleinsten bzw. größten Wertes, den die jeweilige Variable annehmen kann.

Da die Wertebereiche der Variablen endlich sind und bei jedem Durchlaufen des Propagationsprozesses entweder unverändert bleiben oder verkleinert werden, ist der Propagationsprozess endlich. Allerdings endet er nicht notwendigerweise mit der Instanziierung aller Variablen, d.h. der Reduktion der Wertebereiche aller Variablen auf jeweils ein Element. Stattdessen wird ein Zustand erreicht, der als *konsistent* bezeichnet wird.

Es werden verschiedene Grade der Konsistenz unterschieden, die jeweils mit unterschiedlichem Aufwand erreicht werden können. Man unterscheidet üblicherweise *Knoten-*, *Kanten-* und *Pfadkonsistenz* und kann diese zur *k*-Konsistenz verallgemeinern.

Bei der Knotenkonsistenz werden unäre Constraints in einem Vorverarbeitungsschritt betrachtet. Ein Wertebereich D_i für eine Variable x_i ist für eine Constraint c dann knotenkonsistent, wenn c für alle $v \in D_i$ erfüllt ist. Betrachtet man binäre Constraints, kann die Kantenkonsistenz hergestellt werden: Sind x_i und x_j zwei Variablen mit den Wertebereichen D_i bzw. D_j , und ist c eine auf beiden Variablen definierte Constraint, dann sind die Wertebereiche D_i und D_j kantenkonsistent, wenn es zu jedem Wert $v \in D_i$ einen Wert $w \in D_j$ gibt, so daß c erfüllt ist. Die Konsistenz kann dadurch gesichert werden, daß alle Werte $v' \in D_i$ aus dem Wertebereich D_i entfernt werden, für die es kein passendes $w \in D_j$ gibt.

Die Pfadkonsistenz ist eine Verallgemeinerung der Kantenkonsistenz. Zwei Wertebereiche D_i und D_j sind pfadkonsistent, wenn sie auf jedem möglichen Pfad im Constraint Graphen zwischen x_i und x_j kantenkonsistent sind.

Eine weitere Verallgemeinerung der vorangegangenen Konzepte ist die *k*-Konsistenz. Sie ist hergestellt, wenn für jedes konsistente System von $k - 1$ Variablen für jede beliebige *k*-te Variable ein Wert existiert, so daß alle Constraints zwischen diesen *k*

Variablen erfüllt sind.

Die Prüfung der k -Konsistenz ist umso aufwendiger je größer k ist, weshalb man sich für Implementierungen zumeist mit den effizienteren Knoten- und Kantenkonsistenzen begnügt. Da auch die Pfadkonsistenz nicht immer in der Lage ist, die Wertebereiche soweit zu reduzieren, daß eine Lösung angegeben werden kann, erscheint der zusätzliche Aufwand gegenüber den einfacheren Konsistenzkonzepten nicht gerechtfertigt. Stattdessen wird auf eine systematische Suche zurückgegriffen, die Lösungen anhand temporärer Instanzierungen von Variablen konstruiert.

Man kann grundsätzlich *Look Back*- und *Look Ahead*- Suchverfahren unterscheiden. Nach Art des Branch-and-Bound Schemas instanzieren *Look Back* Verfahren Variablen (temporär), so daß die Konsistenz zu bereits durchgeführten Instanzierungen erhalten bleibt. Ist dies nicht möglich, erfolgt ein Rücksprung zu einem vorangegangenen Entscheidungspunkt. Diese Technik ist als *Backtracking* bekannt und in Pseudocodeform in Algorithmus 1 (vgl. (Heipcke 1999)) wiedergegeben.

Algorithmus 1 : Ablauf der Suche für ein CP

```
while keine Lösung and zulässig do
  prüfe/erstelle (lokale) Konsistenz;
  if keine weitere Variableninstanzierung möglich then
    führe Backtracking-Schritt durch;
  else
    wähle eine Variable;
    wähle einen Wert für diese Variable;
  end
end
```

Einfaches (chronologisches) Backtracking weist einige Nachteile auf, die sich daraus ergeben, daß Inkonsistenzen erst sehr spät zum Zeitpunkt ihres Auftretens und nicht bereits vorbeugend erkannt werden können. Zur Lösung dieses Problems sind verschiedene Techniken, vor allem *Backjumping*, *Backmarking* und *Backchecking* (s. (Barták 1999)), entwickelt worden. In Abbildung 3.2 wird anhand des 4-Damen-Problems die Suche per Backtracking veranschaulicht.

Beim Backjumping wird nicht zum vorangegangenen Entscheidungspunkt zurückgesprungen. Vielmehr wird versucht festzustellen, welche vorangegangenen Zuweisungen für den erforderlichen Rücksprung verantwortlich sind und die Suche bei der jüngsten solchen Zuweisung fortgesetzt. Während Backjumping das mehrfache Durchlaufen von zu Konflikten führenden Instanzierungsketten vermeidet, ersparen Backchecking und Backmarking redundante Prüfungen. Backchecking speichert gefundene Inkonsistenzen und vermeidet die erneute Prüfung solange die Voraussetzungen, d.h. die Menge und Werte der temporär instanziierten Variablen, unverändert sind. Beim Backmarking werden zusätzlich Konsistenzprüfungen vermieden, von denen durch

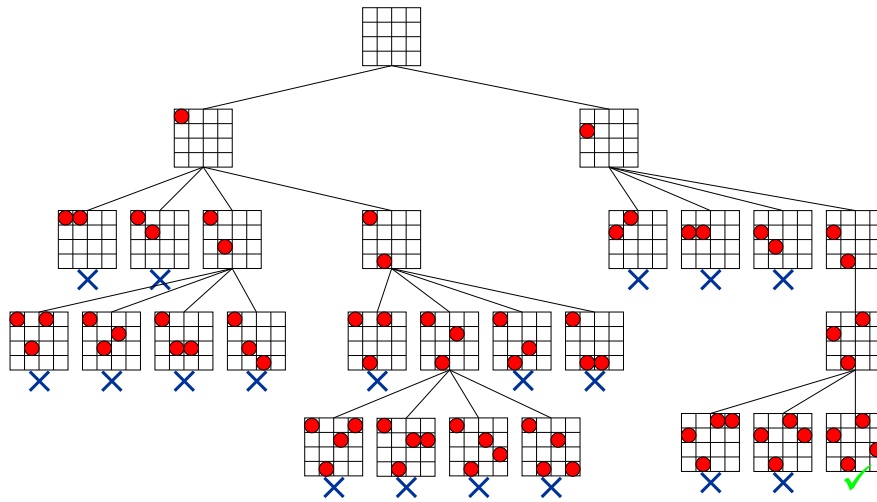


Abbildung 3.2: Suchbaum des Backtracking-Verfahrens zur Lösung des 4-Damen-Problems nach (Barták 1998)

den bisherigen Verlauf der Suche bekannt ist, daß sie erfolgreich sind, d.h. daß sie keine Inkonsistenzen entdecken (vgl. (Barták 1999)).

Ziel der Look Ahead Verfahren ist es, mögliche Inkonsistenzen bereits frühzeitig zu erkennen und Domains von zu Konflikten führenden Werten zu bereinigen. Zu diesem Zweck wird beim *Forward Checking* (vgl. Abbildung 3.3) die Kantenkonsistenz zwischen den im bisherigen Verlauf der Suche noch nicht instanziierten (freien) Variablen und den bereits instanziierten Variablen geprüft. Aus den Wertebereichen der freien Variablen werden diejenigen Werte entfernt, die mit den bereits durchgeführten Zuweisungen inkonsistent sind. Erweitert man diese Prüfungen auf Prüfungen zwischen freien Variablen, gelangt man zum *Partial Look Ahead*.

Eine vergleichende Darstellung der verschiedenen Propagationstechniken zeigt Abbildung 3.4.

Algorithmus 1 läßt den Freiheitsgrad der Variablen- und Wertewahl. Die Gestaltung dieser Auswahlregeln hat großen Einfluß auf die Lösungsgeschwindigkeit (vgl. (Milano 2004)). Der nächste Abschnitt befaßt sich daher mit diesem Aspekt der Suche.

3.3.3 Variableninstanziierung

Für die Bestimmung der Reihenfolge der Variableninstanziierung sind verschiedene Heuristiken entwickelt worden. Dabei kann grundsätzlich zwischen a priori und dynamischen Verfahren unterschieden werden. A priori Verfahren legen die Suchreihenfolge der Variablen vor Beginn der Suche aufgrund bestimmter Eigenschaften fest. Der Vorteil liegt darin, daß nicht in jedem Suchknoten jeweils die nächste zu instanziiere-

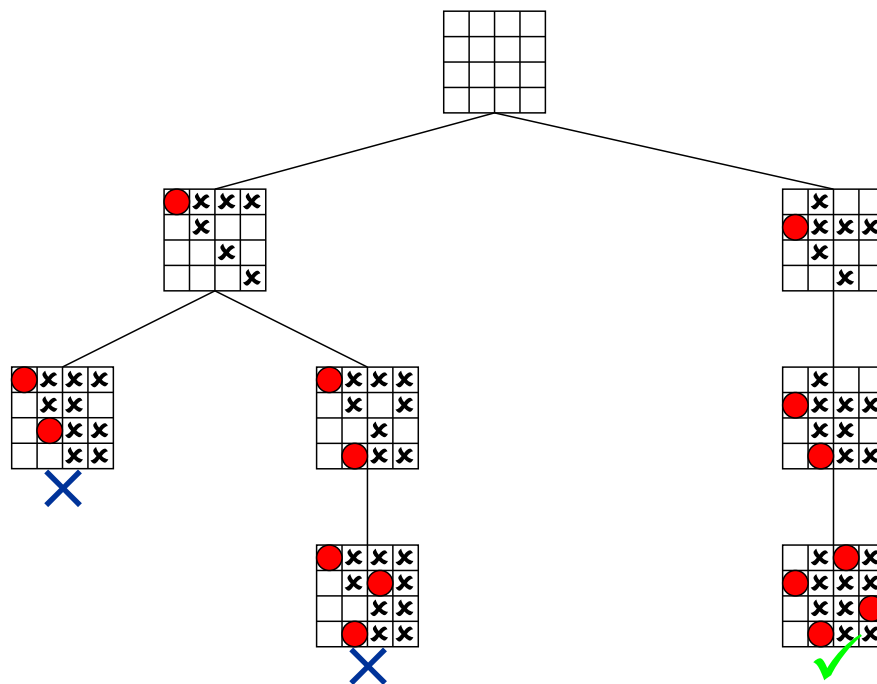


Abbildung 3.3: Suchbaum des Forward-Checking-Verfahrens zur Lösung des 4-Damen-Problems nach (Barták 1998)

de Variable bestimmt werden muß. Dynamische Verfahren passen die Suchreihenfolge im Verlauf des Verfahrens stets entsprechend des Suchfortschritts an.

Eine sehr erfolgreiche dynamische Strategie ist die Verfolgung des *First Fail* Prinzips, bei dem erfolglose Suchpfade so früh wie möglich identifiziert werden sollen. Zu diesem Zweck wird als die zu instanziiierende Variable diejenige gewählt, die von allen Variablen am stärksten beschränkt ist. Ein gutes Maß zur Bestimmung dieser *Constraintness* ist die Anzahl der Elemente im Wertebereich einer Variablen, wobei eine Variable umso stärker beschränkt ist, je weniger Elemente im Wertebereich enthalten sind.

Während bei der Wahl der zu instanziiierenden Variablen die am stärksten beschränkte gewählt wird, wird bei der Wahl des zuzuweisenden Wertes der erfolgversprechendste genommen. Die Idee dahinter ist, daß bei der Variablenwahl im Fall der Unzulässigkeit andere Variablen erst gar nicht untersucht werden müssen, während die Wahl eines erfolgversprechenden Wertes einen Backtracking-Schritt vermeidet.

Neben dieser Instanzierungsstrategie sind ein Vielzahl weiterer Varianten denkbar. Eine Möglichkeit besteht bspw. darin, anstelle einer problemunabhängigen Strategie Wissen über das zu lösende Problem für die Instanzierung zu verwenden.

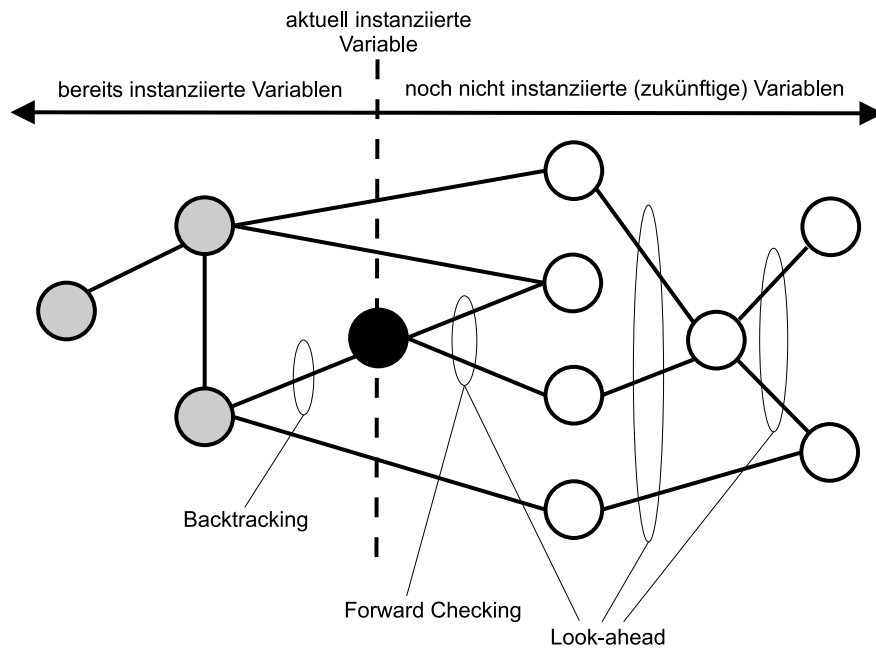


Abbildung 3.4: Vergleich verschiedener Propagationstechniken nach (Barták 1998)

3.3.4 Lösung von Optimierungsproblemen

Ein Constraint-Programm bestimmt gültige Lösungen, die äquivalent zueinander sind. Führt man eine Bewertungsfunktion $f(x)$ für die Lösungen x ein, wird es möglich, die Güte von Lösungen zu vergleichen und die im Sinne der Zielfunktion beste(n) Lösung(en) zu bestimmen.

Die Suche nach einer optimalen Lösung läßt sich iterativ mit den Mitteln des CP lösen: Wann immer eine Lösung x' mit Zielfunktionswert $y = f(x')$ gefunden wird, ergänzt man das CP-Modell um die zusätzliche Constraint $c' : f(x) < y$ (bei Minimierungsproblemen). Auf diese Weise wird ein iteratives Vorgehen festgelegt, bei dem sukzessive Probleme P_1, P_2, \dots, P_n gelöst und um Zielfunktionswertconstraints erweitert werden. Ist das Problem P_{n+1} nicht mehr lösbar, muß die Lösung von P_n die Optimallösung sein.

Im Kontext von Minimierungsproblemen handelt es sich bei dieser Bedingung um eine obere Schranke, d.h. der Wertebereich der Zielfunktion kann von oben her eingegrenzt werden. Allerdings sind im allgemeinen von einer solchen oberen Schranke, bzw. von einer Zielfunktion überhaupt, nur relativ schwache Rückschlüsse auf die betroffenen Variablen möglich und somit ist keine effektive Domainreduktion zu erreichen.

Es ist wünschenswert, über einen Mechanismus zu verfügen, der zu einem Suchknoten angeben kann, ob eine weitere Auslotung erfolgversprechend ist. In der Tat ist dies die Idee des Bounding im Branch-and-Bound, die auch im Kontext des Constraint Pro-

gramming angewandt werden kann. Sobald eine Methode bekannt ist, wie in einem Suchknoten eine untere Schranke des Zielfunktionswertes bestimmbar ist, kann diese wie folgt angewandt werden: Ist der bisher beste Zielfunktionswert \bar{z} und die berechnete untere Schranke \underline{z} , kann auf eine weitere Auslotung des aktuellen Knotens verzichtet werden, wenn

$$\underline{z} > \bar{z} \tag{3.32}$$

ist. Kann man zusätzlich über eine Gradientenfunktion $grad(V, v)$ (zur Notation vgl. (Focacci, Lodi und Milano 1999)) die Veränderung des Zielfunktionswertes bei einer Zuweisung des Wertes v an die Variable x nach unten hin abschätzen, können weiterhin immer dann Werte v aus dem Wertebereich der Variablen x entfernt werden, wenn

$$\underline{z} + grad(x, v) > \bar{z} \tag{3.33}$$

gilt, da diese Zuweisungen niemals zu einem besseren Zielfunktionswert führen.

Eine Möglichkeit, eine solche untere Schranke zu berechnen, besteht darin, eine Relaxation des Problems zu lösen. Vielfach bietet es sich an, als Relaxation ein lineares Programm zu wählen, das zudem über die reduzierten Kosten die gewünschten Gradienteninformationen als Nebenprodukt der Lösung zurückgibt. Zudem ist ein lineares Programm eine etablierte Technik, zu der es leistungsfähige Algorithmen gibt, und die sich für recht allgemeine Problemstellungen eignet. Trotzdem kann es sich als vorteilhaft erweisen, eine Relaxation zu wählen, für deren Lösung es spezialisierte Algorithmen gibt, da bei einem großen Suchbaum des Constraint-Programms der Algorithmus evtl. sehr oft eine Lösung der Relaxation bestimmen muß.

Eine Relaxation und eine damit verbundene Gradientenfunktion ist auch für die Steuerung der Suche einsetzbar. Über die Gradientenfunktion ist das sog. *Regret* (Bedauern) einer Wertzuweisung berechenbar. Das Regret quantifiziert die kleinste Veränderung des Zielfunktionswertes, wenn anstelle der Variablenzuordnung entsprechend der Optimallösung der Relaxation eine andere Wertzuweisung gewählt wird. Die Gradientenfunktion dient hierbei der Abschätzung des Regret. Eingesetzt wird dieses Konzept in der *max-Regret*-Heuristik (s. (Focacci, Lodi und Milano 2004)), die diejenige Variable zur Instanziierung auswählt, die den maximalen Regret aufweist. Nach (Focacci u. a. 2004) liegt die Absicht dieser heuristischen Wahl darin, das Risiko (in Form hoher Kosten) einer unzulässigen Zuweisung des besten Wertes (frühzeitig) zu minimieren.

Eine andere Frage ist die nach der Integration der Bestimmung unterer Schranken in den Suchprozess. Es erscheint relativ geradlinig, hierfür den Propagationsmechanismus zu nutzen und die Bestimmung unterer Schranken in globalen Constraints zu kapseln. Dieses Vorgehen wird von Focacci, Lodi und Milano in (Focacci u. a. 1999) vorgeschlagen. Im Unterschied zu den bekannten globalen Constraints wird bei die-

sen *Optimierungs-Constraints* die Domainreduktion nicht nur im Hinblick auf Zulässigkeit, sondern auch auf Optimalität durchgeführt. Konkret wird dazu eine globale Constraint der Form

$$\text{bounding}([x_1, x_2, \dots, x_n], \underline{z}, \bar{z}) \quad (3.34)$$

eingeführt, die eine Relaxation des Problems bzw. eines Teilproblems kapselt. Dabei bekommt die Constraint Variablen x_1, \dots, x_n sowie untere bzw. obere Zielfunktionswertschranken \underline{z} bzw. \bar{z} geliefert und führt Prüfungen entsprechend (3.32) und – soweit eine Gradientenfunktion zur Verfügung steht – (3.33) durch.

Eine besondere Stärke dieser Form der Integration besteht darin, daß der Propagationsmechanismus voll genutzt werden kann. Wann immer innerhalb der Propagation der *bounding*-Constraint Wertebereiche von Variablen verkleinert werden, wird die Propagation für andere Constraints ausgelöst. Diese kann zu weiteren Domainreduktionen führen und eine Aktualisierung der unteren Schranke auslösen. Andererseits wird aus dieser Beschreibung auch deutlich, daß die Algorithmen zur Berechnung der Relaxation vergleichsweise effizient sein müssen, da sie häufig aufgerufen werden. Ein weiterer Vorteil der Kapselung liegt darin, daß die verwendete Relaxation ausgetauscht werden kann, ohne die Semantik der Constraint und damit das Modell zu verändern.

3.3.5 Gestaltung von CP-Lösungsverfahren

Auch wenn der konzeptionelle Ansatz des CP, wie von Freuder beschrieben, darin besteht, Modellierung und Lösungsverfahren stark zu trennen, zeigt sich bei der praktischen Lösung von Problemen doch die Notwendigkeit, den Suchprozess gezielt zu gestalten.

Eine Maßnahme besteht darin, die Propagation zu verbessern, indem *redundante* Constraints eingeführt werden. Diese verändern das Modell in semantischer Hinsicht nicht, d.h. die Lösungsmenge bleibt unverändert. Die Propagation wird allerdings dadurch gestärkt, daß Wertebereiche häufiger oder stärker reduziert werden können, weil infolge der redundanten Constraints mehr Informationen über unzulässige Lösungen existieren. Die stärkere Reduktion von Domains trägt dazu bei, den Aufwand für die Suche zu verringern und die Lösungsfindung damit zu beschleunigen.

Analog zum Branch-and-Bound bei der Lösung von MIPs, muß während der Suche im Constraint Programming der jeweils nächste auszulotende Knoten des Suchbaums ausgewählt werden. Bekannte Strategien hierfür sind die Tiefensuche, die Bestensuche und die Diskrepanz-Suche (s. (Harvey und Ginsberg 1995)).

Der Gesamtaufwand hinsichtlich Rechenzeit und Speicherplatzbedarf wird von der Anzahl der Suchknoten und dem Aufwand pro Knoten bestimmt. Propagationsintensität und Suchknotenzahl verhalten sich gegenläufig: Eine höhere Propagationsintensität bewirkt eine effektivere Domainreduktion und verringert damit auch den Suchbe-

darf. Demgegenüber steht der erhöhte Aufwand für eine intensivere Propagation, der bspw. durch die Berechnung von Relaxationen oder die Ausführung von komplexeren Filteralgorithmen entsteht. Bei der Gestaltung eines CP-Lösungsverfahrens muß daher anhand der verfügbaren Reduktionstechniken entschieden werden, ob der Schwerpunkt eher auf die Propagation oder alternativ auf die Suche gelegt werden soll. Eine pauschale Feststellung, welche Alternative die effizienteste ist, läßt sich nicht treffen. Vielmehr ist die Wahl abhängig von verschiedenen Parametern, darunter dem Problemtyp, der konkreten Problem Instanz und dem Problemumfang.

3.4 Hybride Verfahren: Constraint Programming-basiertes Column Generation

Nachdem in den vorangegangenen Abschnitten Branch-and-Price bzw. Column Generation und Constraint Programming erläutert wurden, wird in diesem Abschnitt dargestellt, inwiefern sich beide Techniken vorteilhaft miteinander kombinieren lassen.

Wie bereits zuvor dargestellt, sind die Lösung des (Restricted) Master-Problems und des Pricing-Problems bezüglich des Verfahrens unabhängig, d.h. während zur Lösung des RMP Lineare Programmierung eingesetzt wird, kann das Pricing-Problem eine andere Technik verwenden.

Constraint Programming ist besonders zur Lösung von stark beschränkten Problemen geeignet. Die Stärke liegt insbesondere darin, zulässige Lösungen zu finden bzw. festzustellen, daß es keine solchen gibt. Der Zulässigkeitsaspekt ist auch ein wesentlicher Bestandteil des Pricing Problems, da (im Minimierungskontext) Lösungen mit negativen reduzierten Kosten gesucht werden. Unter diesen Voraussetzungen erscheint es folgerichtig, Constraint Programming für das Pricing-Problem einzusetzen (vgl. (Easton, Nemhauser und Trick 2004)).

Im Subproblem finden sich zumeist komplizierte problemabhängige Restriktionen, bei denen Constraint Programming seine Stärken in der Modellierung und der Lösung von Systemen komplexer Nebenbedingungen mittels logischer Implikationen ausspielen kann. Im Unterschied dazu sind die üblicherweise eingesetzten OR-Verfahren zur Lösung typischer Pricing-Probleme wie Rucksack- oder Kürzeste-Wege-Probleme über Jahre entwickelt worden und hochspezialisiert. Es ist – im Gegensatz zum CP – nicht einfach, in diese Algorithmen zusätzliche Nebenbedingungen aufzunehmen, ohne die besondere, für die effiziente Lösung notwendige, Struktur zu zerstören.

Die Flexibilität ist auch vorteilhaft im Branching, weil es wegen der mächtigen Modellierungsmöglichkeiten tendenziell einfacher als mit anderen Methoden ist, die Branching-Regeln für das Pricing-Problem umzusetzen. Die eher schwierige Aufgabe der Abstimmung der Branching-Regeln auf das Zusammenspiel zwischen Restricted Master- und Pricing-Problem wird mit Constraint Programming deutlich vereinfacht.

Die in der Spaltenerzeugung verbreitete Technik, in jedem Pricing-Schritt mehrere

Spalten zu erzeugen, läßt Constraint Programming ebenfalls vorteilhaft erscheinen, weil prinzipiell jede im Verlauf der Suche gefundene Lösung zu einer neuen Spalte werden kann.

Es wird somit deutlich, daß einige Gründe für den Einsatz von CP zur Abbildung und Lösung des Subproblems sprechen. Allerdings liegt im Fall von stark strukturierten Problemen die Lösungseffizienz deutlich unter derjenigen spezialisierter OR-Verfahren. Lassen sich jedoch Teile des Subproblems in einer Weise relaxieren, die zu stark strukturierten Problemen führt, können diese in Form globaler Constraints zusammen mit effizienten Filteralgorithmen (wie dies in Abschnitt 3.3.4 erläutert wurde) in das Subproblem eingebettet werden.

3.5 Heuristiken und Metaheuristiken

Neben Modellierungs- und Berechnungsmethoden für das RMP und das Pricing-Problem sind Verfahren erforderlich, die eine Initillösung bestimmen. In Abschnitt 3.2.8 wurde festgestellt, daß sich dafür Heuristiken empfehlen, weil sie eine gute Lösungsqualität innerhalb kurzer Rechenzeit erreichen. Heuristiken sind zumeist problemspezifisch und nutzen bestimmte Strukturen einer Problemklasse zusammen mit allgemeinen Prinzipien und Suchtechniken (bspw. Tiefensuche, Breitensuche oder Greedy-Suche, vgl. (Cormen, Leiserson und Rivest 1990)) für die algorithmische Bestimmung einer Lösung. Gerade auf dem Gebiet der kombinatorischen Optimierung ist eine Vielzahl von heuristischen Verfahren entwickelt worden. Grund dafür ist, daß die Komplexität vieler dieser Probleme derart hoch ist, daß für einen praktisch relevanten Problemumfang eine beweisbar optimale Lösung nicht in vernünftiger Zeit erzielbar ist. Es handelt sich bei vielen dieser Probleme um \mathcal{NP} -schwere Probleme (s. (Garey und Johnson 1979) für eine Übersicht), von denen derzeit nicht bekannt ist, ob sie überhaupt in polynomialer Zeit gelöst werden können⁸.

Auch das Tourenplanungsproblem fällt in diese Problemklasse, für deren heuristische Problemlösung sich eine zweistufige Vorgehensweise entwickelt hat: Zunächst berechnen Eröffnungsverfahren eine erste zulässige Lösung, anschließend wird mittels Verbesserungsverfahren versucht, den Zielfunktionswert noch weiter zu reduzieren. Es ist ohne weiteres möglich, mehrere Verbesserungsverfahren nacheinander einzusetzen, und tatsächlich ist gerade diese Idee zentral für einige Verfahren.

In den folgenden Abschnitten werden zunächst einige konstruktive Eröffnungsverfahren kurz vorgestellt. Im Anschluß werden lokale Suchverfahren für das VRPTW sowie Grundelemente von *Metaheuristiken*, d.h. Verfahren zur Steuerung von untergeordneten Heuristiken, erläutert. Auf einen der bekanntesten Vertreter dieser Verfahrensklasse, *Tabu Search*, wird genauer eingegangen.

⁸Eine Einführung in die Komplexitätstheorie findet sich bspw. in (Cormen, Leiserson und Rivest 1990)

Die Literatur zu Heuristiken im allgemeinen und zu Heuristiken für die Tourenplanung im besonderen ist sehr umfassend; bereits eine oberflächliche Untersuchung reicht aus, um festzustellen, daß das Design von Heuristiken oftmals von Intuition geleitet ist und vielleicht sogar eher einer Kunst als einer Wissenschaft gleicht. Eine Einführung und Übersicht findet man in (Domschke 1997). Die Arbeit von (Gietz 1994) befaßt sich sehr detailliert mit der Entwicklung und Implementierung von Heuristiken für das Tourenplanungsproblem mit Zeitfenstern, (Vogt 1998) gibt ebenfalls eine gute Übersicht.

3.5.1 Eröffnungsverfahren

Eröffnungs- oder auch Konstruktionsverfahren berechnen auf Grundlage der Problem-
daten eine zulässige Lösung. In den letzten Jahren hat sich die Forschung verstärkt auf die lokalen Suchverfahren konzentriert (s. Abschnitt 3.5.2), weshalb die Eröffnungs-
verfahren überwiegend schon älter sind. Dennoch sind sie weiterhin auch in kommer-
ziellen Planungssystemen im Einsatz (vgl. (Logistik heute 1998; Hall 2004)). Ziel die-
ses Abschnitts ist es, die in dieser Arbeit verwendeten Eröffnungsverfahren vorzu-
stellen und ihre Funktionsweise grob nachzuzeichnen. Dazu gehört auch die kritische
Betrachtung als Motivation für die lokalen Suchverfahren, die im Anschluß erläutert
werden.

Eines der ältesten Eröffnungsverfahren ist das *Savings*-Verfahren aus (Clarke und
Wright 1964). Die Grundidee besteht darin, ausgehend von Pendeltouren, d.h. dem
Besuch nur eines Kunden pro Tour vom Depot aus, die Ersparnis (Saving) bei einem
gemeinsamen Besuch zweier Kunden zu bestimmen und Kunden in der Reihenfolge
dieser a priori berechneten Ersparnisse zu Touren zusammenzustellen. Bezeichnen i, j
zwei Kunden, 0 das Depot und c_{ij} die Kosten einer Fahrt von i nach j , wird das Saving
 s_{ij} durch

$$s_{ij} = c_{0i} + c_{j0} - c_{ij} \quad (3.35)$$

berechnet.

Die Savings s_{ij} können vorab für alle Kundenpaare (i, j) bestimmt und absteigend sor-
tiert gespeichert werden. Zu Beginn des Verfahrens wird jeder Kunde (logisch) durch
eine eigene Tour besucht. Die Liste der Savings wird nun vorwärts durchlaufen und
wann immer die Kunden i', j' des aktuellen Savings $s_{i'j'}$ Randkunden verschiedener
Touren $T(i')$ bzw $T(j')$ sind, können – nach erfolgreicher Prüfung der Restriktionen
– die Touren $T(i')$ und $T(j')$ zu einer neuen Tour verkettet werden⁹. Als Randkun-
den bezeichnet man dabei Kunden, die unmittelbar nach oder vor dem Depot besucht
werden. Abbildung 3.5 illustriert das Prinzip.

⁹Diese Variante des Savings-Algorithmus bezeichnet man als *parallel*. Die *sequentielle* Variante des
Verfahrens erweitert die einzelnen Touren nacheinander um Kunden entsprechend der Reihenfolge
der Savingsliste.

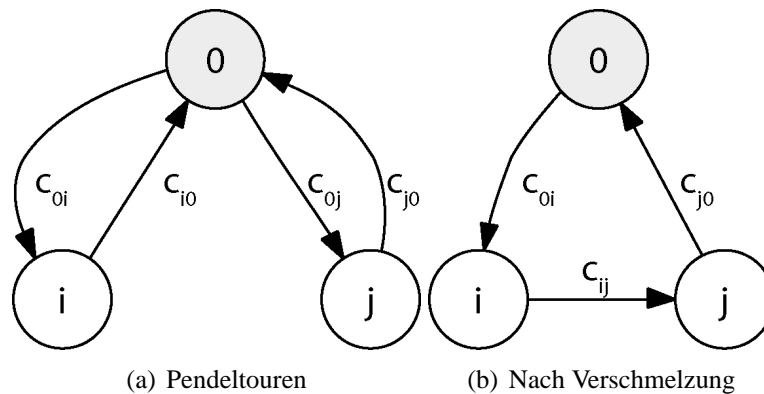


Abbildung 3.5: Prinzip des Savings-Algorithmus

Der größte Vorteil des Savingsverfahrens liegt in der einfachen Implementierbarkeit und der hohen Ausführungsgeschwindigkeit. Die erreichbare Lösungsgüte ist nach (Cordeau, Gendreau, Laporte, Potvin und Semet 2002) allerdings eher schlecht, was darauf zurückgeführt werden kann, daß erstens nur Kosten und keine sonstigen Informationen über die Ressourcen in die Savingsberechnungen eingehen und zweitens für die Tourenverbindung immer nur Savings zwischen Randkunden herangezogen werden.

Der *Sweep*-Algorithmus von Gillett und Miller (s. (Gillett und Miller 1974)) ist ebenfalls ein klassisches Verfahren, das auf Basis der Kundenkoordinaten arbeitet. Es wird angenommen, daß sich das Depot zentral innerhalb einer Menge von n Kunden befindet, deren Polarkoordinaten in einer Liste erfaßt sind. Es werden nun n Tourenpläne ermittelt, wobei in jedem Tourenplan die Touren sequentiell aufgebaut werden. Das Verfahren startet im i -ten Tourenplan beim i -ten Kunden und fügt in der Reihenfolge aufsteigender Polarwinkel Kunden zur aktuellen Tour hinzu, sofern keine Nebenbedingung verletzt wird. Ist dies jedoch der Fall, beginnt der Aufbau der nächsten Tour. Auf diese Weise ermittelt das Verfahren n Tourenpläne, aus denen schließlich der kostenminimale ausgewählt wird. Die Reihenfolge innerhalb einer Tour wird durch ein Reihenfolgeverbesserungsverfahren anschließend noch modifiziert.

Der Algorithmus ist sehr einfach zu implementieren¹⁰, berücksichtigt aber die Entfernung der Kunden vom Depot nicht und nimmt daher in eine Tour sowohl Kunden aus dem Nah- als auch aus dem Fernbereich um das Depot herum auf. Die Clusterung von Kunden anhand der Winkelinformation ist zudem gerade für das VRPTW kritisch, da die zeitliche Gültigkeit von der Reihenfolge abhängt und diese im Verfahren erst in

¹⁰Dies gilt zumindest für moderne Varianten des Verfahrens; die Notation und Beschreibung in der Originalarbeit sind aus heutiger Sicht nahezu unverständlich und der Vergleich mit heutigen Beschreibungen zeigt deutlich, wie sehr die Notation von Algorithmen mit den zur jeweiligen Zeit herrschenden Programmier-techniken zusammenhängt.

zweiter Linie berücksichtigt wird¹¹.

Ein intuitiv naheliegendes Verfahren ist das *Nächster-Nachbar*-Verfahren. Hierbei wird beginnend beim Depot der jeweils nächste (im Sinne der Kostenfunktion $c(i, j)$) Kunde an die aktuelle Tour angefügt, soweit dies zulässig ist. Anderenfalls wird eine neue Tour eröffnet und das Verfahren fortgesetzt, bis alle Kunden in Touren aufgenommen wurden.

Das Verfahren erzeugt im allgemeinen eher schlechte Ergebnisse, weil mit zunehmender Zahl eingeplanter Kunden für die restlichen tendenziell nur noch schlechte, d.h. mit hohen Kosten verbundene, Verknüpfungen möglich sind.

Betrachtet man nicht das Anfügen an existierende Touren, sondern das Einfügen zwischen Kunden, gelangt man zur Gruppe der Einfügeverfahren (engl. *Insertion*). Es werden verschiedene Verfahren unterschieden, je nachdem welche Kriterien zur Auswahl des einzufügenden Kunden und der Einfügeposition verwendet werden. In jedem Fall wird zu Beginn eine Anfangstour erzeugt damit Einfügepositionen bestimmt werden können. Anschließend werden in jeder Iteration ein noch nicht eingeplanter Kunde ausgewählt und eine Einfügeposition bestimmt. Ein konkretes Beispiel ist das *Best-Insertion*-Verfahren, das die kostenminimale Einfügeposition bestimmt, d.h. im Fall des Einfügens des Kunden k zwischen den Kunden i und j werden diese so gewählt, daß der Term

$$c_{ik} + c_{kj} - c_{ij} \tag{3.36}$$

minimal ist.

Die mit Eröffnungsverfahren erreichbare Qualität der Lösungen läßt sich deutlich verbessern. Bis in die 1990er Jahre hinein wurde die Entwicklung von Eröffnungsverfahren in der Forschung vorangetrieben (vgl. (Laporte, Gendreau, Potvin und Semet 1999)). Seit Beginn der 1990er werden lokale Suchverfahren im Rahmen von Metaheuristiken entwickelt, die im folgenden Abschnitt näher beschrieben werden.

3.5.2 Lokale Suche

In diesem Abschnitt sollen grundlegende Konzepte lokaler Suchverfahren – auch Verbesserungsverfahren genannt – aufgezeigt und für Tourenplanungsprobleme typische Vertreter dieser Verfahrensklasse vorgestellt werden.

Lokale Suchverfahren gehen von einer zulässigen Lösung x mit Zielfunktionswert $z(x)$ aus und verändern diese durch Anwendung eines bestimmten Typs von Operation. Die Menge aller möglichen Lösungen x' , die sich durch Anwendung der Operation auf die Lösung x erreichen lassen, wird als *Nachbarschaft* $\mathcal{N}(x)$ von x bezeichnet.

¹¹Aus diesem Grund wird das Sweep-Verfahren auch als *Cluster first, Route second*-Verfahren (Domschke 1997) bezeichnet.

Die Elemente $x' \in \mathcal{N}(x)$ werden als *Nachbarn* mit Zielfunktionswert $z(x')$ bezeichnet und ein Iterationsschritt eines lokalen Suchverfahrens besteht darin, einen Nachbarn $x^* \in \mathcal{N}(x)$ mit $z(x^*) < z(x)$ zu finden und zum Ausgangspunkt des nächsten Iterationsschrittes zu machen. Zwar gibt es auch lokale Suchverfahren, die in einem Schritt eine Verschlechterung des Zielfunktionswertes zulassen, üblicherweise sind allerdings nur Verbesserungen erlaubt, was die Gefahr von lokalen Optima mit sich bringt (siehe (Funke 2003)). Ein lokales Optimum in x aus Sicht des lokalen Suchverfahrens liegt vor, wenn $z(x') > z(x) \forall x' \in \mathcal{N}(x)$ gilt, d.h. der Zielfunktionswert aller Nachbarn größer ist. Wichtig ist in diesem Zusammenhang, daß die lokale Optimalität relativ zur Nachbarschaft festgestellt wird – eine andere Nachbarschaftsstruktur $\hat{\mathcal{N}}$ kann durchaus Nachbarn $\hat{x} \in \hat{\mathcal{N}}(x)$ mit $z(\hat{x}) < z(x)$ enthalten und einen verbessernden Iterationsschritt erlauben. Zur Überwindung lokaler Optima werden Metaheuristiken verwendet, die in Abschnitt 3.5.3 beschrieben werden. Den prinzipiellen Ablauf eines lokalen Suchverfahrens gibt Algorithmus 2 wieder.

Algorithmus 2 : Ablauf des Verfahrens der lokalen Suche ausgehend von x_0

Eingabe : zulässige Lösung x_0

```

setze  $x' = x_0$ ;
setze  $\Delta = \infty$ ;
while  $\Delta > 0$  do
    wähle  $x'' \in \mathcal{N}(x')$  mit  $z(x'') < z(x')$ ;
    if es existiert keine bessere Lösung  $x''$  then
        gib  $x'$  als beste erreichbare Lösung zurück;
    else
         $\Delta = z(x') - z(x'')$ ;
         $x' = x''$ ;
    end
end

```

Für das Tourenplanungsproblem sind eine große Zahl von lokalen Suchverfahren entwickelt worden, die sich in *Inter-* und *Intra-Tour*-Verfahren unterteilen lassen. Erstere modifizieren mehrere Touren gleichzeitig, letztere verändern lediglich die Reihenfolge innerhalb einer Tour.

Aus (Lin 1965) stammt die Familie der k -opt-Verfahren, bei denen in einer Tour k Kanten entfernt und durch k Kanten so ersetzt werden, daß wieder eine Tour entsteht. Die neuen Kanten können zum Teil mit den alten Kanten identisch sein, weshalb die k -opt-Nachbarschaft für $k < k'$ eine Teilmenge der k' -opt-Nachbarschaft ist, d.h. alle Vertauschungen der „kleineren“ Nachbarschaft sind in der „größeren“ Nachbarschaft enthalten. Im Hinblick auf das VRPTW ist es wichtig, festzuhalten, daß der Austausch von Kanten zu zeitlichen Unzulässigigkeiten führen kann, da Teilsequenzen umgekehrt werden können. Das populäre 2-opt-Verfahren entspricht bspw. der Umkehrung der Reihenfolge einer Teilkette der Tour.

Die erreichbare Verbesserung ist um so besser, je mehr Kanten getauscht werden, allerdings verhält sich die Komplexität entsprechend $O(n^k)$, wobei n der Kundenzahl der Tour entspricht. In praktischen Implementierungen beschränkt man sich zumeist auf die 2- oder 3-opt Varianten, um den Aufwand zu begrenzen. In (Funke 2003) wird eine Abschätzung der Nachbarschaftsgröße von k -opt-Verfahren gegeben. In (Savelsbergh 1990) werden effiziente Methoden, die bei Problemen mit Zeitfenstern durch globale Variablen eine Zulässigkeitsprüfung mit konstantem Aufwand realisieren, dargestellt. Abbildung 3.6 illustriert die Funktionsweise des 2-Verfahrens.

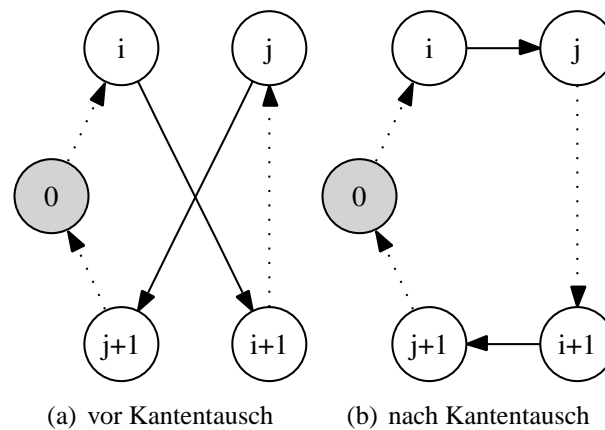


Abbildung 3.6: Funktionsprinzip des 2-opt-Verfahrens nach (Kindervater und Savelsbergh 1997)

Von Or wurde das *Or-opt*-Verfahren entwickelt und in (Or 1976) vorgestellt, das eine vereinfachte Variante des 3-opt ist: Die zugehörige Nachbarschaft umfaßt alle Lösungen, die sich durch Verschieben einer Kette von ein, zwei oder drei Kunden erhalten lassen. Diese Nachbarschaft ist eine Teilmenge der 3-opt-Nachbarschaft, weil sie sich auf maximal dreielementige Ketten beschränkt, während das 3-opt-Verfahren im Gegensatz dazu beliebig lange Ketten verschieben kann. Im Gegenzug beträgt der Aufwand nur $O(n^2)$. Empirische Untersuchungen zeigen sehr gute Resultate, die sich qualitativ mit denen des 3-opt vergleichen lassen. Nach (Funke 2003) ist dies insbesondere für zeitkritische Probleme der Fall, weil hier infolge der umgekehrten Teiltouren ohnehin viele 3-opt-Nachbarn unzulässig sind.

Inter-Tour-Verfahren verändern zugleich mehrere Touren, indem Knoten zwischen Touren verschoben oder ausgetauscht werden. In (Kindervater und Savelsbergh 1997) werden die Verfahren *Exchange*, *Relocate* und *Crossover* beschrieben.

Exchange ist ein Verfahren, das aus zwei Touren je einen Kunden auswählt und einen Austausch durchführt. Hierzu müssen neben den auszutauschenden Kunden auch die neuen Einfügepositionen bestimmt werden. Zuweilen wird das *Exchange*-Verfahren dadurch vereinfacht, daß die Einfügepositionen den vorherigen Positionen entsprechen müssen.

Relocate ist ein recht elementarer Algorithmus, der lediglich einen einzelnen Kunden von einer Tour in eine andere verschiebt. Zu bestimmen ist der aus seiner bisherigen Tour zu entfernende Kunde, die neue Tour, sowie die Position darin.

Crossover – der Begriff wird auch im Zusammenhang mit Genetischen Algorithmen in ähnlicher Bedeutung verwendet – überkreuzt zwei Touren, d.h. tauscht Teilketten von Kunden (jeweils ab einem bestimmten Kunden bis zum Enddepot) zwischen zwei Touren aus. Bestimmt werden müssen dazu jeweils beide Ausgangstouren sowie die Anfangskunden der Teilketten.

Zum Verständnis werden in den Abbildungen 3.7 bis 3.9 alle drei erläuterten Inter-Tour-Methoden veranschaulicht.

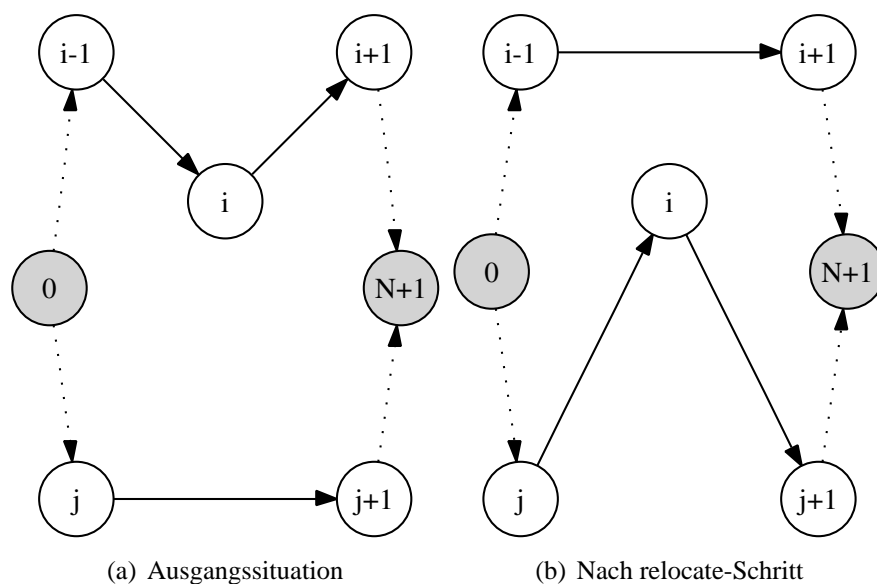


Abbildung 3.7: Funktionsprinzip des relocate-Verfahrens nach (Kindervater und Savelsbergh 1997)

In der Literatur finden sich noch zahlreiche weitere lokale Suchverfahren für Varianten des Tourenplanungsproblems, auf deren Funktionsweise hier jedoch nicht näher eingegangen werden kann. Weiterführende Quellen sind der Band (Toth und Vigo 2002) sowie die Dissertation von Funke.

3.5.3 Metaheuristiken

Obwohl viele lokale Suchverfahren relativ ausgefeilte Techniken und Prinzipien anwenden, darf nicht vergessen werden, daß es sich im wesentlichen um reine Abstiegsverfahren handelt, die einerseits Gefahr laufen, in lokalen Minima steckenzubleiben,

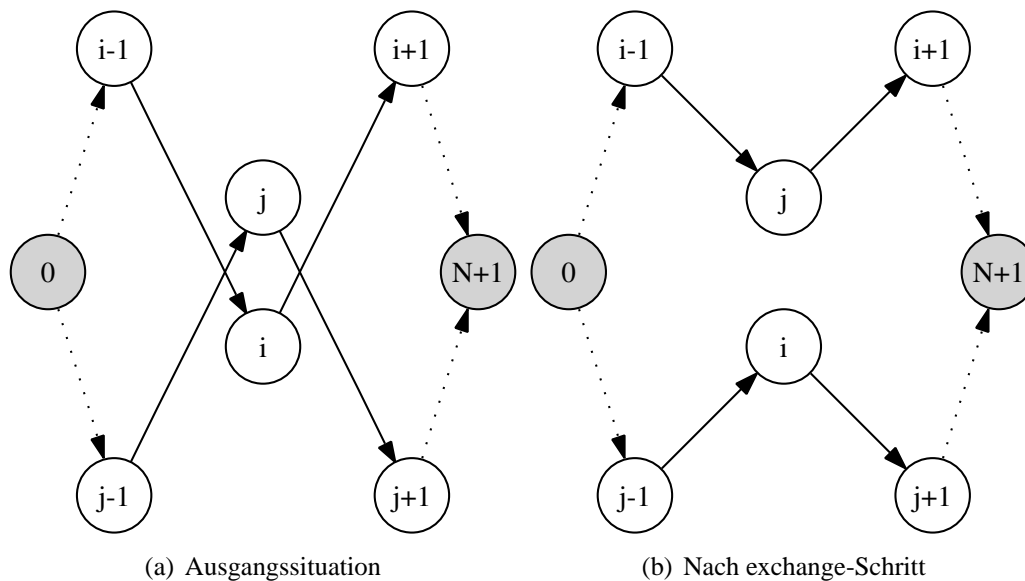


Abbildung 3.8: Funktionsprinzip des exchange-Verfahrens nach (Kindervater und Savelsbergh 1997)

und andererseits nur einen relativ schmalen Ausschnitt aus der gesamten Lösungsmenge untersuchen.

Ein einfacher Ausweg besteht darin, zufällig oder systematisch neue Startpunkte zu bestimmen und das Abstiegsverfahren zu wiederholen. Beschreibt man Metaheuristiken als Verfahren zur Steuerung untergeordneter Heuristiken (wie es bspw. auch (Funke 2003) tut), handelt es sich bereits bei diesem einfachen Ansatz des *Multiple Start* um eine Metaheuristik.

In der Tat ist der Begriff der Metaheuristik in der Literatur nicht einheitlich definiert. Voß beschreibt ihn in (Voß 1999) etwa folgendermaßen:

„A meta-heuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions [...]. A meta-heuristic may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a construction method.“

Eine andere Definition des EU-Projekts *Metaheuristic Network* (Network 1999) ist diese:

„A metaheuristic is a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems. In other

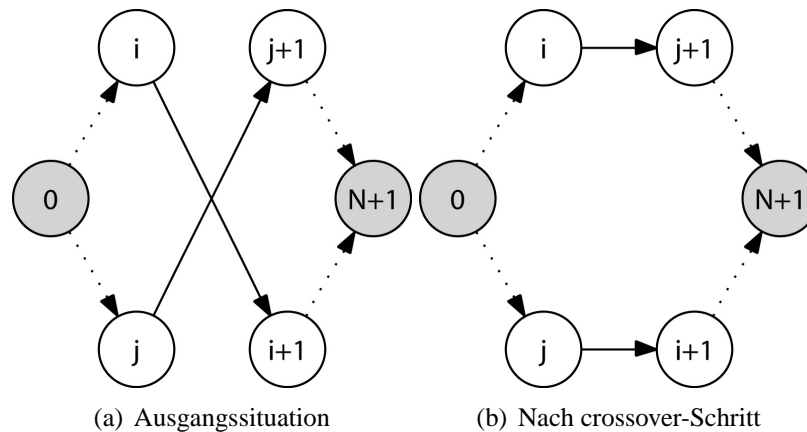


Abbildung 3.9: Funktionsprinzip des crossover-Verfahrens nach (Kindervater und Savelbergh 1997)

words, a metaheuristic can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem.“

Beide Definitionen zeigen, daß sich unter dem Begriff der Metaheuristik viele Verfahren subsumieren lassen und dementsprechend erscheint es sinnvoller, einige häufig anzutreffende Ideen (vgl. (Funke 2003)) vorzustellen, als einzelne Methoden zu erläutern.

Zwei Motive sind entscheidend für die Gestaltung von metaheuristischen Verfahren: *Intensivierung* und *Diversifizierung*. Unter Intensivierung versteht man die genaue Untersuchung von interessanten Bereichen des Lösungsraums. Diversifizierung beschreibt die Suche nach eben diesen interessanten, d.h. vielversprechenden, Regionen des Lösungsraums.

Techniken der Diversifizierung sollen einem Verfahren die Möglichkeit geben, den Lösungsraum möglichst vollständig zu durchsuchen. Strategien der Diversifizierung sind die wiederholte Suche ausgehend von mehreren Startlösungen, sowie der Einsatz von Zufallselementen in der Suche. Die Kombination von Lösungen einer Population oder der Suchhistorie zu neuen Lösungen erlaubt ebenfalls die weiträumige Untersuchung des Lösungsraums.

Komplementär zur Diversifizierung ist die Intensivierung, die die lokale Untersuchung des Lösungsraumes zum Ziel hat. Eine Möglichkeit dazu ist die Verwendung mehrerer Nachbarschaften. Bei der *Variable Neighbourhood Descent* (VND) Metaheuristik (vgl. (Funke 2003)) wird diese Idee umgesetzt, indem eine Menge von Nachbarschaften nacheinander durchsucht werden, bis eine bezüglich aller Nachbarschaften lokal optimale Lösung gefunden wurde. Neben der Anzahl von Nachbarschaften ist eine dynamische Anpassung der Größe der Nachbarschaften eine weitere Intensivierungs-

technik. Insbesondere bei Austauschverfahren bietet es sich an, die Zahl der in einem Verfahrensschritt ausgetauschten Elemente zu verändern. Diese Vorgehensweise liegt der *Variable Depth Search*-Metaheuristik zugrunde; eine erfolgreiche Anwendung auf das TSP findet sich in (Lin und Kernighan 1973).

Die Intensivierung kann jedoch auch bei der Zielfunktion ansetzen, da diese ausschlaggebend für die Akzeptanz oder die Zurückweisung eines Nachbarn ist. Über eine Modifikation dieser, bspw. durch Strafkosten oder das Hinzufügen oder Entfernen von Termen, kann die Suche in ihrer Intensität gesteuert werden. Die Zielfunktion kann bspw. so verändert werden, daß auch bezüglich der ursprünglichen Zielfunktion verschlechternde Schritte akzeptiert werden. Auf diese Weise wird ein Verhalten erreicht, das der bekannten Methode des *Simulated Annealing* nahekommt. Dieses Verfahren von Kirkpatrick u. a. geht auf die Nachbildung von Abkühlprozessen in Kristallen zurück und unterscheidet sich von der lokalen Suche dadurch, daß auch verschlechternde Suchschritte akzeptiert werden. Die Wahrscheinlichkeit dieser Akzeptanz wird dabei anhand von Verfahrensparametern so gestaltet, daß Verschlechterungen im Verfahrensverlauf zunehmend unwahrscheinlicher werden (vgl. (Kirkpatrick, C. D. Gelatt und Vecchi 1983)).

Ein weiteres Element, das sowohl zur Diversifizierung als auch zur Intensivierung dienen kann, ist eine Lern- bzw. Gedächtnisfunktion. Dieses Element wird in der *Tabu Search*-Metaheuristik verwendet, die auch in dieser Arbeit in der Initialphase des hybriden IP/CP-Verfahrens eingesetzt wird.

Tabu Search wurde von Glover entwickelt und in (Glover 1986) und (Glover 1989) vorgestellt. Die Idee von Tabu Search besteht darin, die während des Iterationsfortschritts gesammelten Informationen sinnvoll anzuwenden. Dazu wird eine Art Gedächtnis eingeführt, in dem für eine bestimmte (evtl. variable) Anzahl von Iterationen Merkmale vergangener Iterationen gespeichert werden, die neu erzeugte Lösungen nicht aufweisen sollen. Die gespeicherten Merkmale sind „tabu“, was dem Verfahren seinen Namen gibt. Es gibt weitere Kriterien, sog. *Aspirationskriterien*, die den Tabu-Status von Merkmalen aufheben können. Typische Merkmale in Routing-Problemen sind bestimmte Kundensequenzen oder -zuordnungen und ein verbreitetes Aspirationskriterium ist das Erreichen einer neuen besten Lösung.

Die Nachbarschaft des zugrundeliegenden lokalen Suchverfahrens wird so verändert, daß sie jeweils um die Elemente mit Merkmalen aus der Tabuliste reduziert wird. Nach jedem Schritt wird die beste verbliebene Nachbarlösung zur aktuellen Lösung und die Iterationen werden so lange wiederholt, bis bestimmte Abbruchbedingungen zutreffen. Die Länge der Tabuliste ist ein Parameter des Verfahrens und kann in Varianten des Verfahrens (bspw. dem *Reactive Tabu Search* aus (Battiti und Tecchiolli 1994)) dynamisch angepaßt werden. Kurze Listen führen dazu, daß nur kleine Teile des Lösungsraumes untersucht werden, während lange Listen Nachbarschaften stark einschränken und damit evtl. gute Lösungen ausschließen. Weitere Varianten sowie innerhalb des Verfahrens eingesetzte Techniken werden in (Glover und Laguna 1997) dokumentiert.

Es sind noch zahlreiche weitere Metaheuristiken in der Literatur vorgestellt und analysiert worden. In (Funke 2003) findet sich eine Übersicht, in der von wichtigen Vertretern die charakteristischen Ideen wiedergegeben werden. Daraus wird auch deutlich, daß die Grenzen zwischen den Verfahren diffus sind und sich im Prinzip beliebige Verfahren aus den elementaren Konzepten zusammenstellen lassen. Die in (Voß 1999) gezeigten Anwendungen und Verfahrensentwicklungen geben einen guten Eindruck von der breiten Einsetzbarkeit und den algorithmischen Gestaltungsmöglichkeiten wieder.

4 Aktuelle Arbeiten zu verwandten Problemen – ein Literaturüberblick

Der Fortschritt lebt vom Austausch des Wissens
Albert Einstein

In diesem Kapitel soll eine Übersicht über grundlegende und aktuelle inhaltlich und methodisch verwandte Arbeiten gegeben werden. In methodischer Hinsicht werden Arbeiten vorgestellt, die sich mit der Lösung von Routingproblemen durch Column Generation oder Constraint Programming befassen. Dabei handelt es sich einerseits um Probleme zur Optimierung realer Transportabläufe, und andererseits um solche mit einem stärkeren Modellcharakter, die als Teilprobleme im Rahmen komplexerer Anwendungen auftreten. Beispiele für erstere sind Tourenplanungs- und Netzwerkdesignprobleme, Beispiele für letztere sind Kürzeste-Wege-Probleme innerhalb von Schedulinganwendungen und das Handlungsreisendenproblem als häufig auftretendes Subproblem.

Ein eigenes Unterkapitel ist der Literatur zu hybriden Ansätzen gewidmet, die als Forschungsgebiet in den letzten Jahren viel Aufmerksamkeit auf sich gezogen haben und in den nächsten Jahren interessante Entwicklungen versprechen. Neben Arbeiten, die ähnliche Ansätze wie das in dieser Arbeit vorgestellte hybride Verfahren zeigen, werden auch Arbeiten aus dem thematischen Umfeld präsentiert, um eine qualifizierte Einordnung zu ermöglichen.

Heuristiken sind für die praktische Anwendung von Planungs- und Dispositionssystemen unerlässlich und bieten zudem ein weites Feld für wissenschaftliche Kreativität. In den letzten Jahren sind auch auf diesem Gebiet mit Constraint Programming kombinierte Algorithmen entwickelt und publiziert worden, die in dieser Arbeit ebenfalls überblicksartig dargestellt werden sollen.

Den Abschluß dieses Kapitels bildet eine Bilanz der aus der Literaturanalyse gewonnenen Erkenntnisse.

4.1 Anwendung von Column Generation auf Tourenplanungsprobleme

Desrochers, Desrosiers und Solomon veröffentlichten 1992 den ersten Artikel zur Lösung des VRPTW mit Column Generation (Desrochers, Desrosiers und Solomon 1992). Zur Auswahl zulässiger Routen wird das Masterproblem dort als Set Partitioning-Problem formuliert und das Subproblem als Kürzeste-Wege-Problem mit Kapazitäts- und Zeitfensterrestriktionen modelliert. Zur Lösung des Subproblems wird auf Dynamische Programmierung zurückgegriffen, wobei verschiedene Relaxationen des Zustandsraums vorgestellt werden. Die Relaxationen erlauben dank spezialisierter Algorithmen eine effizientere Lösung des Pricing-Problems, lassen jedoch mehrmalige Besuche eines Kunden auf einer Route zu. Da ein mehrmaliger Besuch eines Kunden den Set Partitioning-Bedingungen des Master-Problems widerspricht, wird für das (Restricted) Master-Problem auf die Set Covering-Formulierung ausgewichen. Aufgrund der zu minimierenden Zielfunktion und bei erfüllter Dreiecksungleichung ist sichergestellt, daß die Optimallösungen beider Formulierungen des Masterproblems gleichwertig sind. Um ganzzahlige Lösungen zu erhalten, wird ein zweistufiges Branching-Schema vorgeschlagen, das in der ersten Stufe eine ganzzahlige Fahrzeuganzahl zum Ziel hat und in der zweiten Stufe auf einzelne Kanten des zugrundeliegenden Netzwerks verzweigt. Die Arbeit wird heute vorwiegend im Zusammenhang mit den dort eingeführten Preprocessing-Schritten für Zeitfenster zitiert. Dabei handelt es sich um einen Satz von vier Regeln zur Eliminierung von Kanten des Netzwerks bzw. zur Reduktion der Zeitfensterbreite. Die Menge der Optimallösungen bleibt auf diese Weise unverändert, während aus dem zugrundeliegenden Netzwerk Kanten entfernt werden, die nicht Bestandteil der Lösung sein können.

Bereits vor Desrochers u. a. lösten (Agarwal, Mathur und Salkin 1989) ein Standard-Tourenplanungsproblem mit Kapazitätsrestriktionen aber ohne Zeitfenster per Column Generation. Während das Masterproblem als Set Partitioning Problem formuliert wird, stellt das Subproblem sich als Rucksackproblem dar. Spezielle Techniken zur Beschränkung der Dualvariablen führen zu verbesserter Konvergenz des Verfahrens.

In (Kohl 1995) wurde die Lösungsmethode von Desrochers u. a. um eine Phase der Erzeugung von Schnittebenen im Wurzel-Knoten des Branch-and-Bound-Baumes erweitert, bei der sog. *2-path*-Schnitte in das Set Partitioning Modell aufgenommen werden. Diese Schnitte identifizieren Knotenmengen, die in der fraktionalen Lösung mit weniger als zwei Fahrzeugen besucht werden, in einer ganzzahligen Lösung aber mit mindestens zwei Fahrzeugen besucht werden müßten. Mit diesem Ansatz ist es in (Kohl 1999) gelungen, die Anzahl der exakt gelösten 100-Kunden-Probleme der Benchmarking-Probleme von Solomon (s. (Solomon 1987)) gegenüber (Desrochers u. a. 1992) zu vergrößern. Außerdem werden verschiedene Methoden auf Basis der Lagrange-Relaxation (s. auch Abschnitt 4.3) entwickelt und verglichen.

Von (Cook und Rich 1999) wird eine Implementierung einer Kombination von Co-

Column Generation und Schnittebenenenerzeugung auf einem Parallelrechnersystem beschrieben. Grundlage der Veröffentlichung ist die Arbeit von (Kohl 1995), die um k -path-Schnitte erweitert wird. Die zu 2-path-Schnitten analoge Idee besteht darin, Schnitte einzuführen, falls in der aktuellen Lösung der LP-Relaxation Kundenmengen enthalten sind, die mit weniger als k Fahrzeugen besucht werden, während die ganzzahlige Lösung mindestens k Fahrzeuge erfordern würde. Schnittebenen werden mit einem randomisierten Verfahren nur im Wurzelknoten des Branch-and-Bound-Baumes erzeugt, wohingegen neue Spalten in allen Knoten berechnet werden. Die Parallelisierung ermöglicht zum einen die Berechnung der Schnittebenen auf verschiedenen Prozessoren, zum anderen die gleichzeitige Abarbeitung der Knoten des Branch-and-Bound-Baumes. Bei der Parallelisierung werden Situationen beobachtet, in denen auf einem Prozessor ein derart aufwendiges Subproblem zur Bestimmung von Schnittebenen gelöst werden muß, daß die anderen Prozessoren warten müssen und damit in der Folge die Effektivität der Parallelisierung sinkt. Im Fall von großen Branch-and-Bound-Bäumen ist die Parallelisierung trotzdem effektiv, da viele Probleme gleichzeitig bearbeitet werden können. Die Autoren sind zwar in der Lage, gegenüber (Kohl 1995) einige zusätzliche Probleme zu lösen, sehen aber noch Bedarf für weitere Arbeiten, da nicht alle Probleme der Solomon-Instanzen optimal gelöst werden können.

Auch in (Larsen 1999) wird eine parallele Berechnungsmethode entwickelt. Zusätzlich werden weitere Verbesserungen im Column Generation Prozess vorgeschlagen, die sich auch bei sequentieller Verarbeitung bewähren. Die von Kohl entwickelten 2-Path-Schnitte zeigen nach Analyse von Larsen einen starken geographischen Bezug, indem sie sich meistens auf Cluster von Kunden beziehen, d.h. Kundenmengen, die relativ nahe beieinander liegen. Es wird daher vorgeschlagen, die Kunden der Schnitte nach dem Nächster-Nachbar-Kriterium auszuwählen. Die dadurch erreichte Beschleunigung der Schnittbestimmung beziffert Larsen auf durchschnittlich 174% für ausgewählte Testprobleme der Solomon-Instanzen.

Im Rahmen des Spaltenmanagements entfernt Larsen nach der Berechnung von jeweils k Branch-and-Bound-Knoten alle Spalten, die bis dahin noch nicht in einer Basis der LP-Relaxation waren. Der Effekt dieser Maßnahme ist in seiner Wirkung nicht offensichtlich: Einerseits reduziert sich bei einer geringeren Spaltenzahl der Aufwand für die Bestimmung der Branching-Variablen und die Lösungszeit der LP-Relaxation, andererseits müssen gelöschte Spalten u. U. erneut generiert werden. Es zeigt sich, daß der Hauptnutzen dieser Technik in der schnelleren Bestimmung der Branching-Variablen liegt und sich erst für größere Werte von k zeigt, wohingegen für kleinere k , d.h. häufigerer Spaltenentfernung, Column Generation ohne Spaltenlöschung im Vergleich schneller ist.

Eine weitere von Larsen implementierte Strategie ist ein vorzeitiger Abbruch des Subproblems, d.h. die Erzeugung von Routen wird gestoppt, bevor die kostenminimale Sequenz gefunden wurde. Dies bewirkt insbesondere eine Reduktion der Berechnungsdauer im Wurzelknoten und verlagert den Aufwand in die Branch-and-Bound-Knoten. Für eine Parallelisierung ist dies ein wünschenswertes Verhalten, die gleiche Tech-

nik zeigt jedoch auch bei einem sequentiellen Ansatz sehr starke Reduktionen in der Rechenzeit. Der Autor führt den Erfolg auf die Beobachtung zurück, daß die innerhalb eines Subproblem-Aufrufs erzeugten Routen von „ähnlicher“ Qualität sind und es demzufolge nicht so sehr darauf ankommt, die beste Route pro Aufruf zu ermitteln. In (Sol 1994) wird ein Algorithmus zur Lösung von Mehrperioden *Pickup & Delivery*-Problemen entwickelt. Probleme dieses Typs stellen eine Erweiterung gegenüber dem Standardproblem dar, bei der für eine Ladung sowohl ein Aufnahme- als auch ein Abgabepunkt bekannt sind und die Ladung ohne zwischenzeitlichen Umschlag an einem Depot vom Ursprung zum Ziel befördert werden muß. Zudem umfassen die Beförderung und Disposition einen Zeitraum von mehr als einem Tag. In der Arbeit wird ein Set Partitioning-Modell verwendet und per Column Generation in einem Branch-and-Price-Ansatz gelöst. Der Autor untersucht verschiedene Varianten der Spaltenerzeugung, u.a. auch die Möglichkeit, Spalten mit negativen reduzierten Kosten zunächst nur approximativ zu bestimmen und erst in dem Fall, daß keine entsprechenden Spalten bestimmt werden, das Pricing-Problem optimal zu lösen. Das Verfahren wurde auf ein reales Planungsproblem einer niederländischen Spedition angewandt und zeigte Verbesserungspotentiale in der Größenordnung von 4-5% gemessen an den Gesamtkosten im Vergleich zu den Plänen der Disponenten. Die Ergebnisse des Algorithmus waren dabei zum Teil trotz höherer Fahrzeuganzahl besser. Sol führt dies darauf zurück, daß den Disponenten bei hoher vorliegender Auftragszahl die Zeit für eine effektivere Verkettung von Transportaufträgen fehlte, weil die Aufträge nach Dringlichkeit sortiert werden mußten und nur die kurzfristigen in die Planung einbezogen werden konnten. Im Gegensatz dazu kann der Algorithmus einen größeren Planungszeitraum berücksichtigen und verfügt somit über ein größeres Optimierungspotential.

4.2 Anwendung des Column Generation auf weitere logistische Problemstellungen

Neben Arbeiten zur Tourenplanung sind in den letzten Jahren einige Veröffentlichungen erschienen, die sich mit der Anwendung von Column Generation auf weitere logistische Probleme befassen.

Löbel betrachtet in seiner Dissertation (Löbel 1997) das Problem der Fahrzeugeinsatzplanung im öffentlichen Personennahverkehr, das sich als Mehrgüterflußmodell formulieren läßt und per Column Generation gelöst wird. Für die Lösung des Pricing-Problems wird eine Pricing-Technik entwickelt, die auf zwei Lagrange-Relaxationen dieses Mehrgüterflußmodells basiert. Mehrere Szenarien aus der Praxis des ÖPNV deutscher Großstädte konnten mit dem entwickelten Verfahren optimal gelöst werden (Lübbecke 2001) beschäftigt sich mit der Einsatzplanung von Rangierlokomotiven bei Industriebahnen. Das Problem besteht darin, Transportaufträge durch verschiedene Lokomotiven durchführen zu lassen und dabei Zeitfenster, Kapazitäts- und Qualifi-

kationsrestriktionen einzuhalten. Ziel ist ein effektiver Einsatz der Lokomotiven mit minimalen Wartezeiten und Leerfahrten. Das Problem wird als Pickup & Delivery-Problem mit Zeitfenstern modelliert, wobei sich die Einsätze aus einer Menge von einfachen Grundmustern zusammensetzen. Diese Grundmuster ergeben sich aus den praktischen Beschränkungen (bspw. der maximalen Zugfähigkeit der Lokomotiven) und der sich daraus ergebenden (geringen) Anzahl möglicher Lade- und Entladefolgen. Aufbauend auf einem sehr umfassenden Literaturüberblick wird ein auf Column Generation basierendes Lösungsverfahren entwickelt, das ganzzahlige Lösungen heuristisch herstellt. Für kleinere und mittlere praktische Probleme mit bis zu 35 Aufträgen bei sechs Lokomotiven werden nach Einschätzung des Autors sehr hohe Lösungsgüten bei akzeptablen Rechenzeiten erreicht.

Probleme des Transportnetzwerk-Designs für den Brieftransport des Nachtluftpostnetzes der Deutschen Post AG werden in (Irnich 2002) untersucht. Das betrachtete zweistufige Transportnetzwerk wird in ein äußeres Netzwerk mit Straßentransporten zwischen Briefzentren und Flughäfen und ein inneres Netzwerk bestehend aus Flügen zwischen Flughäfen zerlegt. Konkret besteht die Planungsaufgabe im Scheduling der Flüge des inneren Netzwerkes, wobei der Autor zwei Fälle unterscheidet. Im ersten Fall finden nur Direktflüge statt, im zweiten Fall können Transportrelationen bei Bedarf über einen Hub geleitet werden. Die Probleme werden als gemischt-ganzzahlige lineare Programme formuliert und per Column Generation mit einem Branch-and-Price-and-Cut-Ansatz gelöst. Während für mittlere Problemgrößen exakte Ergebnisse zu erzielen sind, werden für die Lösung größerer Probleme Heuristiken entwickelt, die im wesentlichen im Preprocessing der Daten für die exakten Verfahren bestehen, um die Problemgröße zu reduzieren. Die exakte Lösung der kleineren Probleme ergibt dann eine heuristische Lösung des Originalproblems.

4.3 Weitere MIP-Ansätze zur Lösung des VRPTW

Neben den in den letzten Jahren untersuchten Column Generation-Ansätzen, sind auch Verfahren auf Basis der Lagrange-Relaxation und Branch-and-Cut-Ansätze in der neueren Literatur beschrieben.

(Bard, Kontoravdis und Yu 2002) entwickeln einen Branch-and-Cut-Ansatz, in dem sie verschiedene Heuristiken zur Berechnung von Schnittebenen und unteren Schranken zu einem exakten Verfahren kombinieren. Die Autoren gehen von einer Zwei-Index-MIP-Formulierung des VRPTW aus, wobei das Ziel der Optimierung die Minimierung der Anzahl benötigter Fahrzeuge ist. Das Vorgehen basiert darauf, durch heuristische Methoden verschiedene Schnitte zu erzeugen und der LP-Relaxation hinzuzufügen, bis keine Schnitte mehr erzeugt werden können. In diesem Fall wird ein Branching-Schritt auf den Zeitvariablen ausgeführt, wenn die Optimallösung noch nicht gefunden wurde. Für die verschiedenen Typen von Schnitten, u.a. Subtour-Eliminierungs-Constraints, Comb-Ungleichungen und paarweise Kundeninkompatibilitäten, werden

teilweise mehrere Heuristiken eingesetzt. Die Ergebnisse sind nach Angaben der Autoren als gleichwertig zu den Ergebnissen von (Kohl 1995) zu sehen.

Ein auf Lagrange-Relaxation basierendes Verfahren wird in (Kohl und Madsen 1997) vorgestellt. Es geht auf die Arbeit (Kohl 1995) zurück und verwendet eine Lagrange-Relaxation gemischt-ganzzahligen Modells des VRPTW. Dadurch wird das (VRPTW) zu einem *ESPPTWCC* (*Elementary Shortest Path Problem with Time Window and Capacity Constraints*, Kürzeste-Wege-Problem mit Zeitfenster- und Kapazitätsrestriktionen und einmaligen Knotenbesuchen) und ist damit identisch zum Subproblem des Column Generation. Zur Lösung des Master-Problems der Lagrange-Relaxation, der Maximierung der Multiplikatoren λ , werden Methoden der nicht-differenzierbaren Optimierung eingesetzt. Kohl und Madsen kommen zu dem Ergebnis, daß das Verfahren mit demjenigen von (Desrochers u. a. 1992) vergleichbar ist.

4.4 Lösung des Handlungsreisendenproblems mittels Constraint Programming

Constraint Programming ist sehr erfolgreich auf dem Gebiet des Scheduling eingesetzt worden (s. (Baptiste, Pape und Nuijten 1995), (Nuijten 1994)) und profitiert dort insbesondere von der Kombination mit OR-Techniken des Scheduling. Das *Traveling Salesman Problem* (TSP) hingegen, das im OR ein typischer (idealisierter) Anwendungsfall für die Bewertung von Algorithmen ist, ist bis Ende der 90er Jahre nur selten mit den Methoden des CP untersucht worden. Andere Routingprobleme, etwa Kürzeste Wege Probleme, sind ebenfalls mittels CP als Subprobleme in hybriden Ansätzen (vgl. Abschnitt 4.5.3) modelliert und gelöst worden. Allerdings sind die wichtigen Modellierungs- und Lösungstechniken sämtlich auch oder sogar speziell für die Lösung des TSP entwickelt worden, weshalb sich an dieser Stelle die Darstellung von CP-Ansätzen zur Lösung von Routingproblemen auf die Betrachtung des Handlungsreisendenproblems beschränkt.

Aufgabe des TSP ist die Bestimmung einer kostenoptimalen Rundreise eines Handlungsreisenden, der eine vorgegebene Anzahl von Orten besuchen muß. Aus graphentheoretischer Sicht besteht das Optimierungsproblem darin, einen kostenoptimalen *Hamiltonschen Zyklus* in einem vollständigen Graphen zu bestimmen. Praktische Probleme der Planung von Werkzeugwechseln, flexiblen Fertigungssystemen oder der Leiterplattenherstellung lassen sich auf das Handlungsreisendenproblem zurückführen.

Das TSP tritt zudem als Teilproblem innerhalb anderer Optimierungsprobleme, insbesondere dem Tourenplanungsproblem, auf. Daraus resultiert der Bedarf für ein effizientes Lösungsverfahren.

Für die Modellierung des Handlungsreisendenproblems mittels Constraint Programming wird als Entscheidungsvariable eine Sequenzvariable S_i eingeführt, die den

Nachfolger des Knotens i angibt. Zur Definition der Zielfunktion $\sum_{i \in V_0} c_{i,S_i}$ (wobei V_0 die Menge der Ausgangsknoten beschreibt) wird von der Möglichkeit variabler Indizes Gebrauch gemacht. Dazu wird die Entscheidungsvariable S_i als Spaltenindex für die Entfernungsmatrix verwendet – ein Konstrukt, das mit Mitteln der Mathematischen Programmierung nicht möglich ist.

Die Modellierung des TSP mittels Nachfolgerbeziehungen als Entscheidungsvariablen und einer `alldifferent`- sowie einer `nocycle`-Constraint ist mit den Modellierungskonstrukten aktueller Werkzeuge einfach durchzuführen. Allerdings sind nach (Caseau und Laburthe 1997) die für die Abbildung von Routing-Problemen in diesen Systemen eingeführten globalen Constraints schon für kleine Probleme aufgrund langer Lösungszeiten kaum einsetzbar.

In ihrem Artikel (Caseau und Laburthe 1997) stellen die Autoren CP-Techniken zur Lösung von kleinen Handlungsreisendenproblemen mit bis zu 30 Knoten vor. Nach Ansicht der Autoren ist diese Beschränkung auf kleine Probleme für die Anwendung auf reale Aufgabenstellungen nicht notwendigerweise nachteilig, da in diesen Fällen bereits andere Restriktionen, bspw. die Fahrzeugkapazität oder Fahrzeugeinsatzzeit, die Problemgröße auf relativ wenige Knoten (Caseau und Laburthe gehen hier von max. 100 Knoten aus) reduzieren.

Im einzelnen werden ein Filteralgorithmus zur Vermeidung von Subtouren und zur Sicherung des (starken) Zusammenhangs, eine Bounding-Constraint und ein Branching-Schema vorgestellt. Im Ergebnis ist es damit möglich, Probleme von doppelter Größe im Vergleich zu einem üblichen CP-Ansatz zu lösen.

Die Vermeidung von Kurzzyklen durch eine globale `nocycle`-Constraint basiert auf Informationen, die zu jedem Knoten gespeichert werden. Konkret werden jeweils Ausgangs- und Endknoten jeder durch den Knoten durchgehenden Knotenkette sowie deren Länge gespeichert. Im Fall einer Variableninstanziierung werden gegebenenfalls die zu einem Kurzzyklus führenden Zuweisungen verboten. Weiterhin können während der Behandlung dieser Constraint Situationen erkannt werden, in denen die Wertebereiche der Nachfolgervariablen bereits derart reduziert worden sind, daß kein zusammenhängender Graph mehr herstellbar ist.

Die untere Schranke der Rundtourlänge wird anhand des Zeilenminimums der Kostenmatrix mit Einträgen $c_{i,j}$ festgestellt, d.h. für jeden Knoten i wird

$$j_i^* = \arg \min_{j \in D(i)} c_{i,j} \quad (4.1)$$

bestimmt und die untere Schranke

$$\underline{z} = \sum_{i=1}^n c_{i,j_i^*} \quad (4.2)$$

berechnet. Da es bei dieser Bestimmung der unteren Schranke zu Mehrfachzuordnungen kommen kann, d.h. $j_l^* = j_k^*$ für $k \neq l$, wird ein Korrekturterm vorgeschlagen, um die untere Schranke \underline{z} zu verbessern. Allerdings wird von den Autoren als weiterführende Technik die Bestimmung einer unteren Schranke mit einer Relaxation (bspw. Minimalen Spannenden Bäumen bzw. Wäldern) empfohlen. Für das Branching wird das Konzept des *Regret* einer Zuweisung eingeführt. Dabei handelt es sich um die Differenz der Kosten zwischen der besten und der zweitbesten Zuweisungsmöglichkeit.

In (Pesant, Gendreau, Potvin und Rousseau 1998) wird Constraint Programming zur Lösung des Traveling Salesman Problems mit Zeitfenstern eingesetzt. Zur Effizienzsteigerung der Suche werden Preprocessing-Techniken zur Zeitfensterreduktion und Kanteneliminierung, wie sie für OR-Ansätze benutzt werden, während der Suche eingesetzt.

Es werden redundante Constraints zur Eliminierung von zeitlich unzulässigen Kanten eingeführt und sowohl vor als auch während der Suche propagiert. Diese zusätzlichen Constraints sind zum Teil auf die Problemstellung des Handlungsreisenden angepaßt, bei dem im Gegensatz zu allgemeineren Routing-Problemen die Menge der zu besuchenden Knoten fest vorgegeben ist. Die aus dem Preprocessing von mathematischen Programmen bekannten Regeln zur Reduktion der Zeitfensterlängen aus (Desrochers, Desrosiers und Solomon 1992) können im CP-Kontext während der Suche zur Beschränkung der Wertebereiche der Entscheidungsvariablen S_i verwendet werden. Die von Pesant, Gendreau, Potvin und Rousseau berichteten Ergebnisse mit einer Implementierung in Prolog zeigen sehr gute Zielfunktionswerte für einen Satz der auf das TSP angepaßten Solomon-Testprobleme. Insbesondere heben die Autoren hervor, daß der infolge von redundanten Constraints und der Schrankenbestimmung erhöhte Aufwand für die Propagation sich für größere Suchbäume lohnt. Diese liegen vor, wenn das Problem viele Freiheitsgrade erlaubt, weil bspw. die Zeitfenster nicht eng sind. Bezüglich des Bounding wird berichtet, daß die Kenntnis über eine untere Schranke nur dann effektiv verwertbar ist, wenn eine (gute) obere Schranke bekannt ist. Im vorliegenden Artikel wurden die oberen Schranken der Literatur entnommen. Falls – was realistischer ist – solche Schranken nicht vorliegen, werden sie im Suchprozess aber u.U. erst relativ spät festgestellt. Dies bedeutet, daß das Bounding zu einem frühen Zeitpunkt – an dem es besonders wertvoll wäre – noch nicht greifen kann.

Dieselben Autoren erweitern in (Pesant, Gendreau, Potvin und Rousseau 1999) das Verfahren um die Fähigkeit zur Behandlung mehrfacher Zeitfenster. Im Vergleich zum bekannten TSPTW ist bei dieser Problemstellung einem Kunden eine Menge von Zeitfenstern zugeordnet, von denen genau eines für einen Besuch ausgewählt wird. Eine andere Interpretation besteht darin, die Öffnungszeiten der Kunden als einen großen Öffnungszeitraum mit mehreren Ausschlußzeiträumen, innerhalb derer kein Besuch stattfinden darf, zu betrachten.

Die Modellbedingungen aus (Pesant u. a. 1998) können nahezu unverändert übernommen und um eine weitere Bedingung erweitert werden, die Besuche innerhalb der Ausschlußzeiträume verbietet. Bezeichnet m_i die Anzahl dieser Zeiträume für einen

Kunden i und das offene Intervall (a_i^k, b_i^k) den k -ten Ausschlußzeitraum, kann die Bedingung durch

$$\bigwedge_{k=1}^{m_i} \neg(a_i^k < T_i < b_i^k) \quad (4.3)$$

ausgedrückt werden, wobei T_i die Besuchszeitvariable für Kunde i ist. Gemeinsam mit der Zeitfensterbedingung

$$a_i \leq T_i \leq b_i \quad (4.4)$$

aus (Pesant u. a. 1998) kann die Zeitfenstereinhaltung dann beschrieben werden.

Beispiel 3 Ein Kunde kann in den Zeiträumen von 8 bis 12 Uhr und von 14 bis 17 Uhr besucht werden. Die Zeitfensterbedingung, zusammengesetzt aus den Bedingungen (4.4) und (4.3), lautet dann

$$(8 \leq T_i \leq 17) \wedge \neg(12 < T_i < 14).$$

Da im Gegensatz zu (Pesant u. a. 1998) für das TSP mit Mehrfachzeitfenstern keine heuristischen oberen Schranken existieren, entwickeln die Autoren eine auf Tiefensuche basierende Strategie zur anfänglichen Bestimmung oberer Schranken.

In (Focacci, Lodi, Milano und Vigo 1998) wird anhand des TSP die Integration von OR-Techniken in globale Constraints gezeigt. Es werden symmetrische und asymmetrische Instanzen des TSP untersucht, wobei das *Assignment Problem* (Zuordnungsproblem) als Relaxation des TSP verwendet wird. Eine inkrementelle Version der Ungarischen Methode aus (Carpaneto, Martello und Toth 1988) wird zur Berechnung der unteren Schranke für den Zielfunktionswert und der reduzierten Kosten jeder einzelnen Variable eingesetzt. Die Ergebnisse zeigen, daß mit diesem Ansatz Probleme, die bis zu einer Größenordnung größer sind als die mit Standardtechniken gelösten Probleme, in der gleichen Zeit lösbar sind. Focacci u. a. verwenden im Modell sowohl eine *alldifferent-Constraint* als auch eine globale Constraint, die das Zuordnungsproblem kapselt. Obwohl eine Lösung des Zuordnungsproblems stets auch für die *alldifferent-Constraint* zulässig ist, sorgt die gemeinsame Verwendung für eine bessere Propagation und reduzierten Suchaufwand, wie das folgende Beispiel der Autoren zeigt.

Beispiel 4 Folgende Elemente eines Constraint Programms seien gegeben:

- $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$,
- $D = \{\{2, 3\}, \{1, 3\}, \{1, 2, 4\}, \{5, 6\}, \{4, 6\}, \{4, 5\}\}$,
- $C_B = \{\text{bounding}([x_1, \dots, x_6], \underline{z}, \bar{z})\}$,

- $C_{AB} = \{\text{alldifferent}([x_1, \dots, x_6]), \text{bounding}([x_1, \dots, x_6], \underline{z}, \bar{z})\}$.

Fall 1 Es wird das Constraint Programm (X, D, C_B) betrachtet. Die Lösung des Zuordnungsproblems ergibt:

$$x_1 = 3, x_2 = 1, x_3 = 2, x_4 = 5, x_5 = 6, x_6 = 4. \quad (4.5)$$

Ohne weitere Constraints könnte in einer der in der Suche folgenden temporären Instanziierungen $x_3 = 4$ gewählt werden. In diesem Fall gibt es aber keine zulässige Lösung des Zuordnungsproblems mehr, weshalb es auch keine Rundreise mehr geben kann. Die Variableninstanziierung ist damit unzulässig und ein Backtracking-Schritt erforderlich.

Fall 2 Es wird das CP (X, D, C_{AB}) betrachtet. Die *alldifferent*-Constraint wird ausgewertet und bei Betrachtung von x_4, x_5, x_6 sowie $D_4 \cup D_5 \cup D_6 = \{4, 5, 6\}$ wird gefolgert, daß $4 \notin D_3$ gelten muß. Die im Fall 1 durchgeführte temporäre Instanziierung $x_3 = 4$ kann daher nicht stattfinden und die Lösung eines Zuordnungsproblems sowie ein Backtracking-Schritt werden eingespart.

In (Focacci, Lodi und Milano 2000) wird die Verwendung von Schnittebenen, wie sie aus der gemischt-ganzzahligen Linearen Programmierung bekannt sind, im CP-Kontext vorgeschlagen. Die Autoren erweitern damit ihre Arbeiten zu Optimierungsconstraints aus (Focacci u. a. 1998) und (Focacci u. a. 1999) in der Absicht, die Relaxationen zu verschärfen, d.h. die untere Schranke zu vergrößern. Bei den Schnittebenen handelt es sich um Ungleichungen der Form

$$\alpha^T x \leq \alpha_0 \quad \alpha, x \in \mathbb{R}^n, \quad (4.6)$$

wodurch Lösungen x' mit $\alpha^T x' > \alpha_0$ ausgeschlossen werden. Um zu verhindern, daß die hinzuzufügenden Schnittebenen die Struktur der Relaxation zerstören, wird in (Focacci, Lodi und Milano 2000) vorgeschlagen, die Schnittebenen per Lagrange-Relaxation in die Zielfunktion zu übernehmen und mit einem Strafterm für den Fall der Verletzung zu versehen. Auf diese Weise wird die Struktur der Nebenbedingungen gewahrt, was es ermöglicht, weiterhin spezialisierte Algorithmen (bspw. die Ungarische Methode) zu verwenden.

In dem Maße wie die Suche voranschreitet und die Wertebereiche der Entscheidungsvariablen eingeschränkt werden, können sich einige der hinzugefügten Schnitte – unabhängig davon, ob global oder lokal erzeugt – als obsolet, da trivialerweise erfüllt, erweisen. Für den Fall der expliziten Aufnahme der Schnittebenen als Nebenbedingung – bspw. in eine LP-Formulierung der Relaxation – verändern diese Redundanzen die untere Schranke nicht, vergrößern jedoch den Problemumfang und verlängern

die Berechnungszeit. Bei Verwendung der Lagrange-Relaxation können auftretende Redundanzen allerdings die untere Schranke abschwächen. In der Tat kann in einem Teilbaum $\alpha^T x < \alpha_0$ gelten, sodaß ein Schnitt der Form $\alpha^T x \leq \alpha_0$ trivialerweise erfüllt ist. Da die Lagrange-Multiplikatoren λ in der Zielfunktion nichtnegativ sind, wird der Zielfunktionswert dann jedoch durch den Zielfunktionsbeitrag $\lambda(\alpha^T x - \alpha_0)$ unnötigerweise verkleinert, d.h. die Relaxation gegenüber der Formulierung ohne den betreffenden Schnitt abgeschwächt.

Focacci u. a. schlagen den Einsatz sog. *Purging*-Techniken (*to purge*: spülen, entleeren) vor, um solche Redundanzen gezielt zu erkennen und die entsprechenden Schnitte aus der Formulierung zu entfernen.

Am Beispiel des Traveling Salesman-Problem (TSP) und des TSP mit Zeitfenstern zeigen die Autoren die Funktionsweise ihres Ansatzes auf, indem im Fall des TSP Bedingungen zur Eliminierung von Subtours einem Zuordnungsproblem hinzugefügt werden. Verschiedene Kombinationen von globalen und lokalen Schnitten, sowie von Lagrange-Relaxation und expliziter Aufnahme von Schnitten in Lineare Programme werden empirisch anhand von TSP-Instanzen aus der Literatur verglichen. Dabei zeigt es sich, daß aufwendige Techniken, bspw. die Berechnung lokaler Schnitte, vergleichsweise weniger Backtracking-Schritte erfordern, die Komplexität der dazu erforderlichen Berechnungen die Gesamtlösungszeit im Vergleich zu einfacheren Ansätzen allerdings erhöht. Die Autoren kommen daher zu dem Schluß, daß sich Schnittebenen-Ansätze für kleinere Problemumfänge nicht eignen, für mittlere und große Probleme jedoch empfehlenswert sind. Bezüglich verschiedener Ansätze der Integration von Schnittebenen kann eine allgemeine Empfehlung jedoch nicht gegeben werden.

4.5 Hybride MIP/CP Ansätze

Die Kombination von gemischt-ganzzahliger Programmierung und Constraint Programming ist ein aktives Forschungsgebiet, das verschiedene Ansätze der Integration beider Methoden hervorgebracht hat. Neben den Lösungsmethoden ist auch die Problemmodellierung vor dem Hintergrund hybrider Ansätze Untersuchungsgegenstand wissenschaftlicher Arbeiten.

4.5.1 Vergleichende Arbeiten

Einige Autoren haben sich mit einem Vergleich der Leistungsfähigkeit von CP- und MIP-Ansätzen zur Lösung kombinatorischer Optimierungsprobleme befasst. Zu diesem Zweck wird das jeweils betrachtete Problem als Constraint Programm und als gemischt-ganzzahliges lineares Programm modelliert und einem CP- bzw. MIP-Solver zur Lösung übergeben. Im folgenden werden einige Arbeiten und die daraus gewonnenen Erkenntnisse vorgestellt.

In (Darby-Dowman, Little, Mitra und Zaffalon 1997) werden MIP- und CP-Ansätze für ein modifiziertes verallgemeinertes Zuordnungsproblem vorgestellt und hinsichtlich ihrer Leistungsfähigkeit verglichen. Die CP-Variante wird dabei als näher am Originalproblem empfunden, was die Formulierung von angepaßten Suchverfahren und redundanten Constraints vereinfacht und infolgedessen die Steuerung der Suche verbessert. Der Vergleich der Ergebnisse beider Verfahren zeigt sehr gute Laufzeiten (bei gleichfalls optimalen Lösungen) des Constraint Programms. Auffällig ist, daß die MIP-Verfahren über die verschiedenen Testinstanzen eine deutlich höhere Varianz der benötigten Rechenzeiten als die Constraint Programme haben.

Darby-Dowman und Little vergleichen in (Darby-Dowman und Little 1998) MIP und CP-Ansätze für ein Golf-Scheduling, ein Crew-Scheduling, ein Maschinen-Konfigurations- und ein Transportproblem und kommen zu ähnlichen Schlüssen wie in ihrem früheren Artikel bezüglich stark beschränkter Probleme. MIP zeigt sich hingegen vorteilhaft bei gut strukturierten Problemen, d.h. solchen, bei denen bspw. eine Netzwerkstruktur oder ein Zuordnungsproblem zugrunde liegt.

Eine der bislang wenigen Veröffentlichungen zu diesem Themengebiet aus Deutschland (dafür allerdings schon vergleichsweise früh) stammt von Jordan und Drexl. MIP und CP werden in (Jordan und Drexl 1995) anhand eines Problems der Einplanung von Fertigungslosen bei reihenfolgeabhängigen Rüstzeiten verglichen. Die Modellierung eines MIP wird von den Autoren im Vergleich zu CP als komplexer angesehen. Tendenziell sind die Rechenzeiten des CP-Ansatzes länger, allerdings ist der MIP-Solver nicht immer in der Lage, das Problem optimal zu lösen¹. Die Rechenzeiten von CP und MIP-Solver verhalten sich zudem gegenläufig: Stark beschränkte Instanzen werden vom CP-Solver schneller als schwach beschränkte gelöst, während der eingesetzte IBM-OSL MIP-Solver sich umgekehrt verhält.

In (Ottosson und Thorsteinsson 2000) wird ein industrielles Konfigurationsproblem mit Methoden des CP, MIP und dem hybriden Werkzeug MLLP untersucht. Kernelement ist eine lineare Relaxation einer *element*-Constraint (diese ist gleichbedeutend mit einem variablen Index) im Zusammenspiel mit der Domainreduktion durch reduzierte Kosten nach (Focacci, Lodi und Milano 1999). Die Rechenzeiten für einen Satz von Testinstanzen sind für das hybride MLLP-Modell bis zu einer Größenordnung geringer als für das MIP-Modell. Die Domainfilterung mittels reduzierter Kosten der Relaxation der *element*-Constraint halbiert nahezu sowohl die Suchbaumgröße als auch die Lösungszeit im Vergleich zum MLLP-Ansatz ohne diese Technik.

4.5.2 Integrierte Modellbildung

Der enge Zusammenhang zwischen Modellierung und Lösung wird in (Ottosson 2000) betont. Der Autor sieht durch die sich abzeichnende Verschmelzung von MIP und CP auf dem Gebiet der Methoden auch einen Bedarf nach neuen Wegen der Modellierung,

¹Von den Autoren werden keine Angaben über die Abbruchgründe gemacht.

die nicht mit den bisherigen Nachteilen verbunden sind. Als solche bezeichnet der Autor einerseits die mangelnde Flexibilität bei algebraischen Modellierungssystemen (bspw. AMPL, s. (Fourer, Gay und Kernighan 1993)) bedingt durch die Beschränkung auf lineare Ungleichungen, sowie andererseits die hohe Komplexität von CP Modellen. Als erstrebenswerte Lösung wird in (Ottosson 2000) eine flexiblere algebraische Modellierungssprache gesehen, die auch Elemente des Lösungsverfahrens umfaßt, indem Preprocessing, Inferenzmechanismen und Suchstrategien durch den Anwender definiert werden können.

Eine neuere Entwicklung der letzten Jahre stellt die Sprache und Modellierungsumgebung OPL (bzw. OPL Studio) durch die Firma ILOG dar (siehe (van Hentenryck 1999)), die als ein vielversprechender erster Schritt zu einer hybriden Modellierung angesehen werden kann. Aus der Umgebung heraus können CP- und LP-Solver angestoßen werden, wobei beide Teile dieselben Variablen verwenden. Die Steuerung der Suche ist in gewissem Umfang möglich und überwindet daher die zuvor angesprochenen Grenzen traditioneller algebraischer Modellierungssysteme.

In (Hooker, Ottosson, Thorsteinsson und Kim 1999) wird die Notwendigkeit aufgezeigt und ein möglicher Lösungsweg einer einheitlichen deklarativen Modellierung angegeben, welcher es erlaubt, sowohl CP- als auch MIP-Methoden zu integrieren. Von einigen der Autoren zunächst in (Hooker und Osorio 1999) vorgestellt und anschließend weiterentwickelt, wird die Sprache MLLP beschrieben, deren Hauptmerkmal die Formulierung der Constraints in Form von Implikationen des Typs

$$h(y) \Rightarrow Ax \geq b, \quad x \in \mathbb{R}^n, y \in D$$

ist. Die linke Seite der Implikation bilden dabei beliebige Constraints auf diskreten Variablen, während die rechte Seite lineare Ungleichungssysteme beinhaltet. Die Form der Implikation erlaubt es, durch Angabe von immer wahren bzw. immer falschen Termen auf der linken bzw. rechten Seite, lineare Ungleichungssysteme bzw. reine Constraints abzubilden. Die Möglichkeiten der Modellierungssprache werden anhand eines Mehrmaschinen-Scheduling-Problems vorgestellt. Aufgrund der Struktur der Implikationen sind Folgerungen nur von diskreten Variablen der linken Seite auf stetige Variablen der rechten Seite möglich – hauptsächlich um durch fixierte stetige Variablen das zu lösende LP auf der rechten Seite so klein wie möglich zu halten. Ein Weg für umgekehrte Schlüsse von einer LP-Lösung auf Wertebereiche der diskreten Variablen wird durch eine *back propagation* genannte Technik in (Ottosson, Thorsteinsson und Hooker 1999) aufgezeigt.

Gegenstand der Arbeit (Ottosson 2000) sind Entwicklungen für einen hybriden Solver zur Lösung von kombinatorischen Optimierungsproblemen. Dabei beziehen sich diese Entwicklungen sowohl auf die Modellierung als auch auf die Lösungsfindung. In der Arbeit wird eine erweiterte Version des MLLP Frameworks aus (Hooker und Osorio 1999) entwickelt, das aus einer auf AMPL (s. (Fourer, Gay und Kernighan 1993)) basierenden Modellierungssprache und einem Solver besteht, dessen CP- und LP-Teil nach außen hin als ein hybrider Solver auftreten, d.h. für den Anwender nicht

einzelnen sichtbar sind. Die Ansprache der einzelnen Solver geschieht vielmehr durch einen Übersetzerlauf über das Modell, in dem den Solvern Teilmodelle zugewiesen werden. Beispielsweise werden lineare Ungleichungen direkt an den LP-Solver weitergereicht, während der CP-Solver komplexere Constraints bearbeitet.

Eine völlige Integration in dieser Form kann als das genaue Gegenteil des Ansatzes von (Heipcke 1999) gesehen werden, bei dem explizit beide Solver ihre Eigenständigkeit behalten und es zudem zwar gekoppelte, aber dennoch getrennte Modelle für beide gibt.

Der von (Heipcke 1999) verfolgte Weg ist der des *Double Modeling*: Das zu lösende Problem wird jeweils vollständig für den LP/MIP und den CP-Solver formuliert. Über ein zwischengeschaltetes Kommunikationsmodul werden die einander entsprechenden Variablen und damit beide Modelle gekoppelt. Dies geschieht in der Erwartung, daß auf diese Weise die jeweiligen Stärken der Verfahren voll zur Geltung kommen.

4.5.3 Kombination von OR und CP

Ein grundlegendes Werk über die Integration von Logik und Optimierungsverfahren des OR ist (Hooker 2000). Aus der Perspektive des Autors ist die Integration von Constraint Programming und OR ein Spezialfall dieses übergeordneten Prinzips. Obwohl derzeit noch kein Werkzeug und keine vereinheitlichende Methode für die Integration existieren, zeigen die Ausführungen in diesem Band die zugrundeliegenden Prinzipien und theoretischen Grundlagen dafür.

Die Integration von Methoden der Mathematischen Programmierung, insbesondere der Linearen Programmierung, hat auf dem Gebiet des Constraint Programming bereits eine gewisse Tradition, da bspw. der Simplex-Algorithmus für die Lösung von Systemen linearer Constraints mit reellen Zahlen verwendet wird. Umgekehrt werden Methoden des CP erst seit kurzer Zeit von der OR-Gemeinschaft weitergehend untersucht.

Für die Integration beider Methodenwelten gibt es prinzipiell die Alternativen der gleichrangigen oder der hierarchischen Kombination. In den folgenden Unterabschnitten wird auf Arbeiten zu diesen zwei Vorgehensweisen näher eingegangen.

Gleichrangige Kombination von MIP und CP

Zentrales Element des Constraint Programming ist die Wertebereichsreduktion durch logische Schlüsse. Gemischt-ganzzahlige lineare Programme bieten eine globale Sicht auf eine Menge linearer Gleichungen und Ungleichungen und können damit einen wertvollen Beitrag zur Domainverkleinerung leisten, sofern Constraints in dieser Form vorliegen. In einer gleichrangigen Kombination ergänzen sich CP- und MIP-Methoden bei der Lösung desselben Problems (jedoch nicht unbedingt desselben Modells), indem die Wertebereiche entweder aus lokaler (CP) oder globaler (MIP) Perspektive reduziert werden.

Eine einheitliche Sicht auf CP und MIP mittels formaler Logik wird in (Bockmayr und Kasper 1998) und (Bockmayr und Kasper 2004) vorgestellt. Bockmayr und Kasper schlagen darin vor, MIP um symbolische Constraints aus dem CP zu erweitern, weil die Modellierungsmöglichkeiten von CP mächtiger als diejenigen von MIP sind. In algorithmischer Hinsicht wird eine Integration der effizienten MIP-Verfahren und CP-Methoden angestrebt. Als Mittel dazu wird ein Konzept für ein vereinheitlichtes System von Constraints, Pruning- und Suchmethoden vorgestellt.

Für ein ganzzahliges mehrdimensionales Rucksackproblem zeigen (Behringer und de Backer 1995) die Verwendung einer LP-Relaxation im Zusammenspiel mit einem CP-Solver. Da der CP-Teil prinzipbedingt keine stetigen Variablen verarbeiten kann, werden die entsprechenden Constraints im LP-Teil abgebildet und über eine LP-Relaxation gelöst. Wertebereichsveränderungen von Variablen, die auch im LP-Modell enthalten sind, werden dem LP-Solver mitgeteilt. Umgekehrt gibt der LP-Solver fixierte, d.h. fest zugeordnete, Variablen an den CP-Solver zurück.

Eine weitere Möglichkeit wird im Solver ECLiPSe verfolgt (siehe (Rodošek, Wallace und Haijan 1999)), bei dem während der Modellierung lineare Constraints syntaktisch über spezielle Symbole als von einem angeschlossenen LP-Solver zu verarbeitende Constraints gekennzeichnet werden. Die Übertragung, Lösung und Rückübertragung der Ergebnisse wird dabei transparent vom CP-Solver gehandhabt. Der Unterschied zum Ansatz von Behringer und de Backer besteht demnach darin, daß Constraints nicht für jeden Solver getrennt aufgestellt werden müssen.

In (Jain und Grossmann 1999) wird ebenfalls ein Schema vorgestellt, das ein Problem in je einen vom MIP-Solver bzw. CP-Solver zu behandelnden Teil zerlegt. Für ein Mehrmaschinen-Scheduling-Problem wird diese Zerlegung gezeigt, wobei die Zuordnung von Jobs zu Maschinen per MIP und die Reihenfolgebestimmung auf den einzelnen Maschinen per CP gelöst werden. Die Lösung verläuft iterativ, indem zunächst die Zuordnung und anschließend die Reihenfolge berechnet werden. Wann immer sich die Reihenfolge als unzulässig erweist, werden Schnittebenen für den MIP-Teil erzeugt und die Lösungssuche fortgesetzt.

Hierarchische Kombination von MIP und CP

Ein Gebiet, auf dem bisher die Integration von OR und CP Methoden sowohl theoretisch als auch in der Anwendung vorangetrieben wurde, ist das Crew Scheduling im Luftfahrtbereich (s. z.B. (Kohl 1999)).

Beim Crew Scheduling werden – zunächst auf der Ebene abstrakter Mitarbeiter – im *Crew Pairing*-Problem Zuordnungen von Crewmitgliedern bestimmter Qualifikation zu Flugabschnitten („flight legs“) erzeugt. Beginnend und endend in einer Basis sollen alle Flugabschnitte dabei mit Crewmitgliedern der richtigen Qualifikation (bspw. Pilot mit Zulassung für Flugzeugtyp, Flugbegleiter mit entsprechenden Sprachkenntnissen) abgedeckt werden. Unter einem *Pairing* versteht man dabei eine unteilbare Arbeitseinheit, die zwar mehrere Flüge und Tage umfassen kann, aber als Ganzes Mitarbeitern

zugewiesen wird. Im Rahmen des anschließenden *Crew Assignment* werden den Pairings dann konkrete Mitarbeiter zugeordnet, wobei hier noch Dienste wie Bereitschaft und Bodendienste, Urlaubszeiten und zusätzliche individuelle Regelungen und Vorlieben berücksichtigt werden müssen.

Beim Crew Assignment-Problem handelt es sich um sehr große Optimierungsprobleme, die per Column Generation gelöst werden können, wobei das Subproblem sich als Kürzeste-Wege-Problem mit Nebenbedingungen darstellt und üblicherweise per Dynamischer Programmierung gelöst wird.

Aufgrund der Vielzahl und Komplexität der zu berücksichtigenden vertraglichen und rechtlichen Rahmenbedingungen und Regeln im Bereich der Luftfahrt haben Junker u. a. in (Junker, Karisch, Kohl, Vaaben, Fahle und Sellmann 1999) Constraint Programming zur Modellierung und Lösung des Subproblems vorgeschlagen und ein Framework zur Integration beider Techniken vorgestellt. Der vorgeschlagene Weg sieht vor, CP-Algorithmen in einen Column Generation-Rahmen einzubetten; im Speziellen das Subproblem per CP zu lösen und das Master-Problem durch Lineare Programmierung zu behandeln. Dabei findet ein Austausch von Variablen bzw. Dualvariablen statt, der den Suchprozeß lenkt. Nach Einschätzung von Junker u. a. ist dieser Informationsaustausch Integrationsansätzen überlegen, die darauf abzielen, lediglich die Domainreduktion zu verbessern. Dies liegt daran, daß das CP über die Dualvariablen globale Informationen über erfolgversprechende Lösungsstrukturen erhält.

Fahle u. a. entwickeln in (Fahle, Junker, Karisch, Kohl, Sellmann und Vaaben 2002) einen CP-basierten Column Generation-Ansatz zur Lösung des Crew Assignment-Problems. Das Subproblem besteht dabei in der Erstellung eines kostenminimalen Einsatzplans für die einzelnen Mitarbeiter und läßt sich als Kürzeste-Wege-Problem modellieren, wobei die Knoten des zugrundeliegenden Graphen einzelnen Aufgaben entsprechen und mit bewerteten gerichteten Kanten verbunden werden, falls sie zeitlich nacheinander ausführbar sind. Zur effizienteren Suche führen die Autoren eine Pfad-Constraint ein, die explizit die Berechnung von Lösungen mit negativen reduzierten Kosten beinhaltet, weil somit bereits frühzeitig Äste des Suchbaums abgeschnitten werden, die keine Variablen mit negativen reduzierten Kosten ergeben können. Die aufgrund der zeitlich gerichteten Kanten azyklische Struktur des Graphen kann dabei vorteilhaft genutzt werden. Die Suche im Suchbaum wird über Heuristiken gesteuert, in deren Gestaltung Erfahrungen aus der Praxis eingeflossen sind.

Ein anderer Ansatz der Kombination von Constraint Programming und Column Generation wird in (Grönkvist 2003) verfolgt. Während die oben beschriebenen Ansätze Constraint Programming zur Lösung des Pricing-Problems verwenden, wird von Grönkvist das Problem zunächst mittels CP reduziert, um es anschließend mit der Methode der Spaltenerzeugung zu lösen. Konkret wird das sog. *Tail Assignment*-Problem behandelt – eine Aufgabe der Flugplanung, bei der konkrete Flugzeuge planmäßigen Flügen unter Beachtung von Wartungszeiten, Flugverboten und unvermeidbaren Bodenzeiten zugeteilt werden müssen. Dieses Problem wird als ressourcenbeschränktes Mehrfachgüter-Netzwerkflußproblem dargestellt. Die Knoten des Netzwerkes ent-

sprechen dabei einzelnen Flügen und zwei Knoten werden dabei – ähnlich wie oben beim Crew Scheduling – durch eine gerichtete Kante verbunden, wenn sie zeitlich nacheinander ausgeführt werden können. Constraint Programming wird in einer LP-Preprocessingphase zur Filterung von ungültigen Verbindungen verwendet. Neben Zulässigkeitsüberlegungen werden aufgrund von Optimalitätsüberlegungen diejenigen Verbindungen entfernt, die nicht Bestandteil einer guten Lösung sein können – wobei die notwendigen oberen Schranken heuristisch ermittelt werden. Das Ergebnis der Filterung wird danach an einen Column Generation-Algorithmus übergeben. Die experimentellen Untersuchungen anhand von Testinstanzen mit realen Flugdaten amerikanischer und europäischer Fluglinien zeigen, daß die vorgeschaltete Filterung die Lösungsgüte (die infolge der Begrenzung der Iterationszahlen von der Rechenzeit abhängig ist) und -geschwindigkeit teilweise sehr stark steigern kann. Allerdings ist eine genauere Quantifizierung schwierig, da die beobachteten Verbesserungen sehr stark von der Netzwerkstruktur der Testinstanzen abhängig sind.

Aufbauend auf den Arbeiten von (Pesant, Gendreau, Potvin und Rousseau 1998) und (Junker, Karisch, Kohl, Vaaben, Fahle und Sellmann 1999) wird in (Rousseau, Gendreau und Pesant 2002) erstmalig das Tourenplanungsproblem mit Zeitfenstern durch eine Kombination von Column Generation und Constraint Programming gelöst. Insbesondere werden Techniken zur Eliminierung von Variablen entwickelt, durch die der Suchaufwand beträchtlich reduziert werden kann.

Das Restricted Master-Problem ist als Set Covering-Problem formuliert, die Routenerzeugung geschieht durch Lösen eines *Resource Constrained Profitable Tour*-Problems. Hierbei handelt es sich um das Problem, die zurückgelegte Distanz einer Route zu minimieren und gleichzeitig die auf dieser Route erworbenen Prämien zu maximieren. Die Prämien ergeben sich im konkreten Problem durch die Dualvariablen des RMP, die aufgrund der Set Covering-Formulierung nur nichtnegative Werte annehmen können. Mathematisch gesehen ist diese Sichtweise allerdings völlig identisch mit einem Kürzeste-Wege-Problem in Graphen mit negativen Zyklen mit Kantenbewertungen $c'_{ij} = c_{ij} - u_i$, wobei u_i die dem Kunden i entsprechende Dualvariable aus dem RMP ist.

Die Autoren heben den Vorteil des Einsatzes von CP gegenüber der Dynamischen Programmierung heraus, indem sie auf die Möglichkeit aufmerksam machen, Bedingungen wie Mehrfachzeitfenster, Prioritäten und Reihenfolgen, sowie beliebige andere logische Beziehungen zwischen Kundennachfragen einbeziehen zu können. Weiterhin wird darauf hingewiesen, daß in Abgrenzung zu Anwendungen des CP bei Crew Scheduling Problemen der zugrundeliegende Graph beim Tourenplanungsproblem zyklisch ist und übliche Transformationsmethoden in zyklensfreie Graphen zu prohibitiv großen Graphen führen.

Neben Constraints zur Sicherung der Konsistenz zwischen Vorgänger- und Nachfolgervariablen und Implikationen zur Berechnung der korrekten Besuchszeiten und Ladungen verwenden Rousseau, Gendreau und Pesant die `NoSubTour`-Bedingung aus (Pesant, Gendreau, Potvin und Rousseau 1998) zur Vermeidung von Kurzzyklen. Eine

zweite globale Constraint sichert den Zusammenhang des Graphen, d.h. gewährleistet, daß es einen Pfad von Start- zu Enddepot gibt, und liefert zudem noch obere Schranken für den Zielfunktionswert.

Eine ganze Klasse von Constraints wird mit den *ArcElimination*-Constraints eingeführt. Ausgehend von der Voraussetzung, daß die Dreiecksungleichung für alle Ressourcen gilt, werden durch drei Typen von Constraints diejenigen Kanten entfernt, die nicht Bestandteil einer optimalen Lösung sein können.

Die entwickelte Methode wird anhand der bekannten Solomon-Testinstanzen analysiert. Bei der Untersuchung der Effektivität der *ArcElimination*-Constraints wird insbesondere der Frage nachgegangen, ob diese einmal pro Suchknoten, d.h. nach einer Wertzuweisung, oder stattdessen bei jeder Wertebereichsänderung angestoßen werden sollen. Im Ergebnis wird deutlich, daß die Prüfung nach jeder Domainänderung zwar die Zahl der Backtrackingschritte senkt, aber die Ausführungszeit verlängert. Insgesamt zeigt sich, daß *ArcElimination* die Zahl der Backtrackingschritte um zum Teil mehrere und die Ausführungszeit um eine Größenordnung senken kann. Die Ergebnisse lassen auch erkennen, daß das Zusammenspiel von *ArcElimination* mit *Bounding*-Constraints zu weiteren starken Verbesserungen führt.

4.6 Heuristische Verfahren

Heuristische Lösungsverfahren für Routing- und insbesondere Tourenplanungsprobleme werden bereits seit vielen Jahren im OR entwickelt. Auch heute noch werden in kommerziellen Systemen zur Tourenplanung Varianten von Heuristiken eingesetzt, die in den 60er Jahren entwickelt wurden. In den letzten Jahren lag der Fokus von Neuentwicklungen auf dem Gebiet der Metaheuristiken.

Klassische Heuristiken und Verbesserungsverfahren werden implementationsnah in (Gietz 1994) beschrieben. Ein neuerer Vergleich von verbreiteten Heuristiken findet sich in (Cordeau, Gendreau, Laporte, Potvin und Semet 2002), während Funke in seiner Arbeit (Funke 2003) einen allgemeinen Rahmen lokaler Suchverfahren präsentiert. Der Band (Toth und Vigo 2002) geht neben exakten Verfahren auch auf Heuristiken und Metaheuristiken für das Standardtoursplanungsproblem und typische Varianten davon ein.

In dieser Arbeit wird die Darstellung der Literatur auf Kombinationen von lokaler Suche und Constraint Programming beschränkt, weil Verfahren dieses Typs zur Bestimmung einer Initiaillösung verwendet werden.

4.6.1 Kombination von Lokaler Suche und Constraint Programming

Die Kombination von Constraint Programming mit Methoden der lokalen Suche verspricht die Verbindung mächtiger Problemeingabe- und -formulierungsmöglichkeiten mit leistungsfähigen Lösungsverfahren für große Probleme.

Mit dem in (Gendreau, Hertz und Laporte 1992) vorgestelltem *GENIUS*-Verfahren entwickelten die Autoren ein aus zwei Komponenten bestehendes lokales Suchverfahren. Bei den Komponenten handelt es sich zum einen um die verallgemeinerte Einfügeprozedur *GENI*, die einen Einfügeschritt mit einer k -opt Reihenfolgeverbesserung kombiniert, und zum anderen um *US*: Einer Prozedur zum Entfernen des Kunden, der im nächsten Verfahrensschritt durch *GENI* eingefügt wird. In (Pesant, Gendreau und Rousseau 1997) wird die Flexibilität von Constraint Programming zur Prüfung der Zulässigkeit eingesetzt und mit der *GENIUS*-Heuristik kombiniert. Dies erlaubt eine Änderung des Modells, bspw. das Hinzufügen neuer Constraints, ohne das Suchverfahren verändern zu müssen.

Die Kombination von CP mit verbreiteten lokalen Suchverfahren für das Tourenplanungsproblem, u.a. 2-opt, Relocate, Exchange und Cross wird von de Backer u. a. untersucht. Das Verfahren arbeitet pro Iterationsschritt mit einer aktuellen Lösung und einer Arbeitskopie derselben. Die lokalen Suchverfahren verändern die Arbeitskopie, wobei der CP-Teil nach einem Schritt die Gültigkeit der neuen Lösung prüft und im Fall einer Verletzung die Veränderungen an der Arbeitskopie wieder zurücknimmt. Am Ende eines Iterationsschrittes wird (falls existent) eine neue bessere zulässige Lösung zur aktuellen Lösung und der nächste Iterationsschritt schließt sich an. Auch hier wird der CP-Teil nicht für die Suche, sondern lediglich für die Zulässigkeitsprüfung benutzt.

In (Shaw 1998) wird CP innerhalb einer Nachbarschaftssuche eingesetzt. Der Autor schlägt mit *Large Neighborhood Search* eine Methode zum Durchsuchen großer Nachbarschaften vor, bei der ausgehend von einer Lösung für eine Teilmenge von Variablen die Wertebereiche wieder vergrößert werden und dadurch ein Suchraum aufgespannt wird, der dann per CP durchsucht wird.

Zwei weitere Artikel wählen den Weg Constraint-basierter Operatoren. Hierbei wird die Nachbarschaft lokaler Suchverfahren durch eine Menge von Constraints beschrieben und auch mittels CP-Suchverfahren durchsucht. Damit eröffnet sich wiederum das volle Repertoire der vollständigen oder unvollständigen Suche des Constraint Programming und der Einsatz von Optimierungsconstraints.

Pesant und Gendreau vergleichen unterschiedliche Grade der Nutzungsmöglichkeiten von CP im Kontext lokaler Suche in (Pesant und Gendreau 1999). Die Autoren entwickeln eine Constraint-basierte Beschreibung der Nachbarschaft eines richtungserhaltenden 3-opt-Schrittes und setzen zur Steuerung der lokalen Suche Tabu Search ein. Es zeigt sich anhand von Testinstanzen für das Handlungsreisendenproblem, daß komplexe CP-Suchverfahren – die in diesem Fall nur eine Nachbarschaft untersuchen – im

Verbund mit Metaheuristiken nicht immer zu effizienten Verfahren führen. Stattdessen kann eine weniger laborierte Suche im Endeffekt schneller sein. Im Ergebnis ist die Kombination von lokaler Suche und Constraint Programming durchaus vergleichbar mit herkömmlichen Verfahren und übertrifft diese bei komplexeren Nachbarschaften.

In (Rousseau und Gendreau 2002) werden ebenfalls Constraint-basierte Operatoren für das Tourenplanungsproblem vorgestellt. Die Operatoren werden in einem Variable Neighborhood-Algorithmus eingebettet. Bei diesem heuristischen Verfahren werden Schritte aus einer Nachbarschaft so lange ausgeführt, bis ein lokales Optimum erreicht wird bzw. eine heuristische Prüfung diesen Schluß nahelegt. Danach wird eine andere (möglichst strukturell verschiedene) Nachbarschaft gewählt und das Verfahren wiederholt. Hinter diesem Vorgehen steht die Erwartung, daß ein lokales Optimum bezüglich einer Nachbarschaft bezogen auf eine andere Nachbarschaft kein lokales Optimum ist und weitere Verbesserungsschritte gefunden werden können. Für den Fall, daß ein lokales Optimum bezüglich aller Nachbarschaften erreicht wird, verändert eine kleine Perturbation die aktuelle Lösung in einer Diversifikationsphase.

4.7 Bilanz des Literaturüberblicks

Aus der Analyse der Literatur lassen sich Erkenntnisse für die Gestaltung des hybriden Verfahrens zur Lösung des Tourenplanungsproblems mit alternativen Liefermöglichkeiten ziehen. Die folgenden Punkte lassen sich als Ergebnis aus der Literaturanalyse festhalten:

- Column Generation eignet sich aufgrund des hierarchischen Aufbaus sehr gut für Kombinationen von Linearer Programmierung für das (Restricted) Master-Problem und anderer Verfahren für das Pricing-Problem,
- für die Modellierung und Lösung von Problemen mit komplexen Nebenbedingungen ist Constraint Programming sehr gut geeignet,
- das Framework von Junker u. a. beschreibt mit Constraint Programming-basiertem Column Generation einen mit heutigen Werkzeugen umsetzbaren Integrationsweg,
- für die Lösung von Optimierungsaufgaben mit Constraint Programming sollten Optimierungsconstraints zur Verbesserung der Domainreduktion und Propagation verwendet werden,
- Relaxationen, zu deren Lösung effiziente Algorithmen verfügbar sind, bieten sich als Optimierungsconstraints an,
- globale Constraints erlauben die Kapselung und damit die Wiederverwendbarkeit sowie effiziente Filtertechniken für Constraintgruppen und Optimierungsconstraints.

Die Ergebnisse dieser Analyse fließen in den beiden folgenden Kapiteln in die Modellierung und Entwicklung des hybriden Verfahrens ein.

5 Modellierung des Tourenplanungsproblems mit alternativen Liefermöglichkeiten

*Je vertrauter und alltäglicher eine Verhaltensweise ist,
desto problematischer wird ihre Analyse*
Desmond Morris

Nach (VDI 3633 1993) handelt es sich bei einem Modell „um eine vereinfachte Nachbildung eines geplanten oder real existierenden Originalsystems mit seinen Prozessen in einem anderen begrifflichen oder gegenständlichen System. Es unterscheidet sich hinsichtlich der untersuchungsrelevanten Eigenschaften nur innerhalb eines vom Untersuchungsziel abhängigen Toleranzrahmens vom Vorbild.“

Modelle lassen sich nach ihrem Einsatzzweck in Beschreibungsmodelle, Erklärungsmodelle und Entscheidungsmodelle unterteilen. Besonders hervorzuheben sind quantitative mathematische Entscheidungsmodelle, die neben den Modellgrößen und ihren Beziehungen untereinander über ein Zielsystem die Bewertung von Alternativen erlauben. Die Wichtigkeit mathematischer Modelle ergibt sich auch aus der infolge der mathematischen Abstraktion gewonnenen Verallgemeinerung des ursprünglichen Problems, die hinter Details verborgene Strukturen offenbart und zwischen auf den ersten Blick verschiedenen Problemen strukturelle Zusammenhänge aufzeigt. Gerade Entscheidungsprobleme der Logistik lassen sich durch eine Abbildung als mathematisches Modell häufig auf eine Komposition bekannter kombinatorischer Optimierungsprobleme zurückführen.

Das Tourenplanungsproblem als Kombination von Zuordnungs- und Reihenfolgeproblem unter Nebenbedingungen ist hierfür ein Beispiel. In den folgenden Abschnitten werden mathematische Modelle für das Tourenplanungsproblem mit alternativen Liefermöglichkeiten entwickelt. Zunächst werden Modellelemente des Standardtoursplanungsproblems mit Kapazitäts- und Zeitfensterrestriktionen eingeführt, um darauf aufbauend ein kompaktes MIP-Modell für das Tourenplanungsproblem mit alternativen Liefermöglichkeiten herzuleiten. Da sich das kompakte Modell für eine effiziente Lösung nicht eignet, wird ein Dantzig-Wolfe-Dekompositionsansatz vorgeschlagen,

der das Problem in ein Master- und ein Pricing-Problem zerlegt. Letzteres weist dabei die problemspezifischen Restriktionen auf und wird als Constraint Program modelliert. Die mathematische Modellierung erlaubt eine widerspruchsfreie und unmißverständliche Beschreibung des Problems. Darüberhinaus ist sie allgemein genug um weitere Anwendungsfälle abzudecken. Auf einige dieser weiteren Anwendungsmöglichkeiten, die sich als Sonderfälle des Tourenplanungsproblems mit alternativen Liefermöglichkeiten darstellen lassen, wird abschließend eingegangen.

5.1 Das Standardtoursplanungsproblem mit Kapazitäts- und Zeitfensterrestriktionen

Das Modell des Standardtoursplanungsproblems mit Kapazitäts- und Zeitfensterrestriktionen ist Grundlage für die Abbildung des Tourenplanungsproblems mit alternativen Liefermöglichkeiten.

Es handelt sich um ein graphentheoretisches Problem, d.h. es läßt sich als Optimierungsproblem auf einem mathematischen Graphen darstellen. Ein Graph $G = (\mathcal{N}, \mathcal{E})$ besteht aus einer Knotenmenge $\mathcal{N} = \{0, 1, \dots, n, n+1\}$ und einer Kantenmenge \mathcal{E} .

Die Knoten 0 und $n+1$ bezeichnen das Ausgangs- bzw. Enddepot und zur Vereinfachung der Darstellung wird die Menge $\mathcal{C} = \{1, \dots, n\}$ als die Menge aller Kunden eingeführt. Ausgangs- und Enddepot können identisch sein, die Modellierung als zwei verschiedene Knoten erlaubt aber eine einfachere Formulierung des mathematischen Modells.

Eine Kante $\langle i, j \rangle \in \mathcal{E}$ mit $i, j \in \mathcal{N}, i \neq j$ ist gerichtet und beschreibt eine direkte Verbindung zwischen den Knoten i und j . Ferner ist eine Funktion $c(e) \in \mathbb{R}_+$ mit $e \in \mathcal{E}$ definiert, die die Kosten der Benutzung einer Kante angibt. Typischerweise spiegeln diese Kosten Entfernungen, Fahrzeiten oder eine gewichtete Kombination davon wider. Es wird von gerichteten Kanten ausgegangen, da insbesondere im städtischen Umfeld sich aufgrund von Einbahnstraßen und Abbiegeverböten große Unterschiede zwischen den Kosten der Kanten $\langle i, j \rangle$ und $\langle j, i \rangle$ ergeben können¹. Da Kanten üblicherweise nicht explizit indiziert werden, verwendet man zur Angabe der Kosten der Kante $e = \langle i, j \rangle$ den Koeffizienten c_{ij} . Die Gesamtheit dieser Koeffizienten geht als Kostenmatrix (c_{ij}) in das Modell ein.

In einer Problemstellung mit Zeitfenstern oder anderen zeitlichen Restriktionen sind die Reisezeiten entlang der Kanten entscheidungsrelevant. Die Reisezeiten $t_{ij} \geq 0$ zwi-

¹Ein häufiges Mißverständnis besteht darin, die einzelnen Kanten als Straßenverbindungen zu interpretieren. Diese Sichtweise trifft für sog. *Routenplanungsprobleme* zu, bei denen der kürzeste, schnellste oder sicherste Weg zwischen zwei Orten gesucht wird. Im Graphen des Tourenplanungsproblems hingegen stellt jede Kante bereits einen (im Sinne der Kostenfunktion) kürzesten Weg dar. Demzufolge gibt eine Route, d.h. eine Abfolge von Knoten, im Graphen des Tourenplanungsproblems eine Besuchsreihenfolge an und ist keine Streckenbeschreibung.

schen den Knoten i und j werden analog zu den Fahrtkosten bestimmt und sind Eingangsgrößen des Modells.

Sowohl für die Kostenmatrix (c_{ij}) als auch für die Fahrzeitenmatrix (t_{ij}) wird gefordert, daß die Dreiecksungleichung gilt. Dies bedeutet, daß der direkte Weg zwischen zwei Knoten stets kürzer als ein Weg über einen dritten Knoten sein muß, d.h.

$$c_{ij} \leq c_{ik} + c_{kj} \quad \forall i, j, k \in \mathcal{N} \quad (5.1)$$

und

$$t_{ij} \leq t_{ik} + t_{kj} \quad \forall i, j, k \in \mathcal{N}. \quad (5.2)$$

Eine weitere Modellgröße ist die Menge der verwendeten Fahrzeuge \mathcal{V} . Fahrzeuge stellen die zu disponierende Ressource dar und sind das Objekt der im folgenden Abschnitt erläuterten Entscheidungsvariablen.

5.1.1 Entscheidungsvariablen

In der Einleitung dieses Kapitels wurde das Tourenplanungsproblem als Kombination von Zuordnungs- und Reihenfolgeproblem erkannt. Das Zuordnungsproblem besteht in der Zuweisung von Kunden zu Fahrzeugen, das Reihenfolgeproblem in der Bestimmung der Besuchsabfolge für jedes eingesetzte Fahrzeug. Beide Entscheidungen lassen sich durch dreifach indizierte Entscheidungsvariablen x_{ijk} verbinden. Diese nehmen den Wert 1 an, falls das Fahrzeug $k \in \mathcal{V}$ vom Knoten $i \in \mathcal{N} \setminus \{n+1\}$ ausgehend unmittelbar zum Knoten $j \in \mathcal{N} \setminus \{0\}$ fährt und hat anderenfalls den Wert 0.

Für die Prüfung der zeitlichen Restriktionen wird ein zweiter Satz von Entscheidungsvariablen s_{ik} verwendet. Diese geben den Zeitpunkt an, zu dem Fahrzeug k mit der Bedienung des Kunden i beginnt.

Mit jedem Fahrzeugeinsatz sind einsatzfixe Kosten in Höhe von f verbunden. Durch die binäre Entscheidungsvariable y_k wird entschieden, ob Fahrzeug $k \in \mathcal{V}$ eingesetzt wird ($y_k = 1$) oder nicht ($y_k = 0$).

5.1.2 Strukturbedingungen

Unter *Strukturbedingungen* wird ein Satz von Nebenbedingungen verstanden, der sicherstellt, daß die Entscheidungsvariablen x_{ijk} eine Menge von einzelnen Touren beschreiben, die insgesamt alle Kunden abdecken. Dies kann formal folgendermaßen ausgedrückt werden:

$$\sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{V}} x_{ijk} = 1 \quad \forall i \in \mathcal{C} \quad (5.3)$$

$$\sum_{j \in \mathcal{N}} x_{0jk} = 1 \quad \forall k \in \mathcal{V} \quad (5.4)$$

$$\sum_{i \in \mathcal{N}} x_{ihk} = \sum_{j \in \mathcal{N}} x_{hjk} \quad \forall h \in \mathcal{C}, \forall k \in \mathcal{V} \quad (5.5)$$

$$\sum_{i \in \mathcal{N}} x_{i,n+1,k} = 1 \quad \forall k \in \mathcal{V}. \quad (5.6)$$

Durch Gleichung (5.3) wird angegeben, daß jeder Kunde genau einmal besucht wird. Bedingungen (5.4) bzw. (5.6) fordern, daß jedes Fahrzeug der Flotte das Ausgangsdepot verläßt und zum Enddepot zurückkehrt. Damit ein den Kunden h besuchendes Fahrzeug k diesen auch wieder verläßt, wird die Gleichheit von Quellen- und Senkenhäufigkeit pro Kunde und Fahrzeug in der Flußbedingung (5.5) gesichert. Besucht ein Fahrzeug k' keine Kunden, d.h. das Fahrzeug wird nicht benötigt, ist $x_{0,n+1,k'} = 1$.

5.1.3 Fahrzeugrestriktionen

Üblicherweise wird die Flotte der Auslieferungsfahrzeuge als homogen angenommen, d.h. jedes Fahrzeug v der Fahrzeugmenge \mathcal{V} hat die gleiche Ladekapazität q .

Auch wenn dies zunächst nicht so erscheinen mag, ist die Annahme gleicher Fahrzeuge durchaus realistisch, da es für ein Unternehmen in betriebswirtschaftlicher und praktischer Hinsicht, bspw. im Hinblick auf Einkaufspreisrabatte, Kosten der Ersatzteilbevorratung oder der Fahrzeuginneneinrichtung, sinnvoll ist, einen homogenen Fuhrpark zu unterhalten.

In das Modell fließt lediglich der Aspekt der Ladekapazität ein, deren Einhaltung durch Gegenüberstellung der summierten Kundenbedarfe d_i und der Fahrzeugkapazität q durch

$$\sum_{i \in \mathcal{C}} d_i \sum_{j \in \mathcal{N}} x_{ijk} \leq q \quad \forall k \in \mathcal{V} \quad (5.7)$$

gewährleistet wird.

Die mit dem Einsatz eines Fahrzeugs verbundenen Fixkosten in Höhe von f fallen an, sobald ein Fahrzeug vom Ausgangsdepot zu einem Kundenknoten fährt. Dies wird durch die Bedingung

$$\sum_{j \in \mathcal{C}} x_{0jk} = y_k \quad \forall k \in \mathcal{V} \quad (5.8)$$

ausgedrückt. Bei Gleichung (5.8) handelt es sich nur um eine abkürzende Schreibweise, da an jeder Stelle des Modells, in der auf y_k Bezug genommen wird, stattdessen auch die Summe auf der linken Seite von (5.8) verwendet werden könnte.

5.1.4 Zeitfensterrestriktionen

Neben der durch t_{ij} bezeichneten Reisezeit, wird zur Abbildung der Zeitfensterrestriktionen als weitere Größe die Standzeit τ_s benötigt. Sie drückt die Dauer eines Kundenbesuchs aus und umfaßt bspw. Zeiten der Entladung oder der Übergabe der Sendung. Zur Vereinfachung wird diese Zeit als identisch für alle Kunden angenommen.

Die Auslieferung an einem Lieferort i kann frühestens zum Zeitpunkt b_i stattfinden und muß bis zum Zeitpunkt e_i begonnen (aber nicht notwendigerweise auch abgeschlossen) werden. Die Zeitfenster $[b_0, e_0]$ und $[b_{n+1}, e_{n+1}]$ des Ausgangs- bzw. Enddepots werden so gewählt, daß sie die früheste Abfahrtsmöglichkeit und die spätestmögliche Rückkehr wiedergeben.

Die Einhaltung der Zeitfenster wird durch eine einfache zweiseitige Ungleichung

$$b_j \leq s_{jk} \leq e_j \quad \forall j \in \mathcal{N}, k \in \mathcal{V} \quad (5.9)$$

gewährleistet. Der Servicezeitpunkt ist abhängig von der Reihenfolge der Kundenbesuche auf einer Route entsprechend

$$s_{jk} \geq s_{ik} + \tau_s + t_{ij}, \quad (5.10)$$

wobei die „ \geq “-Beziehung notwendig ist, um den Fall abzudecken, daß

$$s_{ik} + \tau_s + t_{ij} < b_j \quad (5.11)$$

ist, d.h. das Fahrzeug vor dem Öffnungszeitpunkt b_j beim Kunden j ankommt. In diesem Fall ist es erlaubt, bis zum Öffnungszeitpunkt zu warten. Man kann die Zeitfensterrestriktion in dieser Form daher auch als *weiche* Zeitfensterrestriktion bezeichnen, im Gegensatz zu einer *harten* Restriktion, die eine Ankunft nur im Öffnungszeitraum erlaubt und aus der Gleichheit in (5.10) im Zusammenspiel mit (5.9) resultiert.

Da jeder Kunde i genau einmal besucht wird, ist die Variable s_{ik} nur für ein Fahrzeug k' aussagekräftig und für alle anderen Fahrzeuge bedeutungslos. Den Variablen $s_{ik}, k \in \mathcal{V} \setminus \{k'\}$ müssen im Modell jedoch Werte zugewiesen werden, die der Bedingung (5.9) nicht widersprechen. Gesucht wird daher eine Formulierung, die beide Mengen von Entscheidungsvariablen x_{ijk} und s_{ik} zueinander in Beziehung setzt. Eine Implikation der Form

$$x_{ijk} = 1 \implies s_{jk} \geq s_{ik} + \tau_s + t_{ij} \quad (5.12)$$

ist jedoch in einem MIP nicht unmittelbar verwendbar, da es das Konstrukt der Implikation nicht gibt. Es kann jedoch durch eine sog. *Big-M*-Formulierung (s. z.B. (Williams 1999)) nachgebildet werden:

$$s_{ik} + \tau_s + t_{ij} - M(1 - x_{ijk}) \leq s_{jk} \quad \forall i, j \in \mathcal{N}, \forall k \in \mathcal{V} \quad (5.13)$$

Zum Verständnis können die zwei möglichen Fälle $x_{ijk} = 0$ und $x_{ijk} = 1$ unterschieden werden. Im ersten Fall reduziert sich die Ungleichung zu

$$s_{ik} + \tau_s + t_{ij} - M \leq s_{jk}. \quad (5.14)$$

Da M voraussetzungsgemäß eine sehr große Zahl ist, ist (5.14) trivialerweise erfüllt. Im anderen Fall lautet die Ungleichung

$$s_{ik} + \tau_s + t_{ij} \leq s_{jk}. \quad (5.15)$$

und entspricht damit genau dem gewünschten Ausdruck (5.10).

Das Beispiel der Zeitfenster zeigt die begrenzten Möglichkeiten der Modellierung mit binären Entscheidungsvariablen und linearen Ungleichungen: Zwar steht am Ende der Modellierung eine gültige Formulierung der Bedingung, diese ist jedoch alles andere als elegant und – wesentlich kritischer – interferiert mit dem Lösungsprozeß, da bekannt ist (vgl. (Williams 1999)), daß die Wahl von M im Rahmen der Relaxation des MIP Einfluß auf die Lösungseffizienz hat: M sollte so klein wie möglich und so groß wie nötig gewählt werden, um die Relaxation nicht unnötig zu schwächen.

5.1.5 MIP Formulierung des VRPTW

Die vorangegangenen Ungleichungen lassen sich zu einem gemischt-ganzzahligen Programm zusammenstellen. Die Zielfunktion besteht darin, die gesamten Fahrtkosten

zu minimieren, die sich als Summe der variablen und der einsatzfixen Kosten ergeben.

$$\min \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{V}} c_{ij} x_{ijk} + \sum_{k \in \mathcal{V}} f y_k \quad (5.16)$$

$$\sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{V}} x_{ijk} = 1 \quad \forall i \in \mathcal{C} \quad (5.17)$$

$$\sum_{i \in \mathcal{C}} d_i \sum_{j \in \mathcal{N}} x_{ijk} \leq q \quad \forall k \in \mathcal{V} \quad (5.18)$$

$$\sum_{j \in \mathcal{C}} x_{0jk} = y_k \quad \forall k \in \mathcal{V} \quad (5.19)$$

$$(VRPTW) \quad \sum_{j \in \mathcal{N}} x_{0jk} = 1 \quad \forall k \in \mathcal{V} \quad (5.20)$$

$$\sum_{i \in \mathcal{N}} x_{ihk} - \sum_{j \in \mathcal{N}} x_{hjk} = 0 \quad \forall h \in \mathcal{C}, \forall k \in \mathcal{V} \quad (5.21)$$

$$\sum_{i \in \mathcal{N}} x_{i,n+1,k} = 1 \quad \forall k \in \mathcal{V} \quad (5.22)$$

$$s_{ik} + \tau_s + t_{ij} - M(1 - x_{ijk}) \leq s_{jk} \quad \forall i, j \in \mathcal{N}, \forall k \in \mathcal{V} \quad (5.23)$$

$$b_i \leq s_{ik} \leq e_i \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{V} \quad (5.24)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j \in \mathcal{N}, \forall k \in \mathcal{V}. \quad (5.25)$$

5.2 Das VRPTW als allgemeines Routingproblem

Das Modell (VRPTW) in der obigen Formulierung umfaßt bereits einige Spezialfälle (siehe auch (Kohl 1995; Larsen 1999)).

Sind die Zeitfensterrestriktionen nicht bindend, d.h. sind $b_i = 0$ und $e_i = M$ (wobei M eine hinreichend große Zahl ist), entspricht das Modell dem klassischen Tourenplanungsproblem VRP.

Das Handlungsreisendenproblem (*Traveling Salesman Problem*, TSP) erhält man, wenn nur ein Fahrzeug ($|\mathcal{V}| = 1$) und keine Kapazitäts- oder Zeitfensterrestriktionen betrachtet werden. Läßt man eine Flotte von m Fahrzeugen zu, erhält man das m -TSP, bei dem eine Gruppe von m Handlungsreisenden alle Knoten kostenminimal besucht. Das *Bin-Packing*-Problem erhält man wenn

$$c_{ij} = \begin{cases} 1 & i = 0, j \in \mathcal{C} \\ 0 & \text{sonst,} \end{cases}$$

d.h. die Zielfunktion darin besteht, die Zahl der eingesetzten Fahrzeuge zu minimieren. Ist die Zahl $|\mathcal{V}|$ größer als erforderlich, ist die Flottengröße nicht bindend. Für die nicht eingesetzten Fahrzeuge k' wird in der Lösung $x_{0,n+1,k'} = 1$ gesetzt, d.h. einem

nicht eingesetzten Fahrzeug entspricht im Modell eine direkte Fahrt vom Ausgangs- zum Enddepot. Für die korrekte Abbildung von Optimierungsproblemen mit freier Fahrzeugzahl müssen zusätzlich noch die Kosten für den Nichteinsatz der Fahrzeuge $c_{0,n+1} = 0$ und der entsprechenden Reisezeiten $t_{0,n+1} = 0$ gesetzt werden. Legt man die Flottengröße auf $|\mathcal{V}'|$ fest, d.h. wird der Einsatz aller Fahrzeuge der Flotte gefordert, setzt man umgekehrt $c_{0,n+1}$ bzw. $t_{0,n+1}$ auf einen sehr großen Wert und erzwingt damit, da ein Kostenminimum gesucht wird, den Fahrzeugeinsatz.

Die Varianten des Handlungsreisendenproblems bzw. des m -TSP mit Zeitfenstern, TSPTW bzw. m -TSPTW genannt, ergeben sich durch erneute Hinzunahme der Zeitfensterbedingungen.

Das VRPTW stellt sich nach diesen Ausführungen bereits als ein recht allgemeines Problem dar. Dennoch gibt es einige Erweiterungen und Anwendungsfälle, die mit diesem Modell nicht abgedeckt werden können. Ein wichtiger Fall ist das Tourenplanungsproblem mit *Pickup & Delivery* Transporten, bei dem Waren von einem Ort an einen anderen Ort transportiert werden müssen, ohne zwischenzeitlich ein Depot zu besuchen. Modelle für dieses Problem werden in (Sol 1994) und (Toth und Vigo 2002) angegeben. Ebenfalls eine wichtige Erweiterung ist das Tourenplanungsproblem mit Rückfrachten (*VRP with backhauls*, vgl. (Toth und Vigo 2002)), bei dem neben Lieferungen auch Abholungen stattfinden. Modellerweiterungen im Hinblick auf die Abbildung von Zeitfenstern sind in der Literatur ebenfalls mehrfach untersucht worden (für eine Übersicht siehe (Toth und Vigo 2002)). Betrachtet wurden u.a. unscharfe Zeitfenster, deren Verletzung über Kosten graduell bestraft wird, und mehrfache Zeitfenster pro Kunden. Das in dieser Arbeit untersuchte Problem der alternativen Liefermöglichkeiten stellt eine Verallgemeinerung des Problems mit Mehrfachzeitfenstern dar und ist bisher noch nicht wissenschaftlich untersucht worden. Ein entsprechendes Modell wird im nächsten Abschnitt entwickelt.

5.3 Erweiterung um alternative Liefermöglichkeiten

Für die Erweiterung des Standardmodells um alternative Liefermöglichkeiten ist eine Differenzierung zwischen kunden- und knotenbezogenen Bedingungen erforderlich. Letztere dienen der Unterscheidung alternativer Lieferorte, während die erstgenannten diejenigen Aspekte abdecken, die davon unabhängig sind. Dies ist insbesondere der Kundenbedarf d_i , der für alle alternativen Lieferorte eines Kunden als gleich angenommen wird.

Die Modellierung erfordert eine logische Trennung zwischen der Menge der Kunden \mathcal{C} und der Menge der Knoten \mathcal{N} . Für die Kundenmenge \mathcal{C} gilt weiterhin $\mathcal{C} = \{1, \dots, n\}$, die Knotenmenge \mathcal{N} wird verändert zu $\mathcal{N} = \{0, 1, \dots, n, n+1, \dots, n+k = m, m+1\}$. Die Elemente dieser Menge bezeichnen Knoten im dazugehörigen modifizierten Netzwerk, wobei die Knoten 0 und $m+1$ die neuen Bezeichnungen des Ausgangs-

bzw. Enddepots sind. Die Menge $\mathcal{N}_C = \{1, \dots, m\} = \mathcal{N} \setminus \{0, m+1\}$ bezeichnet die zu Kunden gehörigen Knoten.

Datengrundlage des Modells sind die Informationen über die von einem Kunden angebotenen alternativen Lieferorte und -zeitfenster. Mit jedem Kunden i wird die Alternativenmenge \mathcal{A}_i assoziiert, die die Knoten der alternativen Liefermöglichkeiten des Kunden i umfaßt. Für eine konsistente Notation werden den Depots die Alternativenmengen $\mathcal{A}_0 = \{0\}$ bzw. $\mathcal{A}_{m+1} = \{m+1\}$ zugeordnet.

Für die Alternativenmengen gilt

$$\mathcal{A}_p \cap \mathcal{A}_q = \emptyset, \quad \forall p, q \in C, p \neq q \text{ und } \bigcup_{i \in C} \mathcal{A}_i = \mathcal{N}_C,$$

d.h. die Alternativenmengen sind paarweise disjunkt und ihre Vereinigung entspricht der Knotenmenge ohne Depotknoten. Damit stellen die Alternativenmengen die eindeutige Verbindung zwischen Kunden- und Knotenindizes dar. Dies vereinfacht die Notation im Modell, da Knotenindizes ohne Angabe des Kundenindex verwendet werden können. Beispielsweise ist es ausreichend, die Zeitfenster anhand der Knoten zu indizieren, um die unterschiedlichen Zeitfenster der alternativen Liefermöglichkeiten der Kunden anzugeben. Ebenfalls knotenbezogen indiziert werden die Entscheidungsvariablen x_{ijk} bzw. s_{ik} .

Zentrale Modellerweiterung ist die Bedingung der Auswahl genau eines Knotens aus der Alternativenmenge jedes Kunden. Mit den eingeführten Bezeichnern kann dies über die Bedingung

$$\sum_{p \in \mathcal{A}_i} \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{V}} x_{pjk} = 1 \quad \forall i \in C \tag{5.26}$$

ausgedrückt werden, durch die festgelegt wird, daß von allen Knoten aus der Alternativenmenge eines Kunden i insgesamt nur genau eine Fahrt ausgehen darf.

5.3.1 MIP Formulierung des Tourenplanungsproblems mit alternativen Liefermöglichkeiten

Mit den im vorangegangenen Abschnitt eingeführten Erweiterungen des Standardmodells kann jetzt eine Formulierung des Tourenplanungsproblems mit alternativen Lie-

fermöglichkeiten VRPA² angegeben werden.

$$\min \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{V}} c_{ij} x_{ijk} + \sum_{k \in \mathcal{V}} f y_k \quad (5.27)$$

$$\sum_{p \in \mathcal{A}_i} \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{V}} x_{pjk} = 1 \quad \forall i \in \mathcal{C} \quad (5.28)$$

$$\sum_{i \in \mathcal{C}} d_i \sum_{j \in \mathcal{N}} \sum_{p \in \mathcal{A}_i} x_{pjk} \leq q \quad \forall k \in \mathcal{V} \quad (5.29)$$

$$\sum_{j \in \mathcal{N}_C} x_{0jk} = y_k \quad \forall k \in \mathcal{V} \quad (5.30)$$

(VRPA)

$$\sum_{j \in \mathcal{N}} x_{0jk} = 1 \quad \forall k \in \mathcal{V} \quad (5.31)$$

$$\sum_{i \in \mathcal{N}} x_{ihk} - \sum_{j \in \mathcal{N}} x_{hjk} = 0 \quad \forall h \in \mathcal{N}_C, \forall k \in \mathcal{V} \quad (5.32)$$

$$\sum_{i \in \mathcal{N}} x_{i,m+1,k} = 1 \quad \forall k \in \mathcal{V} \quad (5.33)$$

$$s_{ik} + \tau_s + t_{ij} - M(1 - x_{ijk}) \leq s_{jk} \quad \forall i, j \in \mathcal{N}, \forall k \in \mathcal{V} \quad (5.34)$$

$$b_i \leq s_{ik} \leq e_i \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{V} \quad (5.35)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j \in \mathcal{N}, \forall k \in \mathcal{V}. \quad (5.36)$$

5.4 Das Dekompositionsmodell

Ein gemischt-ganzzahliges Optimierungsproblem wie (VRPA) ist bereits für kleine Problemumfänge nicht mit vertretbarem Aufwand zu lösen. Dekompositionsansätze bieten sich stattdessen bei geeigneten Problemstrukturen an.

Im vorliegenden Fall des (VRPA) ergibt sich durch die folgende Beobachtung eine Möglichkeit der Dekomposition: Lediglich (5.28) setzt die Fahrzeuge miteinander in Beziehung, während alle anderen Nebenbedingungen keine Wechselwirkung zwischen Fahrzeugen aufweisen.

Diese Beobachtung wird dazu verwendet, das Optimierungsproblem mittels der Dantzig-Wolfe-Dekomposition in ein Master- und ein Subproblem zu zerlegen, um es daraufhin mit der Methode der Spaltenerzeugung zu lösen.

²Es hat sich in der Literatur eingebürgert, ein Optimierungsproblem – und insbesondere die Erweiterung von bekannten Problemen – nach der jeweils neuen bzw. restriktivsten Anforderung zu benennen. Daher wird dieses Optimierungsproblem mit VRPA für *vehicle routing problem with alternatives* bezeichnet, anstelle des sperrigen Kürzels VRPCCTWA für *vehicle routing problem with capacity constraints, time windows, and alternatives*.

5.4.1 Das Master-Problem

Das Master-Problem (MP) wird als Set Partitioning-Problem formuliert. Ziel dieser Problemklasse ist die kostenminimale Aufteilung von m Objekten auf eine Gruppe von Teilmengen, wobei jedes Objekt genau einmal ausgewählt werden muß. Die Formulierung lautet

$$(SP) \quad \min c^T x \quad (5.37)$$

$$Ax = 1 \quad (5.38)$$

$$x_i \in \{0, 1\} \quad \forall i \in \{1, \dots, n\}, \quad (5.39)$$

wobei A eine $m \times n$ -Matrix mit Einträgen $a_{ij} \in \{0, 1\}$ und c und x n -dimensionale Vektoren sind. Ein Matrixkoeffizient $a_{ij} = 1$ bedeutet dabei, daß das i -te Objekt in der j -ten Gruppe enthalten ist, der Kostenkoeffizient c_j gibt die Kosten der j -ten Aufteilung an. Übertragen auf das Problem der Tourenplanung entsprechen die Zeilen der Matrix A den Kunden und die Spalten den einzelnen Routen. Ein Wert $x_r = 1$ in einer Lösung besagt, daß die r -te Route, deren Kosten inkl. der einsatzfixen Kosten f c_r betragen, für den Tourenplan ausgewählt wird. Anzumerken ist, daß zuweilen auch Matrixeinträge $a_{ir} \geq 2$ zugelassen werden und dann besagen, daß der Kunde i auf Route r a_{ir} Mal besucht wird. Dies ist jedoch zumeist nicht erwünscht und auch in dieser Arbeit wird von $a_{ir} \in \{0, 1\}$ ausgegangen.

Das Masterprogramm wird jedoch nicht als reines Set Partitioning-Problem ausgeführt, sondern um zusätzliche Bedingungen für die Mindest- und Höchstzahl von Fahrzeugen ergänzt. Bezeichnet man die Menge der Spalten und damit die Menge der Routen mit \mathcal{R} , haben diese Bedingungen die Form

$$\sum_{r \in \mathcal{R}} x_r \geq \underline{v} \quad (5.40)$$

bzw.

$$\sum_{r \in \mathcal{R}} x_r \leq \bar{v}, \quad (5.41)$$

wobei \underline{v} bzw. \bar{v} der Mindest- bzw. Höchstzahl an einsetzbaren Fahrzeugen entsprechen. Bei einer freien Fahrzeugzahl und einer Optimierung der Fahrt- und einsatzfixen Kosten sind die Beschränkungen der Flottengröße unter Modellierungsgesichtspunkten nicht zwingend erforderlich. Technisch gesehen handelt es sich bei beiden Ungleichungen (5.40) und (5.41) um praktisch aufwandslos zu bestimmende Beschränkungen des Lösungsraumes, die bereits überraschend effektiv sind (vgl. (Kohl 1995)) und daher in das Modell des Master-Problems aufgenommen werden. Die Höchstzahl von

Fahrzeugen ist üblicherweise durch die reale Größe des Fuhrparks gegeben, während sich die untere Grenze für die Fahrzeuganzahl durch die einfache Berechnung

$$\underline{v} = \left\lceil \frac{\sum_{i \in C} d_i}{q} \right\rceil \quad (5.42)$$

der aufgrund der gesamten Bedarfsmenge mindestens benötigten Fahrzeuganzahl bestimmen läßt.

Insgesamt lautet die Set Partitioning-Formulierung des Master-Problems mit zusätzlichen Flottenbeschränkungen damit:

$$\min \sum_{r \in \mathcal{R}} c_r x_r \quad (5.43)$$

$$\sum_{r \in \mathcal{R}} a_{ir} x_r = 1 \quad \forall i \in C \quad (5.44)$$

$$(MP) \quad \sum_{r \in \mathcal{R}} x_r \geq \underline{v} \quad (5.45)$$

$$\sum_{r \in \mathcal{R}} x_r \leq \bar{v} \quad (5.46)$$

$$x_r \in \{0, 1\} \quad \forall r \in \mathcal{R}. \quad (5.47)$$

Im Modell (MP) wird davon ausgegangen, daß die Menge aller Routen \mathcal{R} bekannt ist und die Optimierung daher über alle Routen erfolgen kann. Die Methode der Spalten-erzeugung arbeitet aber wie zuvor erläutert auf einer Teilmenge aller Routen und löst dazu das Restricted Master-Problem, auf das im folgenden Unterabschnitt eingegan-gen wird.

Das Restricted Master-Problem

Das Verfahren des Column Generation beruht auf der Idee, Variablen dynamisch nach Bedarf zu generieren und nicht von Beginn an mit der vollständigen Variablenmenge zu arbeiten. Im Fall der Tourenplanung bedeutet dies, daß nicht sämtliche Routen $r \in \mathcal{R}$, sondern nur jene einer speziell erzeugten beschränkten Routenmenge \mathcal{R}' betrachtet werden. Darüberhinaus wird kein gemischt-ganzzahliges Problem gelöst, sondern eine LP-Relaxation, d.h. die Ganzzahligkeitsbedingungen werden fallengelassen. Das RMP

lautet daher

$$\min \sum_{r \in \mathcal{R}'} c_r x_r \quad (5.48)$$

$$\sum_{r \in \mathcal{R}'} a_{ir} x_r = 1 \quad \forall i \in \mathcal{C} \quad (5.49)$$

$$\text{(RMP)} \quad \sum_{r \in \mathcal{R}'} x_r \geq \underline{v} \quad (5.50)$$

$$\sum_{r \in \mathcal{R}'} x_r \leq \bar{v} \quad (5.51)$$

$$0 \leq x_r \leq 1 \quad \forall r \in \mathcal{R}'. \quad (5.52)$$

Die Lösung des (RMP) liefert duale Variablen $u_i, i \in \mathcal{C}$ für die Gleichungen (5.49), sowie u_0 für (5.50) und v_0 für (5.51). Daraus ergeben sich die reduzierten Kosten einer Route r zu

$$\hat{c}_r = c_r - \sum_{i=1}^{|\mathcal{C}|} a_{ir} u_i - u_0 - v_0. \quad (5.53)$$

Die Formulierungen sowohl des Master- als auch des Restricted Masterproblems (MP) bzw. (RMP) beziehen sich in den Gleichungen (5.44) und (5.49) auf Kunden und nicht auf die alternativen Liefermöglichkeiten dieser. Aus der Formulierung ist demnach nicht unmittelbar zu entnehmen, welche alternative Liefermöglichkeit eines Kunden besucht wird.

Obwohl nicht explizit im (Restricted) Master-Problem enthalten, ist diese Information dennoch greifbar, da sie auf der Ebene des im folgenden Abschnitt vorgestellten Pricing-Problems bei der Routenbestimmung erzeugt wird. Neben den ausgewählten Alternativen gilt dies im Übrigen analog auch für die Reihenfolgeinformation der Kunden- bzw. Knotenbesuche. Im Rahmen der Implementierung des Verfahrens muß lediglich durch geeignete Datenstrukturen und Schnittstellen sichergestellt werden, daß diese Information für eine vollständige Beschreibung einer Lösung abrufbar ist.

5.4.2 Das Pricing-Problem – Ein Constraint Programming-Modell

Aufgabe des Pricing-Problems ist die Bestimmung kürzester Wege unter Berücksichtigung von Ressourcenbeschränkungen und weiteren Nebenbedingungen. Dazu gehören neben Kapazitäts- und Zeitfensterrestriktionen auch die Auswahl der alternativen Lieferorte. Weiterhin müssen die errechneten Wege negative reduzierte Kosten aufweisen, um einen verbessernden Beitrag zum Restricted Master-Problem zu leisten.

In diesem Abschnitt wird ein Constraint Programming-Modell zur Lösung des Kürzeste-Wege-Problems mit Ressourcenbeschränkungen formuliert. In der Literatur

wird dieses Problem als *Elementary Shortest Path Problem with Resource Constraints* (ESPPRC) bezeichnet. Das hier vorgestellte Constraint Programming Modell führt die Formulierungen von (Pesant, Gendreau, Potvin und Rousseau 1998; Focacci, Lodi, Milano und Vigo 1998) und (Rousseau, Gendreau und Pesant 2002) zusammen und erweitert sie um alternative Liefermöglichkeiten.

Es bietet sich an, verschiedene Bezeichner zur Abkürzung einzuführen. Mit $\mathcal{N}_O = \{0, 1, 2, \dots, m\}$ werden Knoten bezeichnet, die von einem Fahrzeug verlassen werden können, mit $\mathcal{N}_D = \{1, 2, \dots, m, m+1\}$ solche, an denen ein Fahrzeug ankommen kann. Beide Mengen unterscheiden sich offensichtlich nur durch das jeweils enthaltene Depot. Weiterhin wird eine vereinfachte Schreibweise für die Beschreibung der Domains der Variablen und deren Instanziierung verwendet: Um den Wertebereich einer Variablen X anzugeben, wird anstelle der Domainschreibweise $D_X = \{v_0, v_1, \dots, v_k\}$ kürzer $X \in \{v_0, v_1, \dots, v_k\}$ geschrieben. Eine Instanziierung von X mit dem Wert v wird durch $X = v$ ausgedrückt.

Kernelement der Modellierung ist die Wahl der Entscheidungsvariablen. Gegenüber den Variablen x_{ijk} der kompakten MIP-Formulierungen (VRPTW) bzw. (VRPA) entfällt für die Routenbestimmung die Zuordnungskomponente. Stattdessen wird die Route über die Angabe der Nachfolgerbeziehungen beschrieben: Für jeden Knoten $i \in \mathcal{N}_O$ bezeichnet Variable S_i den Nachfolgeknoten. Für ihren Wertebereich gilt

$$S_i \in \{1, 2, \dots, m, m+1\} = \mathcal{N}_D, \forall i \in \mathcal{N}_O. \quad (5.54)$$

Um auszudrücken, daß ein Knoten i nicht auf einer Route vom Ausgangs- zum Enddepot besucht wird, wird er sich selbst als Nachfolger zugewiesen, d.h. $S_i = i$ gesetzt. Solange daher i im Wertebereich von S_i ist, d.h. $i \in S_i$ gilt, ist nicht festgelegt, ob der Knoten i auf der aktuell untersuchten Route besucht wird oder nicht.

Neben der Sequenzvariable S_i werden weitere Variablen zur Formulierung der Constraints benötigt. Konkret werden eine kumulierte Kostenvariable C_i , eine Kapazitätsvariable Q_i und eine Servicezeitvariable T_i (wobei jeweils $i \in \mathcal{N}$ ist) benötigt. Die Kostenvariable C_i gibt die kumulierten Kosten bei Ankunft im Knoten i an und die Kapazitätsvariable Q_i den kumulierten Bedarf bei Erreichen des Kunden i (ohne dessen Bedarf d_i). Die Servicezeitvariable T_i gibt analog zu Variable s_{ik} aus den kompakten MIP-Modellen (VRPTW) und (VRPA) den Auslieferzeitpunkt beim Kunden i an.

Die Zielfunktion, d.h. die Minimierung der Routenkosten, kann über die einfache Funktion

$$\min C_{m+1} \quad (5.55)$$

angegeben werden. Die Kosten einer Route ergeben sich aufgrund der folgenden Im-

plikation, die die Kostenvariable mit den Sequenzvariablen verknüpft:

$$S_i = j \implies C_j = C_i + \hat{c}_{ij} = C_i + c_{ij} - u'_i, \quad \forall i \in \mathcal{N}_O, j \in \mathcal{N}_D, i \neq j. \quad (5.56)$$

Zu beachten ist hierbei, daß im Pricing Problem die reduzierten Kosten \hat{c}_{ij} der Kanten $\langle i, j \rangle$ verwendet werden. Die Größe u'_i ist gleich der Dualvariablen u_k wenn i aus der Alternativenmenge \mathcal{A}_k des Kunden k ist, d.h.

$$u'_i = u_k \text{ für } k \in \mathcal{C}, i \in \mathcal{A}_k. \quad (5.57)$$

Durch diese Festlegung werden die Dualvariablen des auf Kunden definierten (RMP) den jeweils zugehörigen alternativen Liefermöglichkeiten zugewiesen. Für $i = 0$ wird $u'_i = u_0 + v_0$ gesetzt, d.h. die mit der unteren bzw. oberen Schranke der Fahrzeuganzahl verbundenen Dualvariablen werden dem Ausgangsdepot zugerechnet.

Die in (5.56) verwendete Implikation ist eine elegante Konstruktion, um die Abhängigkeit der eigentlichen Wertberechnung von der Reihenfolge der Besuche abzubilden und wird ebenfalls für die Kumulierung der Kapazitäts- und Servicezeitpunktvariablen verwendet:

$$S_i = j \implies Q_j = Q_i + d_i, \quad \forall i \in \mathcal{N}_O, j \in \mathcal{N}_D, i \neq j, \quad (5.58)$$

$$S_i = j \implies T_j \geq T_i + \tau_s + t_{ij}, \quad \forall i \in \mathcal{N}_O, j \in \mathcal{N}_D, i \neq j. \quad (5.59)$$

Die Kumulierung von Bedarfsmengen und Reisezeiten erfordert eine Initialisierung der dem Startdepot zugeordneten Variablen mit

$$Q_0 = 0, T_0 = b_0 \text{ bzw. } C_0 = 0. \quad (5.60)$$

Die Begrenzung der Kapazität erfolgt über eine Schranke

$$Q_{m+1} \leq q, \quad (5.61)$$

die Einhaltung der Zeitfenster wird über die Restriktion

$$b_i \leq T_i \leq e_i \quad \forall i \in \mathcal{N} \quad (5.62)$$

ausgedrückt. Der gegenseitige Ausschluß von alternativen Liefermöglichkeiten eines Kunden ist naturgemäß die Kernforderung dieses Modells und wird über Metaconstraints realisiert. Damit kann die Bedingung an die Auswahl maximal eines Lieferortes pro Kunde durch

$$\sum_{j \in \mathcal{A}_i} (S_j \neq j) \leq 1 \quad \forall i \in \mathcal{C} \quad (5.63)$$

beschrieben werden: Falls der Nachfolger S_j des alternativen Lieferortes j ein anderer Knoten $k \neq j$ ist, ist $S_j \neq j$ erfüllt und hat den Wahrheitswert 1. Daraus folgt durch die Propagation unmittelbar, daß für alle anderen $j' \in \mathcal{A}_i, j \neq j'$ $S_{j'} = j'$ sein muß. Umgekehrt ist es zulässig, für alle $j \in \mathcal{A}_i$ $S_j = j$ zu setzen, d.h. alle Alternativen des Kunden i aus der Route zu entfernen und den Kunden i somit auf der betrachteten Route nicht zu besuchen.

Die path-Constraint

Die bisher aufgeführten Bedingungen allein garantieren nicht, daß eine gültige Route berechnet wird. Insbesondere sind Zyklen und mehrfache Anfahrten (bei einer Abfahrt) mögliche Ergebnisse einer Route.

Um einen gültigen Pfad zu beschreiben, bietet es sich an, der Sequenzvariable S_i eine Vorgängervariable P_i gegenüberzustellen, für die

$$P_i \in \{0, 1, 2, \dots, m\} = \mathcal{N}_O, \forall i \in \mathcal{N}_D \quad (5.64)$$

gilt und die mit S_i über

$$S_i = j \Leftrightarrow P_j = i, \quad \forall i \in \mathcal{N}_O, j \in \mathcal{N}_D \quad (5.65)$$

verknüpft ist.

Ein gültiger Pfad zeichnet sich dadurch aus, daß

1. vom Ausgangsdepot ein Kundenknoten besucht wird: $S_0 \in \mathcal{N}_C$,
2. es keine Subtouren gibt, d.h. keine Knotenmenge $\mathcal{Z} = \{z_1, z_2, \dots, z_{|\mathcal{Z}|}\}$ mit $S_{z_1} = z_2, S_{z_2} = z_3, \dots, S_{z_{|\mathcal{Z}|}} = z_1$ und $|\mathcal{Z}| \geq 2$ existiert.

Die genannten Bedingungen werden in einer globalen Constraint

$$\text{path}(S, P, 0, m + 1)$$

gekapselt. Im Hinblick auf die Lösungssuche erlaubt eine solche Kapselung die Anpassung spezieller Propagationsalgorithmen auf die Constraint und kann damit effizienter behandelt werden als die Teilconstraints einzeln. Die globale Constraint `path` sichert die Existenz eines Pfades von Knoten 0 nach $m + 1$ und arbeitet auf dem Vektor von Nachfolgervariablen S sowie dem Vektor von Vorgängervariablen P .

Die in dieser Arbeit verwendete `path`-Constraint unterscheidet sich von der in der Literatur verwendeten Form (s. bspw. (Hooker 2000; Focacci u. a. 2004)). Bei diesen wird ein Pfad gefordert, der jeden Knoten genau einmal besucht, d.h. durch die Constraint wird ein *Hamiltonscher Pfad* bestimmt. Die hier vorgestellte `path`-Constraint ist insofern komplexer, als daß zusätzlich zur Reihenfolge noch eine Auswahlentscheidung erforderlich ist.

Constraint Programming-Modell

Die im vorangegangenen Abschnitt aufgeführten Bedingungen ergeben zusammen das folgende Constraint Programming-Modell der Suche kürzester Wege unter Berücksichtigung von Kapazitäten, Zeitfenstern und alternativen Liefermöglichkeiten.

$$\begin{aligned} & \min C_{m+1} && (5.66) \\ S_i = j & \implies C_j = C_i + \hat{c}_{ij}, \quad \forall i \in \mathcal{N}_O, \forall j \in \mathcal{N}_D && (5.67) \\ & C_0 = 0 && (5.68) \\ S_i = j & \implies Q_j = Q_i + d_i, \quad \forall i \in \mathcal{N}_O, \forall j \in \mathcal{N}_D && (5.69) \\ & Q_0 = 0 && (5.70) \\ \text{(CPPP)} & & Q_{m+1} \leq q && (5.71) \\ & & T_0 = b_0 && (5.72) \\ S_i = j & \implies T_j \geq T_i + \tau_s + t_{ij} \quad \forall i \in \mathcal{N}_O, \forall j \in \mathcal{N}_D, i \neq j && (5.73) \\ & & b_i \leq T_i \leq e_i \quad \forall i \in \mathcal{N} && (5.74) \\ & & \sum_{j \in \mathcal{A}_i} (S_j \neq j) \leq 1 \quad \forall i \in \mathcal{N}_C && (5.75) \\ & & \text{path}(S, P, 0, m + 1) && (5.76) \end{aligned}$$

5.5 Weitere Anwendungsfälle des Modells

Die mathematische Modellierung verschiedener Probleme erlaubt es oftmals, gemeinsame Strukturen zu entdecken und die gleichen Lösungsalgorithmen anzuwenden. In der Tat zeigt sich, daß das Tourenplanungsproblem mit alternativen Liefermöglichkeiten einerseits eine Verallgemeinerung bekannter Probleme ist und andererseits auf weitere interessante Fragestellungen angewandt werden kann. Im Folgenden werden

mögliche Anwendungsfälle vorgestellt, indem beschrieben wird, inwiefern diese Probleme als Instanzen des VRPA gesehen werden können.

5.5.1 Mehrfachzeitfenster

Das Problem der Mehrfachzeitfenster ist in (Pesant u. a. 1999) bereits mittels Constraint Programming formuliert worden und ist ein Spezialfall des Tourenplanungsproblems mit alternativen Liefermöglichkeiten. Die Überführung einer Instanz des Problems mit Mehrfachzeitfenstern P_{MZF} in ein VRPA ist unmittelbar in zwei Schritten möglich, indem

1. jedem Zeitfenster eines Kunden im P_{MZF} eine alternative Liefermöglichkeit zugeordnet wird,
2. die örtliche Lage aller Alternativen eines Kunden identisch und gleich der Kundenlage im P_{MZF} ist.

5.5.2 Einfache intermodale Transporte: Abbildung von Fahrplänen für Shuttle-Züge

Intermodale Transporte mit Shuttle-Zügen, d.h. Güterzügen, die aufgrund eines regelmäßigen Fahrplans verkehren, sind angesichts des steigenden Güterverkehrsaufkommens unter Umständen eine Alternative zum reinen Straßentransport.

Die Abwicklung dieser Transporte kann sich folgendermaßen gestalten: An einem Ausgangsbahnhof fahren LKW auf den Zug auf³, um ihn am Zielbahnhof wieder zu verlassen. Die Züge fahren gemäß eines Fahrplans und verlangen demnach eine pünktliche Ankunft des LKW am Abfahrtsbahnhof. Die integrierte intermodale Tourenplanung muß neben den üblichen Nebenbedingungen zusätzlich die Fahrpläne der Shuttle-Züge berücksichtigen.

Um Bahnstrecken im Modell einzubeziehen, werden Bahnhöfe und Umschlagzentren als Knoten abgebildet, die durch Kanten entsprechend dem Schienennetz verbunden sind. Diese Kanten werden analog zu Straßenverbindungen mit Kosten und Reisezeiten versehen. Die Abfahrtszeiten bzw. Ankunftszeiten der Züge werden durch halboffene Zeitfenster der Form $(-\infty, t_{ab}]$ bzw. $[t_{an}, \infty)$ abgebildet, wobei die verschiedenen fahrplanmäßigen Verbindungen, durch alternative Knoten dargestellt werden.

³Es gibt hierbei verschiedene Systeme, bei denen entweder nur der Auflieger oder das gesamte Fahrzeug inkl. Zugmaschine befördert wird. Eine logistische Bewertung verschiedener Konzepte wurde am IFL im Rahmen des vom BMBF geförderten Projekts RoRoRail durchgeführt.

5.5.3 Tageszeitabhängige Fahrtkosten

Eine der Ideen hinter Straßennutzungsgebühren, wie sie bspw. in Deutschland während der Erstellung dieser Arbeit für LKW auf Autobahnen eingeführt wurde, besteht in einer aufkommensabhängigen Verkehrslenkung mit dem Ziel, Teile des Verkehrs in potentiell verkehrsärmere Zeiträume zu verlagern.

Ein Tag kann in mehrere Zeiträume unterteilt werden, die sich hinsichtlich des Verkehrsaufkommens unterscheiden. Es ist daher konsequent anzunehmen, daß bei einer starken Gewichtung des zeitlichen Aspekts in der Zielfunktion die Fahrtkosten der einzelnen Zeiträume unterschiedlich sind. Die Aufgabe der Disposition wird daher erweitert um die Auswahl des Zeitraumes innerhalb dessen die Transportabwicklung stattfinden soll. Für jeden Zeitraum und jeden Kunden wird daher eine alternative Liefermöglichkeit im Modell vorgesehen und die Kanten des zugehörigen Netzwerkes geben die möglichen direkten Fahrten innerhalb eines Zeitraumes an. Auch Fahrten, die sich über mehrere Zeiträume erstrecken, können durch dieses Modell abgebildet werden.

6 Lösungsverfahren für das VRPA

*Gescheitert bin ich nicht –
ich habe nur 10.000 Wege gefunden, die nicht funktionieren*
Thomas Alva Edison

Aus den grundsätzlichen methodischen Möglichkeiten und den Erkenntnissen der Literaturanalyse erwachsen zahlreiche Gestaltungsmöglichkeiten für das hybride Optimierungsverfahren zur Lösung von VRPA-Modellen. Das Gesamtverfahren besteht aus den Komponenten

- Master-Problem,
- Pricing-Problem,
- Herstellung ganzzahliger Lösungen,
- Behandlung von Spalten.

Die Rollen der ersten drei Bestandteile wurden bereits in der Darstellung der methodischen Grundlagen in Kapitel 3 beschrieben. Beim vierten Bestandteil – der Behandlung von Spalten – handelt es sich um eine zusätzliche Vermittlungsschicht zwischen Master- und Pricing-Problem. Aufgabe der Vermittlungsschicht ist die Versorgung des (Restricted) Master-Problems mit Routen, wobei diese entweder durch Neuberechnung des Pricing-Problems erzeugt oder aus einem Routenpuffer entnommen werden.

Für die einzelnen Komponenten stehen unterschiedliche Umsetzungsmöglichkeiten zur Wahl, die jedoch die Schnittstelle der Komponenten unverändert lassen. Durch diesen Aufbau ist der Austausch von Verfahrenselementen unterhalb der Komponentenebene möglich, ohne die grundsätzliche Architektur des Verfahrens zu modifizieren. Durch die Implementierung verschiedener Varianten für einzelne Komponenten entsteht eine Verfahrensfamilie, deren grundsätzlicher Aufbau schematisch in Abbildung 6.1 wiedergegeben wird.

In den folgenden Unterkapiteln werden die vier Komponenten und die jeweils implementierten Verfahrenselemente und -varianten detailliert vorgestellt. Besonderes Augenmerk finden alternative Liefermöglichkeiten, deren Behandlung Anpassungen und Erweiterungen gegenüber den in der Literatur empfohlenen Techniken und Vorgehensweisen erfordert.

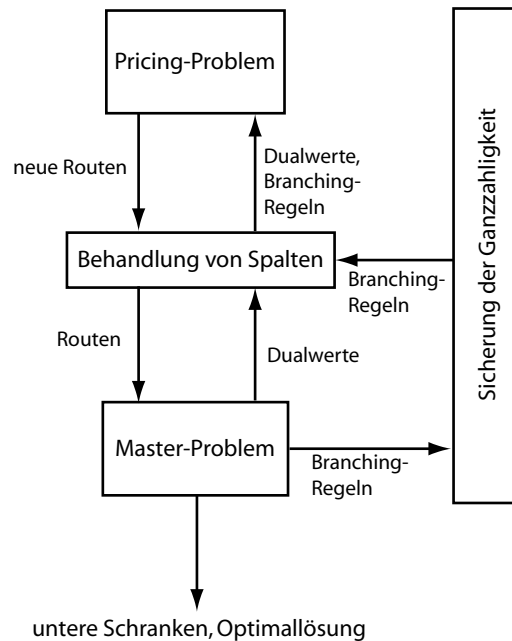


Abbildung 6.1: Schematische Darstellung des Optimierungsverfahrens

6.1 Master-Problem

Für das Master-Problem existieren im wesentlichen drei Freiheitsgrade:

- die Formulierung des Modells,
- die Herstellung einer Initiallösung und
- die Wahl einer Strategie zur Behandlung neuer Spalten.

Während die ersten beiden Punkte nur das Master-Problem betreffen, sollte die Strategie zur Behandlung neuer Spalten in Verbindung mit dem Pricing-Problem gewählt werden. Aus diesem Grund wird die Beschreibung der Umsetzung dieses Merkmals zunächst zurückgestellt.

6.1.1 Set Partitioning- und Set Covering-Formulierungen

Aufgabe des (Restricted) Master-Problems ist die Zusammenstellung von Routen zu einem Tourenplan. Da jeder Kunde genau einmal besucht werden soll, ist ein Set Partitioning-Modell im Prinzip die richtige Wahl. Numerische Gründe (s. (Barnhart u. a. 1998)) sprechen allerdings für ein Set Covering-Modell, bei dem die „=“ Bedingung gegen eine „ \geq “ Bedingung ausgetauscht wird. Es wäre demnach zulässig, einen Kunden mehrmals zu besuchen. Wenn die Dreiecksungleichung gilt, ist dies bei einem

Minimierungsproblem allerdings ausgeschlossen, da es immer zulässig und kostengünstiger ist, einen Kunden aus einer Route zu entfernen. Die Optimallösung einer Set Covering-Formulierung ist daher zugleich eine optimale Set Partitioning-Lösung.

Im umgekehrten Fall, d.h. wenn die Dreiecksungleichung nicht gilt, kann es eine Routenkonstellation der Kunden i, j, k, p, q geben, für die

$$c_{ik} + c_{kj} < c_{ij}$$

und

$$c_{pk} + c_{kq} < c_{pq}$$

gilt. In diesem Fall lohnt es sich, den Kunden k zwischen i und j bzw. p und q einzufügen, um die Kosten zu reduzieren. Weil demnach die Aufnahme eines zusätzlichen Kunden zu geringeren Kosten führt, wird in der Optimallösung des Set Covering-Problems der Kunde k mehrfach besucht, was nicht mit der Set Partitioning-Formulierung vereinbar ist.

Eine analoge Situation tritt bei der Abbildung alternativer Lieferorte auf, weil in der Koeffizientenmatrix des (R)MP einzelne Alternativen nicht unterschieden werden können.

Bezeichnen $i_1, i_2 \in \mathcal{A}_i$, $j_1, j_2 \in \mathcal{A}_j$ jeweils zwei alternative Knoten der Kunden i bzw. j und ist $p \in \mathcal{A}_k$ ein weiterer Knoten, kann entsprechend der Datenlage

$$c_{i_2 p} + c_{p j_2} < c_{i_1 j_1} \tag{6.1}$$

gelten, d.h. daß es kostengünstiger ist, zwischen den jeweils zweiten Alternativen den Knoten p zu besuchen, als die jeweils ersten Alternativen direkt zu verbinden¹

Findet sich die Kantenfolge der linken Seite von (6.1) in Route r_1 und der rechten Seite in Route r_2 , lauten die Einträge der Koeffizientenmatrix des (R)MP $a_{i r_1} = a_{j r_1} = a_{k r_1} = 1$ bzw. $a_{i r_2} = a_{j r_2} = 1$ und für die Kosten gilt $c_{r_1} < c_{r_2}$.

Im RMP ist diese Konstellation nicht unterscheidbar von der oben gezeigten Form der unerfüllten Dreiecksungleichung, die Set-Covering-Formulierung daher nicht anwendbar.

Aus diesem Grund wird im Verfahren daher bei vorhandenen Alternativen eine Set Partitioning und andernfalls eine Set Covering Formulierung für die kundenbezogenen Restriktionen des (Restricted) Master-Problems gewählt.

¹Die weiteren Routenkosten, d.h. zu den Knoten i_1 und j_1 sowie von den Knoten i_2, j_2 , werden als gleich angenommen.

6.1.2 Berechnung einer Initiallösung

Für die Berechnung einer Initiallösung wird auf Heuristiken zurückgegriffen, die in relativ kurzer Zeit zulässige Lösungen erzeugen. Im Rahmen von Vorstudien hat es sich als günstig erwiesen, zu Beginn des Column Generation über eine umfangreiche Sammlung von möglichen Routen zu verfügen. Dadurch wird bereits eine gute Qualität der Ausgangslösung erreicht und das Pricing-Problem wird über die Dualvariablen mit zutreffenden Kosteninformationen versorgt. Da diese Sammlung mit schnellen Heuristiken erzeugt wird, muß das aufwendige CP-basierte Pricing-Problem seltener durchlaufen werden.

Abzuwägen ist zwischen der Qualität und der Anzahl der erzeugten Routen. Tendenziell erzeugen Metaheuristiken Routen besserer Qualität, deren Berechnung jedoch im Vergleich zu einfacheren Kombinationen von Eröffnungs- und Verbesserungsverfahren länger dauert. Verschiedene graduelle Abstufungen bis hin zu trivialen Eröffnungslösungen mit jeweils unterschiedlicher Routenzahl, Routenqualität und Berechnungsdauer sind in der Initialphase denkbar. Für die experimentelle Untersuchung des Einflusses der Initiallösung auf die Leistungsfähigkeit des Verfahrens sind folgende Varianten der Initialphase implementiert worden:

Trivaleröffnung: Die Eröffnungslösung besteht aus einer Menge von Touren, die jeweils nur einen Knoten besuchen. Dieser Tourenplan wird daher fast immer unzulässig sein, da die Anzahl der Touren die Flottengröße übersteigt.

Einfache Konstruktionsheuristik: Mit dem Savings-Verfahren wird eine zulässige erste Lösung bestimmt.

Einfache Konstruktionsheuristik und lokale Suche: Im Anschluß an die Berechnung der Eröffnungslösung mit dem Savings-Verfahren werden bessere Lösungen durch die lokalen Suchverfahren 2-opt, Or-opt, Relocate, Exchange und Crossover gesucht. Die Verbesserungsverfahren werden abgebrochen, sobald keine bessere Lösung gefunden wird, d.h. ein lokales Optimum bezüglich aller Nachbarschaften gefunden wurde.

Mehrfache Konstruktionsheuristik: Es werden mehrere Eröffnungsverfahren durchlaufen. In einzelnen handelt es sich um die Verfahren Savings, Sweep und zwei Nächster-Nachbar-Verfahren.

Mehrfache Konstruktionsheuristik und lokale Suche: Gegenüber dem zuvor genannten Ansatz werden für jedes Eröffnungsverfahren zusätzlich noch lokale Suchverfahren angestoßen, um verbesserte Lösung zu erhalten. Es werden die gleichen Nachbarschaften wie im Fall der einfachen Konstruktionsheuristik verwendet.

Einfache Konstruktionsheuristik und Tabu Search: Für die Bestimmung einer zulässigen Eröffnungslösung wird die Savings-Heuristik verwendet. In der zweiten

Phase wird die lokale Suche über eine Tabu Search-Metaheuristik gesteuert. Es finden maximal 100 Iterationsschritte statt.

Das Ergebnis der Eröffnungsverfahren besteht jeweils aus einem vollständigen Tourenplan mit einer a priori nicht bekannten Zahl von Touren. Jeder erfolgreiche, d.h. verbessernde, Schritt der lokalen Suche führt zu einem weiteren Tourenplan, dessen Touren gespeichert werden. Um mehrfach vorhandene Routen bereinigt, werden sämtliche gesammelten Routen aus der Initialphase dem Restricted Master-Problem übergeben.

6.2 Pricing-Problem

Zur Lösung des Subproblems werden neben einfachen Constraints auch globale Constraints entwickelt und eingesetzt. Diese lassen sich in Optimierungsconstraints und Filteralgorithmen unterscheiden. Letztere adressieren die Zulässigkeit des Problems und kommen für die Strukturbedingungen, die Kapazitätsbedingung und die Zeitfensterreduktion zum Einsatz. Constraints zur Eliminierung von Kanten, die nicht Bestandteil der Optimallösung sein können und die relaxationsgestützte Berechnung unterer Schranken sind globale Constraints vom ersten Typ. In diesem Unterkapitel wird die algorithmische Realisierung dieser Constraints im Detail vorgestellt.

6.2.1 Strukturelle Constraints

Aufgabe der strukturellen Constraints ist es, die Konsistenz der Wertebereiche in einer Weise zu sichern, daß es einen Pfad vom Ausgangs- zum Enddepot gibt. Dies wurde im Modell durch die `path`-Constraint erreicht. Das wesentliche Element hierbei ist ein Algorithmus, um alle Werte aus den Wertebereichen der Vorgänger-/Nachfolgervariablen zu entfernen, die zu Subtouren, d.h. Zyklen, führen.

Zu diesem Zweck wird auf die `NoSubTour`-Constraint aus (Pesant, Gendreau, Potvin und Rousseau 1998) zurückgegriffen. Die Idee dieser globalen Constraint ist es, bei der Verkettung zweier Teilsequenzen von Knoten den ersten Knoten der neuen Sequenz aus dem Wertebereich des letzten Knotens der neuen Sequenz zu entfernen. Sind zwei Teilketten gegeben mit

$$S_{i_0} = i_1, S_{i_1} = i_2, \dots, S_{i_k} = i_{k+1}$$

und

$$S_{j_0} = j_1, S_{j_1} = j_2, \dots, S_{j_l} = j_{l+1}$$

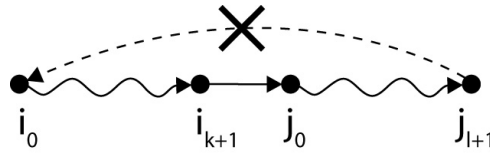


Abbildung 6.2: Funktionsweise der NoSubTour-Constraint

und wird im Verlauf der Suche $S_{i_{k+1}} = j_0$ gesetzt, wird in der NoSubTour-Constraint i_0 aus dem Wertebereich von $S_{j_{l+1}}$ entfernt. Abbildung 6.2 verdeutlicht das Vorgehen.

Die NoSubTour-Bedingung ist besonders wichtig, wenn die Suchstrategie zur Bildung vieler Teilsequenzen führt. Dies ist typischerweise der Fall für eine First Fail-Strategie. Der Grund dafür ist die Wahl der zu instanziiierenden Variable aufgrund kleinster Domänengröße, die zu einer Struktur von mehreren Teilketten führt, die schließlich zu einer Route verschmolzen werden. Zwischenzeitliche Instanzierungen eröffnen neue Teilketten oder verlängern bestehende, die zum Zeitpunkt der Instanzierung jedoch nicht notwendigerweise mit dem Ausgangs- oder Enddepot verbunden sein müssen.

Um eine effiziente Behandlung der Constraint zu erreichen, werden die Teilketten in den Suchknoten gespeichert und in jedem Schritt aktualisiert. Gespeichert werden zu jeder Teilkette T der jeweils erste und letzte Knoten T^A bzw. T^E sowie zu jedem Knoten i die Teilkette T_i , in der dieser sich befindet. Die Propagation wird bei Instanzierung einer Nachfolgervariablen S_i angestoßen. Wird in einem Suchschritt die Variable S_i mit dem Wert j ($i \neq j$) instanziiert, werden zunächst die dazugehörigen Teilketten T_i bzw. T_j nachgeschlagen. Anschließend wird aus dem Wertebereich von $S_{T_j^E}$ der Wert T_i^A entfernt. Der letzte Schritt besteht darin, die Teilketteninformation zu aktualisieren. Die weiteren strukturellen Constraints sind einfachen Typs und entsprechen denen des Modells. Konkret werden über

$$S_i = j \Leftrightarrow P_j = i, \quad \forall i \in \mathcal{N}_O, j \in \mathcal{N}_D \quad (6.2)$$

Vorgänger- und Nachfolgervariablen miteinander verknüpft und durch

$$S_0 \in \mathcal{N}_C \quad (6.3)$$

dafür gesorgt, daß auf dem Pfad mindestens ein Kunde besucht wird.

6.2.2 Behandlung von Zeitfenster- und Kapazitätsrestriktionen

Zeitfenster

In (Desrochers, Desrosiers und Solomon 1992) werden vier Bedingungen zur Reduktion von Zeitfenstern angegeben. Diese werden zum einen in einer Preprocessing Phase zur Verkleinerung der Zeitfenster (und damit zur Verkleinerung der Wertebereiche der Servicezeitvariablen) verwendet, zum anderen werden die Bedingungen

$$T_i \geq \min_{k \in P_i} (T_k + t_{ki}) + \tau_s \quad \forall i \in \mathcal{N}_D \quad (6.4)$$

und

$$T_i \leq \max_{k \in S_i} (T_k - t_{ik}) - \tau_s \quad \forall i \in \mathcal{N}_O \quad (6.5)$$

auch während der Propagation infolge von Wertebereichsveränderungen der Vorgänger- oder Nachfolgervariablen angestoßen. Bedingung (6.4) gibt eine untere Schranke für die Servicezeit aufgrund der frühestmöglichen Ankunft von einem möglichen Vorgänger aus an. Ähnlich verhält sich Bedingung (6.5), die einen spätestmöglichen Abfahrtszeitpunkt zu einem möglichen Nachfolger angibt.

Beide Regeln werden bei einer Wertebereichsveränderung der Vorgänger- oder Nachfolgervariablen angestoßen. Während der Behandlung der Regel (6.4) wird die untere Grenze des Wertebereichs der Variablen T_i gegebenenfalls vergrößert. Infolgedessen können u.U. einige der bis dahin möglichen Nachfolger nicht mehr erreicht werden und die Menge S_i wird verkleinert. Dies stößt die Propagation erneut an, diesmal jedoch erfolgt die Behandlung durch Regel (6.5). Der Propagationsmechanismus sorgt durch wechselweisen Aufruf der Regeln zur Zeitfensterreduktion für eine Reduktion der Domänen der Vorgänger- und Nachfolgervariablen.

Obwohl in der ursprünglichen Veröffentlichung vier Reduktionsregeln angegeben wurden, werden für die Propagation lediglich die beiden Regeln (6.4) und (6.5) verwendet, da in (Pesant u. a. 1998) festgestellt wurde, daß die restlichen Regeln nur eine schwache Wertebereichsreduktion erlauben. In den Abbildungen 6.3 und 6.4 werden beide Regeln nochmals veranschaulicht.

Kapazitäten

Für die Berücksichtigung der Kapazitätsrestriktionen wird ein sehr einfacher *forward checking* Filteralgorithmus verwendet: Nachdem eine Nachfolgervariable an einen Wert gebunden wurde, wird die Restkapazität des Fahrzeugs berechnet (aufgrund der feststehenden Bedarfe in der aktuellen Teillösung) und aus den Wertebereichen der

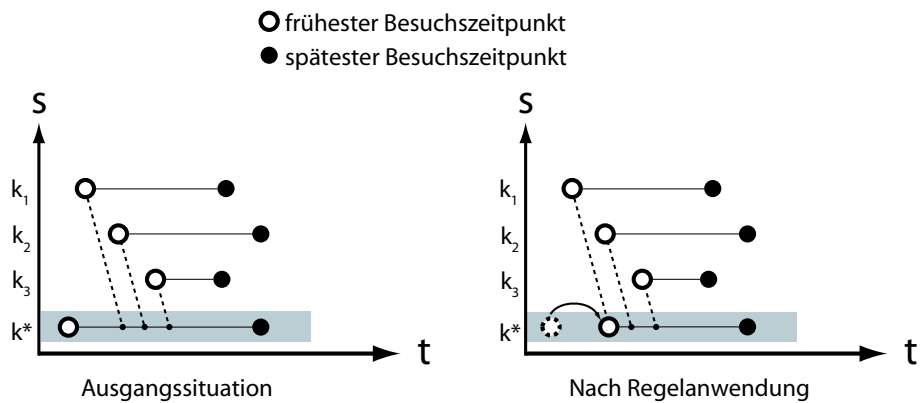


Abbildung 6.3: Vorgehen der Regel 1 aus (Desrochers u. a. 1992): Früheste Ankunftszeit

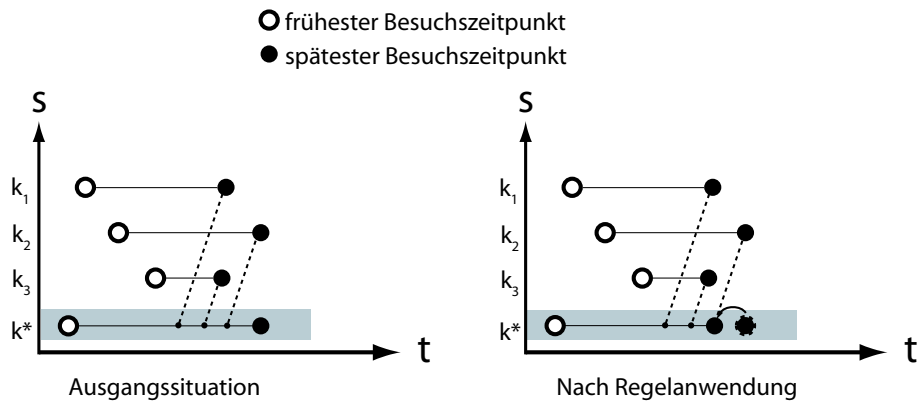


Abbildung 6.4: Vorgehen der Regel 4 aus (Desrochers u. a. 1992): Späteste Abfahrtszeit

noch ungebundenen Nachfolgervariablen diejenigen Knoten entfernt, deren Besuch die Restkapazität übersteigen würde.

6.2.3 Optimierungsconstraints

Neben Constraints, die hauptsächlich die Konsistenz der Wertebereiche im Hinblick auf Zulässigkeit sicherstellen, werden Optimierungsconstraints eingesetzt, die diejenigen Variable–Wert-Zuordnungen aus den Wertebereichen der Variablen entfernen, die nicht Bestandteil der Optimallösung sein können.

Eliminierung von Kanten

Bei der Bestimmung kürzester Wege ist es naheliegend, Umwege zu vermeiden. Wann immer es demnach möglich ist, einen Knoten auf direktem Weg mit niedrigen Kosten zu erreichen, können andere – indirekte und teurere – Wege vermieden werden. Voraussetzung dafür ist allerdings, daß die Dreiecksungleichung für alle Ressourcen (Bedarfsmengen, Fahrzeiten und Entfernungen) gilt: In diesen Fällen ist eine direkte Verbindung zwischen zwei Knoten immer besser als eine indirekte Verbindung über einen Zwischenknoten und zudem ist die direkte Verbindung zulässig wenn es die indirekte auch ist.

Für ein fest gegebenes Netzwerk wäre es demnach möglich, in einem Vorverarbeitungsschritt Umwege zu entfernen. Bei der Lösung des Pricing-Problems mittels Constraint Programming ist das Netzwerk allerdings nicht statisch, da sich die Kantenbewertungen zwar durch

$$\hat{c}_{ij} = c_{ij} - u'_i$$

ergeben und somit feststehen, einzelne Kanten aber infolge der Constraint Propagation aus dem Netzwerk entfernt werden. Die Prüfung auf redundante Umwege wird daher in den Propagationsprozeß integriert und bei Veränderung der Netzwerkstruktur (infolge der Veränderung der Wertebereiche von Vorgänger- und Nachfolgervariablen) angestoßen.

Für das VRPTW sind drei Typen dieser Constraints, ArcElimination genannt, von (Rousseau u. a. 2002) entwickelt worden, die im folgenden erläutert werden sollen.

Typ 1 Ausgehend von einer Kante $\langle i, j \rangle$, d.h. $j \in S_i$, werden alle Nachfolger k des Knotens j , d.h. $k \in S_j$, dahingehend überprüft, ob sie von Knoten i aus immer günstiger als von j aus erreicht werden können, d.h. es wird geprüft ob

$$\hat{c}_{ik} \leq \hat{c}_{ij} + \hat{c}_{jk} \Leftrightarrow c_{ik} \leq c_{ij} + c_{jk} - u'_j \quad (6.6)$$

für alle $k \in S_j, k \neq i \neq j$ gilt. Ist dies der Fall, kann die Kante $\langle i, j \rangle$ entfernt, d.h. $S_i \neq j$ gesetzt werden, da sie nie Bestandteil einer optimalen Lösung sein kann.

Typ 2 Die Argumentation für Kanteneliminierungsbedingungen vom Typ 2 ist ähnlich wie für Typ 1. Der Unterschied besteht darin, daß die Vorgänger des Knotens i untersucht werden und die Kante $\langle i, j \rangle$ verworfen wird wenn es immer günstiger ist, direkt von einem Vorgängerknoten $k \in P_i$ nach j zu gehen. Formal muß also die Bedingung

$$\hat{c}_{kj} \leq \hat{c}_{ki} + \hat{c}_{ij} \Leftrightarrow c_{kj} \leq c_{ki} + c_{ij} - u'_i \quad (6.7)$$

für alle $k \in P_i, k \neq i \neq j$ gelten um $S_i \neq j$ zu setzen.

Typ 3 Bei den Bedingungen vom Typ 3 handelt es sich im wesentlichen um eine Komposition der beiden vorangegangenen Typen. Betrachtet werden wiederum eine Kante $\langle i, j \rangle$ sowie ein Vorgängerknoten $k \in P_i$ und ein Nachfolgerknoten $m \in S_j$. Untersucht wird anhand des Kriteriums

$$\hat{c}_{km} \leq \hat{c}_{ki} + \hat{c}_{ij} + \hat{c}_{jm} \Leftrightarrow c_{km} \leq c_{ki} + c_{ij} - u'_i + c_{jm} - u'_j \quad (6.8)$$

ob eine Direktverbindung (k, m) für alle Vorgänger $k \in P_i, k \neq i \neq j \neq m$ und alle Nachfolger $m \in S_j$ günstiger als über die Kante $\langle i, j \rangle$ ist.

Einleitend wurde in diesem Abschnitt für die Anwendbarkeit der ArcElimination-Constraints die Gültigkeit der Dreiecksungleichung für alle Ressourcen vorausgesetzt. Wie bereits in Abschnitt 6.1.1 festgestellt, ist im Fall der Tourenplanung mit alternativen Liefermöglichkeiten diese Voraussetzung nicht gegeben. Am Beispiel der ArcElimination Constraint vom Typ 1 soll verdeutlicht werden, daß für die Anwendbarkeit der ArcElimination Constraints auf das VRPA eine Erweiterung in Form einer zusätzlichen Prüfung erforderlich ist.

Aus der gültigen Dreiecksungleichung kann aus $j \in S_i$ und $k \in S_j$ gefolgert werden, daß $k \in S_i$ ist, d.h. es gilt die Transitivität der \in -Relation bezüglich Wertebereiche von Variablen. Diese Annahme wird jedoch im VRPA verletzt, wenn k und i alternative Liefermöglichkeiten des gleichen Kunden p , d.h. $i, k \in \mathcal{A}_p$, sind.

Um die ArcElimination-Constraints dennoch anwenden zu können, muß daher zusätzlich geprüft werden, ob die Transitivität erfüllt ist. Im Fall der ArcElimination Constraint vom Typ 1 bedeutet dies die zusätzliche Forderung nach $k \in S_i$.

In ähnlicher Weise ergeben sich die zusätzlich benötigten Prüfungen für alle drei Fälle. In der folgenden Aufstellung sind für alle drei Typen von ArcElimination-Constraints die Implikationen angegeben, die zu einer Kanteneliminierung führen.

Typ 1

$$\begin{aligned} i \in \mathcal{N}_O, j \in S_i : \forall k \in S_j, i \neq j \neq k, k \in S_i : \\ \hat{c}_{ik} \leq \hat{c}_{ij} + \hat{c}_{jk} \implies j \notin S_i \end{aligned} \quad (6.9)$$

Typ 2

$$\begin{aligned} i \in \mathcal{N}_O, j \in S_i : \forall k \in P_i, i \neq j \neq k, j \in S_k : \\ \hat{c}_{kj} \leq \hat{c}_{ki} + \hat{c}_{ij} \implies j \notin S_i \end{aligned} \quad (6.10)$$

Typ 3

$$\begin{aligned} i \in \mathcal{N}_O, j \in S_i : \forall k \in P_i, m \in S_j, i \neq j \neq k \neq m, m \in S_k : \\ \hat{c}_{km} \leq \hat{c}_{ki} + \hat{c}_{ij} + \hat{c}_{jm} \implies j \notin S_i \end{aligned} \quad (6.11)$$

Gegenüber dem VRPTW ist die ArcElimination Constraint schwächer, weil die zusätzliche Prüfung auf Mitgliedschaft in der Domäne die Effizienz reduziert und die Bedingungen (6.9) bis (6.11) naturgemäß seltener erfüllt sind als (6.6) bis (6.8). In Abbildung 6.5 bis 6.7 sind die drei Methoden zur Kanteneliminierung nochmals veranschaulicht.

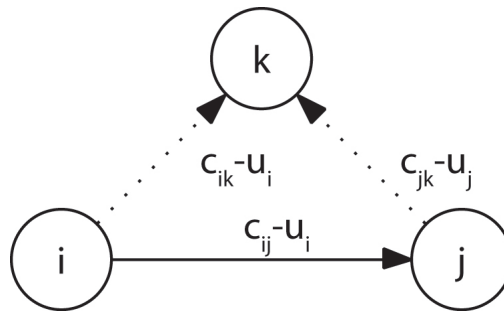


Abbildung 6.5: ArcElimination vom Typ 1

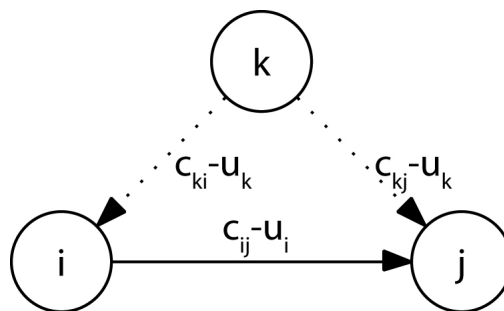


Abbildung 6.6: ArcElimination vom Typ 2

Bounding

Die zweite Optimierungsconstraint dient dazu, eine untere Schranke des ressourcenbeschränkten Kürzeste-Wege-Problems zu bestimmen. Eine Relaxation des Problems wird in einer globalen Constraint `lowerbound` gekapselt und durch effiziente Algorithmen gelöst. Konkret wird das (lineare) Zuordnungsproblem (*Assignment Problem*)

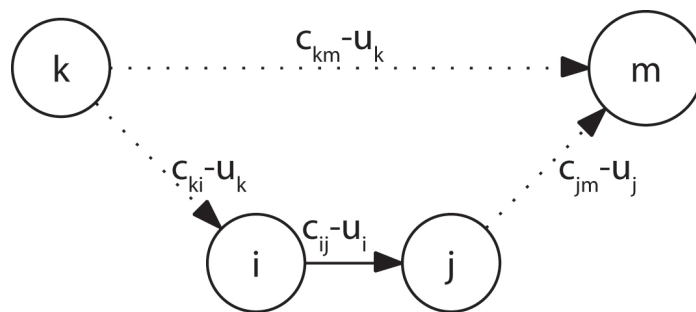


Abbildung 6.7: ArcElimination vom Typ 3

als Relaxation verwendet. Dieses MIP hat im allgemeinen für einen bipartiten Graphen mit n Knoten die Form

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (6.12)$$

$$(AP) \quad \sum_{j=1}^n x_{ij} = 1 \quad \forall i \in \{1, \dots, n\} \quad (6.13)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in \{1, \dots, n\} \quad (6.14)$$

$$x_{ij} \in \{0, 1\} \quad (6.15)$$

und weist auf Grund der *totalen Unimodularität* (vgl. (Wolsey 1998)) der Koeffizientenmatrix die *Integralitätseigenschaft* auf, d.h. die optimale Lösung der LP-Relaxation des Problems liefert stets eine ganzzahlige Lösung.

Die Bedingungen (6.13) und (6.14) des Problems (AP) besagen, daß jeder Knoten genau einen Nachfolger bzw. Vorgänger hat. Fügt man dem Kürzeste-Wege Problem eine kostenneutrale Hilfskante vom End- zum Ausgangsdepot zurück hinzu und setzt zudem $c_{ii} = 0$ für jeden Knoten i , ist ein Pfad vom Ausgangs- zum Enddepot zugleich eine gültige Lösung des Zuordnungsproblems, wenn alle Knoten, die nicht auf dem Pfad liegen, sich selbst zugeordnet werden. Umgekehrt beschreibt eine Lösung des Zuordnungsproblems allerdings nicht notwendigerweise einen Pfad, da eine Lösung des Zuordnungsproblems auch Subtouren enthalten kann. Daraus folgt, daß nach Konstruktion des Graphen das Zuordnungsproblem eine Relaxation des Kürzeste-Wege-Problems ist und seine optimale Lösung eine untere Schranke darstellt.

Um die Lösung der Relaxation für die Suche zu verwerten, bedarf es einer festgelegten Beziehung zwischen Variablen der Relaxation und des Constraint Programming-Modells. Im vorliegenden Fall entsprechen die binären Variablen x_{ij} einer Kante $\langle i, j \rangle$ des zugrundeliegenden Graphen, die im CP Modell durch $j \in S_i$ impliziert wird. Es

gilt

$$x_{ij} = 1 \iff S_i = j. \quad (6.16)$$

Der Graph auf dem das Zuordnungsproblem gelöst wird, ergibt sich durch die jeweiligen Wertebereiche der Nachfolger- und Vorgängervariablen. Dabei sind einige Zuordnungen durch bereits erfolgte Instanziierungen vorgegeben, so daß lediglich ergänzende Zuordnungen für noch nicht instanziierte Knoten ermittelt werden müssen. Die Menge möglicher Vorgänger und Nachfolger ergeben sich dabei aus den im Suchablauf bereits erfolgten Wertebereichsreduktionen. Das zugehörige MIP lautet folgendermaßen:

$$\min \sum_{i \in \mathcal{N}_O} \sum_{j \in S_i} \hat{c}_{ij} x_{ij} \quad (6.17)$$

$$(DBAP) \quad \sum_{j \in S_i} x_{ij} = 1 \quad \forall i \in \mathcal{N}_O \quad (6.18)$$

$$\sum_{i \in P_j} x_{ij} = 1 \quad \forall j \in \mathcal{N}_D \quad (6.19)$$

$$x_{ij} \in \{0, 1\}. \quad (6.20)$$

Die optimale Lösung von (DBAP) (für *domainbasiertes Assignment Problem*) liefert einen Permutationsvektor $(\pi(0), \pi(1), \dots, \pi(m))^T$ mit $x_{0, \pi(0)} = 1, x_{1, \pi(1)} = 1, \dots, x_{m, \pi(m)} = 1$, den Zielfunktionswert z , sowie die $|\mathcal{N}_O|$ - bzw. $|\mathcal{N}_D|$ -wertigen Dualvektoren u und v . Mittels letzterer können die reduzierten Kosten

$$\tilde{c}_{ij} = \hat{c}_{ij} - u_i - v_j \quad \forall i \in \mathcal{N}_O, j \in \mathcal{N}_D \quad (6.21)$$

der Kanten $\langle i, j \rangle$ bestimmt werden, um die Wertebereiche der Variablen entsprechend (3.33) zu reduzieren. Gegebenenfalls kann nach (3.32) die weitere Auslotung des Teilbaums abgebrochen werden. Die Lösung des Zuordnungsproblems wird im Suchknoten gespeichert und an die Nachfolgerknoten im Teilbaum weitergegeben, um dort falls erforderlich aktualisiert zu werden.

Die Propagation der *lowerbound-Constraint* wird angestoßen, sobald sich der Wertebereich einer Nachfolgervariablen verändert. Zunächst wird geprüft, ob die aktuelle Lösung des Zuordnungsproblems, wie sie vom übergeordneten Knoten des Suchbaums übernommen wurde, nach der auslösenden Wertebereichsveränderung noch gültig ist. Dies ist immer dann der Fall, wenn für alle Elemente des Permutationsvektors $\pi(i) \in S_i$ gilt. Anderenfalls ist eine Neuberechnung des Zuordnungsproblems notwendig, wozu im Rahmen dieser Arbeit der *Shortest Augmenting Path*-Algorithmus aus (Jonker und Volgenant 1987) – angepaßt für die Verwendung von Gleitkommazahlen – verwendet wird. Zwar ist aus theoretischer Sicht auch die direkte Lösung der LP-Relaxation von

(DBAP) möglich, in der Praxis zeigt sich aber gemäß Jonker und Volgenant, daß spezialisierte polynomiale Verfahren effizienter sind. Der Aufwand für die Berechnung einer optimalen Zuordnung in einem bipartiten Graphen mit n Knoten beträgt $O(n^3)$.

Behandlung alternativer Liefermöglichkeiten

Die bisherige Darstellung der Funktionsweise der `lowerbound-Constraint` berücksichtigt nicht explizit alternative Liefermöglichkeiten.

Eine Bedingung in der Form von

$$\sum_{p \in \mathcal{A}_i} \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{V}} x_{pjk} = 1 \quad \forall i \in \mathcal{C}, \quad (6.22)$$

wie sie in (5.28) für das kompakte Modell angegeben wurde, in das Modell (AP) einzuführen würde die Struktur des Problems zerstören: Beispielsweise wäre die totale Unimodularität nicht mehr gegeben und spezialisierte Verfahren u.U. nicht mehr anwendbar.

Es stellt sich allerdings die Frage, inwiefern eine Berücksichtigung des gegenseitigen Ausschlusses alternativer Liefermöglichkeiten in der Relaxation erforderlich ist. Tatsächlich handelt es sich bei (DBAP) nicht nur um eine Relaxation des ressourcenbeschränkten Kürzeste-Wege-Problems, sondern auch um eine Relaxation des für die Behandlung alternativer Liefermöglichkeiten eigentlich benötigten *DBAP mit Alternativenausschluss*: Jede Lösung des letzteren ist damit auch eine Lösung des (DBAP) und die Lösung von (DBAP) enthält² nur dann von einer Teilmenge der Kunden mehr als eine Alternative, wenn dadurch ein niedrigerer Zielfunktionswert erreicht werden kann. Der Fall einer Aufnahme mehrerer Alternativen eines Kunden kann aber nur dann auftreten, wenn noch keine Festlegung im Suchprozess stattgefunden hat. Über eine entsprechende Gestaltung des Suchprozesses, bspw. des Branchings im Masterproblem oder der Variableninstanziierung im Subproblem, kann eine frühzeitige Festlegung allerdings angestrebt und diese Situation weitestgehend vermieden werden. In der Konsequenz bedeutet dies, daß die untere Schranke im Fall vorhandener alternativer Liefermöglichkeiten zu optimistisch (zu niedrig) geschätzt wird und die Reduktions- und Abbruchmechanismen später als im Fall ohne Alternativen greifen.

Verbesserung der unteren Schranke

Zu einem kombinatorischen Optimierungsproblem gibt es u.U. mehrere strukturell verschiedene Relaxationen, zu deren Lösung effiziente Algorithmen existieren. Das von (Fischetti und Toth 1989) aufgezeigte Verfahren des *additive Bounding* eröffnet eine

²Die Formulierung „die Lösung des Zuordnungsproblems enthält einen Knoten i “ ist so zu verstehen, daß in der Lösung des Zuordnungsproblems der Knoten i nicht sich selbst zugewiesen wird.

Möglichkeit, die unterschiedlichen Strukturen effektiv zu nutzen. Das Verfahren funktioniert im wesentlichen so, daß die Relaxationen nacheinander berechnet werden. Die Lösungsinformation einer Relaxation geht dabei als Kosteninformation in die folgende ein. Voraussetzung für das Funktionieren des Verfahrens ist, daß aus der Lösung einer Relaxation neben einer unteren Schranke auch die reduzierten Kosten der Lösung ableitbar sind. Die Summe der unteren Schranken über alle Relaxationen ergibt dann eine untere Schranke des Ausgangsproblems. Erfolgreich angewandt wurde die Methode in (Fischetti und Toth 1992) und jüngst auch im Suchprozess von Constraint Programmen in (Lodi und Milano 2003).

In dieser Arbeit wird diese Technik so angewandt, daß zunächst – wie zuvor beschrieben – das Zuordnungsproblem gelöst und die reduzierten Kosten der Kanten bestimmt werden. Diese bilden die Kantenbewertungen eines neuen Graphen, für den ein *Minimaler Spannender Baum* (engl. *Minimum Spanning Tree*) mittels des *Kruskal-Verfahrens* (vgl. bspw. (Jungnickel 1999)) bestimmt wird. Beide Relaxationen ergänzen sich, da im Zuordnungsproblem Zyklen auftreten können, während der MST keine Zyklen aufweist, dafür jedoch einem Knoten mehrere Nachfolger zuweisen kann. Während die Qualität der unteren Schranke auf diese Weise wächst, fällt andererseits ein entsprechender Berechnungsaufwand für die Lösung des MST an, auch wenn der Aufwand sich entsprechend $O(m \cdot \ln m)$ verhält (wobei m hier die Kantenzahl des Graphen ist) und die Lösung des Zuordnungsproblems damit den Gesamtaufwand bestimmt. Die Ergebnisse einer experimentellen Untersuchung über Vor- und Nachteile dieses Ansatzes werden in Kapitel 7 wiedergegeben.

6.2.4 Zielfunktion

In Abschnitt 3.2.5 wurde deutlich, daß es für die Berechnung neuer Routen im Pricing-Problem nicht erforderlich ist, die optimale Lösung zu bestimmen. Vielmehr werden im Prinzip lediglich Routen mit negativen reduzierten Kosten gesucht. Diese Vorgehensweise verspricht eine schnelle Erzeugung einer großen Anzahl von Routen. Es ist zu erwarten, daß die erzeugten Routen aufgrund ihres Erzeugungsmusters ähnlich sind, d.h. viele gleiche Knoten aufweisen und von vergleichbarer Länge sind. Im RMP könnte dies zu vielen zulässigen Lösungen mit nur geringem Fortschritt im Zielfunktionswert führen.

Demgegenüber erfordert die Berechnung der optimalen Route eine längere Berechnungszeit, erzeugt dafür jedoch qualitativ bessere Routen. Im Rahmen von Vorstudien hat sich die reine Optimierung, d.h. die Erzeugung nur einer Route, als nicht effektiv erwiesen. Stattdessen wird eine Vorgehensweise gewählt, bei der in mehreren Iterationen Routen gesammelt werden, deren Kosten pro Iteration abnehmen. Da die Anzahl maximal erzeugter Routen begrenzt wird, ist für diesen Ansatz nicht gesichert, daß stets die optimale Lösung gefunden wird. In einem solchen Fall entspricht das Verhalten dieser Verfahrensvariante demjenigen der Variante ohne Zielfunktion.

Die Entscheidung, ob die Routenbestimmung innerhalb des Pricing-Problems mit oder ohne Zielfunktion erfolgt, berührt andere Gestaltungsentscheidungen nicht und kann daher beliebig kombiniert werden. Die experimentellen Untersuchungen sollen Aufschluß darüber geben, inwiefern sich die Entscheidung für oder gegen eine Zielfunktion in Abhängigkeit von Problem- und anderen Verfahrenscharakteristika auf die Leistungsfähigkeit auswirkt.

6.2.5 Variableninstanziierung

Das Schema der Variableninstanziierung ist die wichtigste Teilentscheidung bei der Festlegung der Suchstrategie. Die Wahl der im nächsten Suchschritt zu instanziierten Variablen bestimmt maßgeblich die Gestalt des Suchbaums, während der zugewiesene Wert über die Suchreihenfolge entscheidet. Das Instanzierungsschema ist insbesondere bei der Verwendung von Bounding-Constraints wichtig, weil diese um so wirksamer sind, je früher eine gute untere Schranke gefunden wird.

Für die Variablenauswahl sind die folgenden Varianten entwickelt und implementiert worden:

first-fail-Prinzip: Bei dieser Vorgehensweise wird die Variable bestimmt, die den kleinsten Wertebereich aufweist. Im Fall gleich großer Wertebereiche wird die Variable mit dem kleinsten Index ausgewählt.

Verkettung: Es wird eine Kette ausgehend vom Startdepot aufgebaut, indem stets diejenige Variable gewählt wird, die im vorangegangenen Schritt als Nachfolger zugewiesen wurde. Auf diese Weise wird gezielt eine Kette vom Startdepot zum Enddepot konstruiert.

Die Variablenauswahl entsprechend des first-fail-Prinzips ist eine im allgemeinen gute und nicht problemspezifische Strategie, während die zweite – nach Kenntnis des Verfassers bislang nicht veröffentlichte – Strategie problemspezifisch für den Aufbau einer Route ist. Im Gegensatz zu anderen bekannten Ansätzen strebt diese Art der Variablenselektion die Instanziierung aller Variablen nur implizit an und instanziiert stattdessen nur eine Teilmenge – entsprechend einer Verkettung von Knoten zu einer zulässigen Route – explizit. Die Motivation für dieses Vorgehen liegt darin, möglichst schnell eine erste Lösung zu erhalten, damit die *lowerbound*-Constraint bereits früh ihre Wirkung entfalten kann und möglichst viele Teile des Suchbaums aufgrund der Abschätzung der unteren Schranke abgeschnitten werden.

Sobald die zu instanziiierende Variable bestimmt wurde, muß der zuzuweisende Wert aus dem Wertebereich gewählt werden. Für diese Entscheidung werden in dieser Arbeit die folgenden Möglichkeiten untersucht:

chronologisch: Die Werte aus dem Wertebereich der betrachteten Variable werden in aufsteigender Reihenfolge zugewiesen.

relaxationsgeleitet: Es wird zunächst versucht, der Variablen den Wert (Nachfolger) entsprechend der Lösung des Zuordnungsproblems aus der *lowerbound*-Constraint zuzuweisen.

minimales Bedauern: Der zuzuweisende Wert wird entsprechend des *Regret*-Wertes ermittelt. Dieser Wert wird anhand der reduzierten Kosten der Lösung des Zuordnungsproblems ermittelt und entspricht dem Index des Knotens, der die zweitniedrigsten reduzierten Kosten aufweist.

Desweiteren wird mit der in Abschnitt 3.3.4 erläuterten *max Regret*-Heuristik eine Instanziierungsmethode implementiert, die Variablen- und Wertbestimmung gleichzeitig durchführt.

Die chronologische Wertzuweisung entspricht dem standardmäßigen Vorgehen, während die relaxationsgeleitete Auswahl als optimistisch interpretiert werden kann, weil sich dahinter implizit die Annahme verbirgt, die getroffenen Zuweisungen führten auch in den weiteren Suchschritten zu zulässigen Lösungen. Die Strategie des minimalen Bedauerns verwendet dieselbe Grundinformation anhand der *lowerbound*-Constraint, ist jedoch „vorsichtiger“. Sowohl die unmittelbare Verwendung der Relaxationslösung als auch die Strategie des minimalen Bedauerns für die Wertzuweisung in der Suchphase sind nach Kenntnis des Autors bislang nicht untersucht worden.

Die Heuristiken zur Auswahl von Variablen und Werten lassen sich zu Instanziierungsstrategien kombinieren. Im Rahmen der experimentellen Bewertung werden verschiedene Instanziierungsschemata einander gegenübergestellt.

6.3 Herstellung ganzzahliger Lösungen: Branching-Schema

Um eine ganzzahlige Lösung zu erhalten, wird der Column Generation-Ansatz in einen Branch-and-Bound-Rahmen eingebunden. Die Strategie, die in dieser Arbeit verfolgt wird, entspricht weitestgehend der Vorgehensweise von (Kohl 1995), die für die gezielte Behandlung alternativer Liefermöglichkeiten um eine erste zusätzliche Phase erweitert wird.

Der Vorteil dieser Branching-Regel liegt zum einen in den positiven Erfahrungen aus der Literatur (vgl. (Kohl 1995; Larsen 1999)) mit dieser Regel, sowie in der guten Integration mit dem Pricing-Problem, da die Branching-Restriktionen mittels Constraint Programming gut abzubilden sind. Gegenüber dem VRPTW weist das kompakte MIP-Modell mit Gleichung (5.28) die Erweiterung um alternative Liefermöglichkeiten auf, deren Gültigkeit in jedem Knoten des Branch-and-Bound-Baumes gewährleistet werden muß. Implizieren demnach die Branching-Regeln für einen Teilbaum die Verwendung einer bestimmten Alternative eines Kunden, muß sich gleichzeitig auch das

Verbot aller anderen Alternativen desselben Kunden aus diesen Regeln ergeben und im Pricing-Problem erzwungen werden.

Für das Lösen einer Instanz des Planungsproblems mit alternativen Liefermöglichkeiten bietet sich eine natürliche Verzweigungsmöglichkeit, wenn – wie in dieser Arbeit – pro Kunde maximal zwei Alternativen zur Auswahl stehen: In der ersten Phase des Branchings wird versucht, auf den Alternativen zu verzweigen und somit zwei Teilbäume zu erzeugen. Es muß daher geprüft werden, ob und ggf. auf welchem Alternativenpaar verzweigt werden soll.

Im ersten Schritt wird dazu die Menge \mathcal{B} der Kunden mit Alternativen gebildet, bei denen eine Auswahl aus den Alternativen im betrachteten Teilbaum noch möglich ist. Hierbei sind vorangegangene Branching-Schritte (die sich ebenfalls auf Alternativen bezogen) zu berücksichtigen. Im zweiten Schritt wird innerhalb dieser Menge das Alternativenpaar gesucht, das ein bestimmtes Kriterium minimiert. Bezeichnet $k^* \in \mathcal{B}$ den zugehörigen Kunden und $\mathcal{A}_{k^*} = \{a_{k^*}, b_{k^*}\}$ die gewählte Alternativenmenge, werden zwei Unterprobleme P_L und P_R erzeugt: Im Problem P_L wird der Besuch des Knotens a_{k^*} und in P_R der Besuch des Knotens b_{k^*} verboten.

Das Kriterium, das zur Auswahl des Knotens k^* verwendet wird, basiert auf der Bestimmung des *Flusses* durch einen Knoten. In der Literatur (vgl. bspw. (Kohl 1995)) wird der Fluß f_{ij} einer Kante $\langle i, j \rangle$ anhand der in der Basis enthaltenen Routen $\mathcal{R}' \subseteq \mathcal{R}$ folgendermaßen bestimmt:

$$f_{ij} = \sum_{\substack{r \in \mathcal{R}' \\ \langle i, j \rangle \in r}} x_r. \quad (6.23)$$

Der Fluß f_{ij} ist ein Maß dafür, wie sehr die Kante $\langle i, j \rangle$ in der aktuellen Lösung des RMP präsent ist. Analog dazu wird in dieser Arbeit der Fluß f_i durch einen Knoten i nach

$$f_i = \sum_{\substack{r \in \mathcal{R}' \\ i \in r}} x_r. \quad (6.24)$$

berechnet. Für jeden Knoten $k \in \mathcal{B}$ mit Alternativenmenge $\mathcal{A}_k = \{a_k, b_k\}$ wird der Wert

$$\phi_k = |f_{a_k} - f_{b_k}| \quad (6.25)$$

bestimmt. Diese Knotenflußdifferenz kann als Maß dafür angesehen werden, wie deutlich für einen Kunden k eine Alternative gegenüber der anderen in der Lösung bevorzugt wird. Kleine Werte von ϕ_k deuten ein ausgeglichenes Verhältnis an, während

große Werte für eine klare Festlegung sprechen. Infolge dieser Interpretation besteht das Auswahlkriterium für k^* in der Wahl von k^* entsprechend

$$k^* = \arg \min_{k \in \mathcal{B}} \phi_k. \quad (6.26)$$

Die Idee besteht also darin, zunächst einmal diejenigen alternativen Liefermöglichkeiten festzulegen, bei denen eine Entscheidung über die Bevorzugung der einen oder der anderen Alternative – da annähernd gleichwertig – schwierig ist.

Handelt es sich bei \mathcal{B} um die leere Menge, d.h. sind die Alternativen in vorangegangenen Branchingschritten sämtlich festgelegt worden und liegt noch keine ganzzahlige Lösung vor, wird in der zweiten und dritten Phase des Branchings entsprechend der in (Kohl 1995) vorgeschlagenen Weise fortgefahren.

Für die zweite Phase wird die Anzahl der Routen (Anzahl der Fahrzeuge) für die Verzweigungsentscheidung herangezogen. Ist die Anzahl der Routen

$$\sum_{r \in \mathcal{R}} x_r = v$$

in der Optimallösung fraktional, wird in zwei Unterprobleme P_L und P_R verzweigt, wobei im Problem P_L die obere Grenze der Fahrzeuganzahl \bar{v} zu

$$\bar{v} = \lfloor v \rfloor \quad (6.27)$$

und im Problem P_R die untere Grenze der Fahrzeugzahl \underline{v} zu

$$\underline{v} = \lceil v \rceil \quad (6.28)$$

verändert werden.

Ist im Gegensatz dazu die Routenanzahl in der Lösung des RMP ganzzahlig und gibt es mindestens eine fraktionale Variable x_r , wird das Branching auf einzelnen Kanten $\langle i^*, j^* \rangle$ ausgeführt. Der Ansatz besteht darin, eine Kante $\langle i^*, j^* \rangle$ zu bestimmen, die in der Optimallösung entweder enthalten ist oder nicht.

Für das Branching ausgewählt wird jene Kante $\langle i^*, j^* \rangle$ mit

$$\langle i^*, j^* \rangle = \arg \max_{\langle i, j \rangle} c_{ij} \cdot \min(f_{ij}, 1 - f_{ij}). \quad (6.29)$$

Wie zuvor, werden zwei Unterprobleme P_L und P_R erzeugt, wobei im ersten die Kante $\langle i^*, j^* \rangle$ in der Lösung enthalten sein muß, im zweiten diese Kante hingegen aus dem Netzwerk entfernt wird.

Die Branching-Regeln müssen in den Unterproblemen und allen darunterliegenden Knoten des Branch-and-Bound-Baumes sowohl für das RMP als auch für die dazugehörigen Subprobleme durchgesetzt werden. Auf Ebene der Restricted Master-Probleme könnten im Prinzip die Spalten entfernt werden, die den Branching-Regeln widersprechen. Damit die linearen Programme durch das Entfernen von Spalten nicht unzulässig werden, versieht man stattdessen ungültige Spalten mit hohen Strafkosten. Auf diese Weise werden die Variablen sehr unattraktiv für die Aufnahme in die Basis.

Die Bestrafung von Variablen ist abhängig von der angewandten Regel. Im Fall des Verbots einzelner Kanten $\langle i^*, j^* \rangle$ werden alle Variablen (Routen) bestraft, die diese Kante enthalten. Die komplementäre Regel bestraft Routen ohne die Kante $\langle i^*, j^* \rangle$ dann, wenn i^* oder j^* auf der Route besucht werden und es sich bei i^* und j^* weder um das Ausgangsdepot noch um das Enddepot handelt. Auf der ersten Stufe, d.h. dem gezielten Verbot einzelner Alternativen, werden stets diejenigen Routen mit Strafkosten versehen, die die jeweils verbotene Alternative enthalten.

Die Behandlung der Branching-Regeln in den Subproblemen ist vergleichsweise einfach und besteht für das Verbot einzelner Kanten lediglich darin, eine Constraint der Form $S_{i^*} = j^*$ bzw. $S_{i^*} \neq j^*$ aufzunehmen. Der Ausschluß von Alternativen wird analog (und mit den oben verwendeten Bezeichnungen) durch Constraints $S_{a_{k^*}} = a_{k^*}$ bzw. $S_{b_{k^*}} = b_{k^*}$ im Pricing-Problem umgesetzt.

Ein weiteres Gestaltungsmerkmal des Branching-Schemas ist die Auswahlmethode für den nächsten zu behandelnden Branch-and-Bound-Knoten. Eingesetzt wird die *Best First*-Methode, bei der die Knoten in der Reihenfolge aufsteigender unterer Schranke bearbeitet werden. Bei diesem Vorgehen handelt es sich um eine verbreitete Heuristik, die (optimistischerweise) davon ausgeht, auf diese Weise möglichst schnell die optimale Lösung zu erhalten.

6.4 Behandlung von Spalten

Die Lösung eines Pricing-Problems ist im Hinblick auf die benötigte Rechenzeit im Vergleich zur Lösungszeit eines LP relativ aufwendig. Aus Vorstudien wurde ersichtlich, daß ca. 85-95% der gesamten Rechenzeit für die Lösung des Pricing-Problems aufgewendet werden.

Es erscheint daher sinnvoll, die Ergebnisse des Pricing-Problems weitestgehend auszunutzen. Zu diesem Zweck werden zunächst pro Aufruf des Subproblems nicht nur eine, sondern alle gefundenen zulässigen Routen mit negativen reduzierten Kosten zurückgegeben und einem Pool hinzugefügt. Sobald ein Pricing Schritt erforderlich ist, werden zunächst die Routen des Pools nach solchen mit negativen reduzierten Kosten durchsucht und gegebenenfalls in das RMP aufgenommen. Aufgenommen werden können solche Routen, die entsprechend der im jeweiligen Suchbaum-Knoten geltenden Bedingungen in der Zielfunktion des RMP nicht mit Strafkosten versehen werden.

Die Suche nach geeigneten Routen im Puffer ist aufgrund der erforderlichen Prüfungen vergleichsweise aufwendig. Die algorithmische Bewertung muß daher der Frage nach dem Nutzen des Spaltenpuffers nachgehen und die Ersparnisse bei der Lösung des Pricing-Problems dem Verwaltungs- und Zugriffsaufwand des Spaltenpuffers gegenüberstellen.

6.5 Verfahrensvarianten

Die vorangegangenen Abschnitte haben für die einzelnen Komponenten des Optimierungsverfahrens Realisierungsalternativen aufgezeigt, die zu Varianten des Gesamtverfahrens zusammengestellt werden. Eine rein schematische Synthese einzelner Verfahrenselemente zu einem Gesamtverfahren erscheint angesichts der im Rahmen von Vorstudien gewonnenen Erfahrungen und Einsichten nicht sinnvoll. Vielmehr haben sich bestimmte Kombinationen von Verfahrenselementen bewährt, während andere sich als wenig leistungsfähig erwiesen haben. Dieser Feststellung wird im folgenden Kapitel vertieft nachgegangen. Abbildung 6.8 verdeutlicht den Aufbau des Verfahrens als Ishikawa-Diagramm.

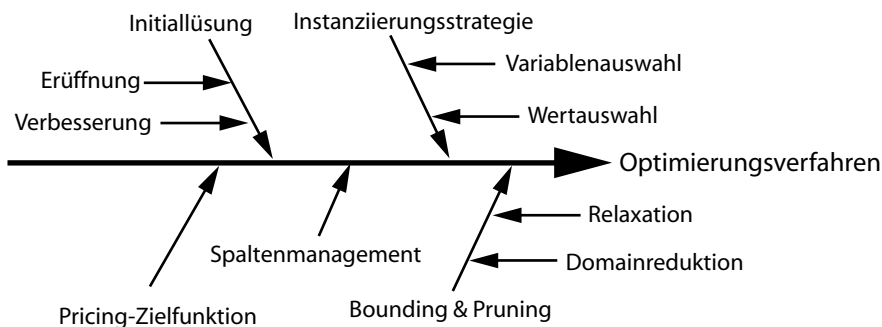


Abbildung 6.8: Schematische Darstellung des Optimierungsverfahrens

6.6 Implementierungsaspekte

Die Implementierung des hybriden IP/CP Verfahrens basiert auf einer Reihe von Bibliotheken. Für den Constraint Programming-Teil wird die C++-Klassenbibliothek ILOG Solver ((ILOG 2001c)) eingesetzt. Die Bibliothek bietet Klassen und Funktionen zur Modellierung von Constraint-Programmen. Über Mechanismen der objektorientierten Programmierung, namentlich der Vererbung und Polymorphie, können eigene globale Constraints und Suchstrategien definiert werden.

Für die heuristischen Verfahren zur Berechnung einer Initiallösung wurde mit der ILOG Dispatcher Bibliothek (s. (ILOG 2001b)) auf eine Erweiterung des ILOG Sol-

vers zur Modellierung und heuristischen Lösung von Routingproblemen zurückgegriffen. Diese Klassenbibliothek bietet typische Modellobjekte, bspw. Fahrzeuge, Kundenbesuche und Entfernungsmaße, und wandelt ein mit diesen Bausteinen erstelltes Modell intern in ein CP Modell um.

Zur Lösung stehen verschiedene Nachbarschaftsstrukturen und Metaheuristiken zur Steuerung der lokalen Suche zur Verfügung.

Für das Branch-and-Price kommt das Framework ABACUS (A Branch A CUt System) aus (Thienel 1995) zum Einsatz. Dieses Framework bildet mathematische Modellgrößen auf C++-Klassen ab und bietet ebenfalls objektorientierte Schnittstellen zur Implementierung eigener Pricing-Algorithmen und Branching-Regeln.

Die Linearen Programme des RMP selbst werden (gekapselt innerhalb ABACUS) durch den LP-Solver CPLEX (s. (ILOG 2001a)) gelöst.

7 Algorithmische und logistische Bewertung

Der Worte sind genug gewechselt, lasst uns endlich Daten sehen!

Gerhard Kocher

Gegenstand dieses Kapitels ist eine Untersuchung der Leistungsfähigkeit der entwickelten Verfahren. Die Untersuchung zielt zum einen auf die Leistungsfähigkeit in methodischer Hinsicht, zum anderen bezieht sich die Untersuchung auf das Konzept alternativer Liefermöglichkeiten selbst, das hinsichtlich seiner logistischen Wirkung beleuchtet werden soll.

Die Bewertung sowohl des algorithmischen als auch des logistischen Aspekts basieren auf Testinstanzen. Die Verwendung derartiger Instanzen ist, wie zuvor dargestellt, eine verbreitete Vorgehensweise zur Analyse von Optimierungsverfahren. Die verwendeten Testinstanzen müssen dabei entsprechend des Untersuchungsziels ausgewählt werden. Für die Tests der Leistungsfähigkeit des Algorithmus empfehlen sich grundsätzlich die Solomon-Instanzen aus (Solomon 1987). Für die Bewertung des VRPA müssen sie jedoch modifiziert werden, weil in den Originaldatensätzen keine Alternativenbeziehungen vorgesehen sind.

Die Ermittlung des Verbesserungspotentials und weiterer logistischer Kenngrößen sollte dagegen auf Datensätzen beruhen, deren Struktur repräsentativ für die B2C-Zustellung ist. Reale Daten sind mangels eines entsprechenden Serviceangebots nicht verfügbar; zufällige Daten weisen ebenso wie die Solomon-Instanzen nicht die geforderte Struktur auf. Daher wurde für diese Arbeit ein geeigneter Szenariengenerator entwickelt.

Ziel der Algorithmentests ist es, die Leistungsfähigkeit des Optimierungsverfahrens – insbesondere die Laufzeit – zu analysieren und daraus Empfehlungen für die Zusammensetzung des Verfahrens aus Einzelementen abzuleiten. Zu diesem Zweck sind Verfahrensvarianten implementiert worden, die gezielt einzelne Teilaspekte des Gesamtverfahrens umsetzen. Neben der Abhängigkeit der Leistungen des Optimierungsverfahrens von einzelnen Verfahrenselementen, soll der Einfluß ausgewählter externer Faktoren.

Das Kapitel ist wie folgt aufgebaut: Im folgenden Unterkapitel werden die algorithmischen Aspekte des Verfahrens untersucht. Zunächst wird das Testumfeld bestehend

aus Testszenarien und Verfahrensvarianten vorgestellt. Den anschließenden Hauptteil bilden die Versuchsreihen, anhand derer einzelnen Fragestellungen gezielt nachgegangen wird. Das zweite Unterkapitel widmet sich der Konzeptbewertung und beschreibt zunächst den Aufbau und die Erzeugung der Szenarien. Darauf aufbauend wird das Potential alternativer Liefermöglichkeiten aus der Perspektive der Logistikdienstleister untersucht.

7.1 Algorithmische Bewertung

Letztlich wird ein Optimierungsalgorithmus – der per Definition ein optimales Ergebnis liefert – an einem Kriterium gemessen: der Lösungsgeschwindigkeit. Bei einer etwas differenzierteren Betrachtung stellen sich jedoch weitere Fragen, deren Beantwortung zu einer Einschätzung eines Verfahrens beitragen kann:

- Welche Verfahrenselemente, bzw. welche Kombinationen von Verfahrenselementen, erweisen sich als effektiv?
- Wie wird die Leistung des Verfahrens bzw. der Verfahrensvariante von der Problemstruktur beeinflusst?
- Damit zusammenhängend: Gibt es Verfahrensvarianten, die sich für bestimmte Problemstrukturen besonders eignen?
- Wie entwickelt sich die Lösungsgüte über die Laufzeit?

Im folgenden Abschnitt wird das Testumfeld beschrieben, mit dem diese Fragen beantwortet werden.

7.1.1 Testumfeld

Auch wenn die Bemühungen in dieser Arbeit darauf abzielten, die Laufzeit der Berechnungen durch bestimmte Verfahrenselemente zu reduzieren, sind die Laufzeiten in vielen Fällen für die Bewertung der Verfahren zu lang. Aus diesem Grund ist die Rechenzeit jedes Berechnungslaufs einer Verfahrensvariante auf 1000 Sekunden begrenzt worden. Die Experimente fanden statt auf einem Rechner mit einem AMD Opteron-Prozessor (Takt von 1,8 GHz)¹ und einem Arbeitsspeicher von 3 GByte unter dem Betriebssystem Microsoft Windows 2000 Server.

¹Tatsächlich handelte es sich um einen Rechner mit zwei Prozessoren dieses Typs, allerdings wurde stets nur ein Prozessor verwendet.

Aufbau der Testinstanzen

Grundlage der Testinstanzen sind die aus der Literatur bekannten Benchmark-Probleme von Solomon. Diese Sammlung von Instanzen wird nach der Problemstruktur in die Klassen C, R und RC aufgeteilt. Während die Standorte in der Klasse C ausgeprägte Cluster bilden, sind sie in der R-Problemlasse zufällig verteilt. Die RC-Klasse stellt entsprechend eine gemischte Klasse dar. Von jeder Klasse existieren wiederum jeweils zwei Varianten, die durch eine an die Klassenbezeichnung angehängte 1 oder 2 gekennzeichnet werden. Die Planungsprobleme der Variante 1 weisen einen kurzen Planungshorizont und engere Zeitfenster auf, so daß pro Fahrzeug nur wenige Standorte besucht werden können. Probleme der Variante 2 haben entsprechend einen längeren Planungshorizont und weniger bindende Zeitfenster. Pro Kombination von Klasse und Variante gibt es bis zu zwölf Instanzen. Diese Instanzen unterscheiden sich nur hinsichtlich der Zeitfensterbreiten, die Lage der Standorte ist identisch. Jedes der Originalprobleme umfaßt 100 Kundenstandorte (zzgl. eines Depotstandorts) deren Lage durch Koordinaten gegeben ist.

Für die experimentelle Untersuchung werden Testinstanzen auf Basis der C1-, R1- und RC1-Probleme erzeugt. Zunächst einmal werden kleinere Probleme mit 25 bzw. 50 Kunden erstellt, indem die jeweils ersten 25 bzw. 50 Kunden des Originaldatensatzes gewählt werden (ebenso das Depot). Um Probleme mit einer verschiedenen großen Anzahl von Kunden mit Alternativen untersuchen zu können, werden aus den reduzierten Problemen weitere Instanzen mit einem Alternativenanteil $r_A \in \{0\%, 20\%, 40\%, 60\%, 80\%, 100\%\}$ generiert. Der Alternativenanteil einer Instanz gibt den Anteil der Kunden mit alternativen Liefermöglichkeiten an der Gesamtzahl der Kunden an. Die Erzeugung erfolgt nach folgendem Schema: Für ein reduziertes Problem mit n Kunden werden für die ersten $n \cdot r_A$ Kunden alternative Liefermöglichkeiten aus dem zugehörigen Originalproblem gewählt. Dabei wird dem ersten Kunden der $n + 1$ -te Standort, dem zweiten Kunden der $n + 2$ -te Standort usw. zugewiesen. Bei dieser Zuordnung werden allerdings nur geographische Lage, Zeitfenster und die Standzeit aus dem Originalproblem übernommen, die Bedarfsmenge nicht. Das folgende Beispiel verdeutlicht das Prinzip.

Beispiel 5 Aus dem Originalproblem R105 soll eine Testinstanz mit 25 Kunden und einem Alternativenanteil von 20% erzeugt werden. Die (ersten) Liefermöglichkeiten sind wie folgt:

Nr.	X	Y	Bedarf	Öffnungszeit	Schließzeit	Standzeit
0 (Depot)	35	35	0	0	230	0
1	41	49	10	151	181	10
2	35	17	7	40	70	10
3	55	45	13	106	136	10
4	55	20	19	139	169	10
5	15	30	26	24	54	10
6	25	30	3	89	119	10
7	20	50	5	71	101	10
⋮	⋮	⋮	⋮	⋮	⋮	⋮
22	45	10	18	87	117	10
23	55	5	29	58	88	10
24	65	35	3	143	173	10
25	65	20	6	156	186	10

Aufgrund des Alternativenanteils von 20% werden 5 weitere Standorte benötigt. Es handelt sich dabei um die Standorte 26 bis 30 von R105, wobei die Bedarfsinformation der Kunden 1 bis 5 übernommen wird.

Nr.	X	Y	Bedarf	Öffnungszeit	Schließzeit	Standzeit
26	45	30	10	122	152	10
27	35	40	7	27	57	10
28	41	37	13	29	59	10
29	64	42	19	53	83	10
30	40	60	26	61	91	10

Die Standorte 1 und 26, 2 und 27 usw. bis 5 und 30 sind damit zueinander alternativ. Die Graphiken in Abbildung 7.1 und 7.2 zeigen die zwei Szenarien R105/25 ohne bzw. mit 5 Alternativen.

Durch den Aufbau der Testinstanzen in dieser Form ist gewährleistet, daß jede Lösung einer Instanz mit niedrigem Alternativenanteil auch zulässig für die (ansonsten gleiche) Instanz mit höherem Alternativenanteil ist.

Um die Übersichtlichkeit in der Auswertung zu steigern, findet eine Aggregation auf drei Klassen von Alternativenanteilen statt. Alternativenanteile von 0% und 20% werden in dieser Klasseneinteilung mit „niedrig“, von 40% bis 60% mit „mittel“ und Quoten von 80% bis 100% mit „hoch“ bezeichnet.

Die Daten der einsetzbaren Fahrzeugflotte, d.h. Anzahl und Kapazität der Fahrzeuge, wird aus den Originalproblemen übernommen und nicht verringert – da die Flotte

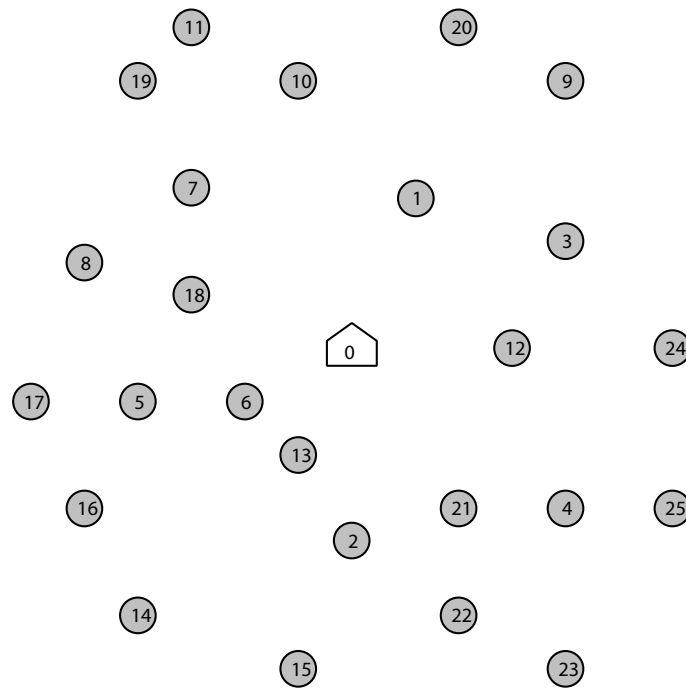


Abbildung 7.1: Solomon-Instanz R105 mit 25 Kunden

für die C1, R1 und RC1-Probleme ohnehin keine beschränkende Ressource darstellt, ist diese Entscheidung unproblematisch. Entfernungen werden über die euklidische Distanz bestimmt, eine Umrechnung in ganzzahlige Werte wird wie in (Kohl 1995) und (Larsen 1999) vorgenommen. Fahrzeiten entsprechen betragsmäßig den Entfernungen, d.h. es wird eine Geschwindigkeit von $60 \frac{km}{h}$ angenommen. Diese Annahme ist sicherlich unrealistisch, an dieser Stelle jedoch unbedenklich, da lediglich eine algorithmische Bewertung erfolgt und auf diese Weise eine Vergleichbarkeit mit bereits veröffentlichten Lösungen der Probleme ohne Alternativen möglich ist.

Infolge der Konstruktionsvorschrift der Solomon-Instanzen bestehen zwischen einzelnen aufeinanderfolgenden Problemen relativ große Ähnlichkeiten. Um den Umfang des Testdatensatzes zu beschränken wurde der Umfang von 348 Datensätzen² auf 108 Datensätze reduziert, indem nur die Problemkonstellationen aus Tabelle 7.1 für die Experimente herangezogen wurden.

Untersuchte Verfahrensvarianten

Für die experimentelle Untersuchung wurden insgesamt 25 implementierte Verfahrensvarianten ausgewählt. Die Varianten unterscheiden sich überwiegend in Details

²Dieser Gesamtumfang ergibt sich aus den Umfängen der einzelnen Klassen (C1: 9, R1: 12, RC1: 8 Probleme) mit 25 bzw. 50 Kunden und Alternativenanteilen 0%,20%,40%,60%,80% und 100%.

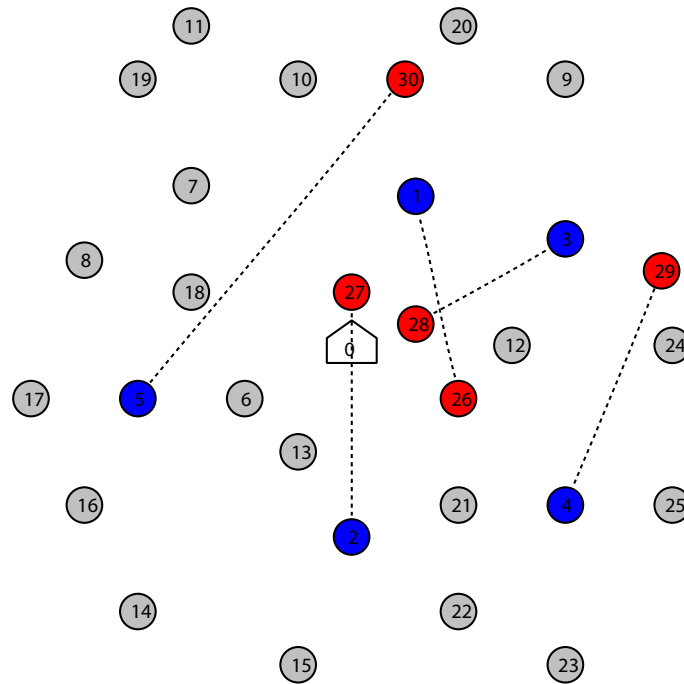


Abbildung 7.2: Modifizierte Solomon-Instanz R105, 25 Kunden und 20% Alternativenanteil, gestrichelte Linien zeigen die Alternativenbeziehung an

Problemklasse	Datensätze	Kundenzahl: Alternativenzahl
C1	C101, C106, C109	25: 0, 5, 10, 15, 20, 25
		50: 0, 10, 20, 30, 40, 50
R1	R101, R105, R109	25: 0, 5, 10, 15, 20, 25
		50: 0, 10, 20, 30, 40, 50
RC1	RC101, RC105, RC107	25: 0, 5, 10, 15, 20, 25
		50: 0, 10, 20, 30, 40, 50

Tabelle 7.1: Ausgewählte Testinstanzen

einzelner Elemente. Tabelle 7.2 gibt eine Übersicht über die getesteten Programmvarianten.

Tabelle 7.2: Untersuchte Verfahrensvarianten

Bezeichnung	Initiallösung	Pricing-Problem					Spaltenmanagement
		Variablenauswahl	Wertzuweisung	Bounding	Pruning	Zielfunktion	
basic	multi+ ^a	first-fail ^b	incr ^c	AP ^d	TW ^e , AE ^f	opt ^g	buffer ^h
basic_additive	multi+	first-fail	incr	AP+MST ⁱ	TW, AE	opt	buffer
basic_feasible	multi+	first-fail	incr	AP	TW, AE	rc < 0 ^j	buffer
basic_multiinit	multi ^k	first-fail	incr	AP	TW, AE	opt	- ^l
basic_multiplusinit	multi+ ^m	first-fail	incr	AP	TW, AE	opt	-
basic_multiplusinit_feasible	multi+	first-fail	incr	AP	TW, AE	rc < 0	-
basic_relaxguided	multi+	first-fail	relax ⁿ	AP	TW, AE	opt	buffer
basic_relax_regret	multi+	first-fail	relax+regret ^o	AP	TW, AE	opt	buffer

(Fortsetzung auf nachfolgender Seite)

^aInitiallösung über mehrere Eröffnungsverfahren, die jeweils anschließend noch Verbesserungsverfahren durchlaufen.^bFirst-Fail-Strategie implementiert über die Wahl der Variablen mit dem kleinsten Wertebereich.^cWahl der Werte aus dem Wertebereich in aufsteigender Reihenfolge.^dBounding mittels Zuordnungsproblem.^eZeitfensterreduktion durch *Timewindow pruning*.^fKanteneeliminierung durch *ArcElimination*.^gOptimallösung, d.h. kürzester Weg, wird gesucht.^hVerwendung des Spaltenpuffers.ⁱAdditive Bounding mit Anwendung des Minimalen Spannenden Baumes auf die Matrix der reduzierten Kosten des Zuordnungsproblems.^jEs werden lediglich Routen mit negativen reduzierten Kosten bestimmt.^kMehrfache Konstruktionsheuristik.^lEs findet kein Spaltenmanagement, d.h. keine Pufferung, statt.^mMehrfache Konstruktionsheuristik und anschließende lokale Suche.ⁿWerteauswahl erfolgt relaxationsgeleitet. Es wird zunächst der Wert entsprechend der Lösung der Relaxation gewählt.^oDer zu instanziierten Variable wird der Wert zugewiesen, der die zweitbesten reduzierten Kosten aufweist.

(Fortsetzung von vorangehender Seite)		Pricing-Problem						Spaltenmanagement
Bezeichnung	Initialisierung	Variablenauswahl		Wertzueweisung	Bounding	Pruning	Zielfunktion	
basic_relax_regret_feasible	multi+	first-fail		relax+regret	AP	TW, AE	rc < 0	buffer
basic_relax_regret_feasible_nobuffer	multi+	first-fail		relax+regret	AP	TW, AE	rc < 0	–
basic_relax_regret_nobuffer	multi+	first-fail		relax+regret	AP	TW, AE	opt	–
basic_savingsinit	savings ^a	first-fail		incr	AP	TW, AE	opt	–
basic_savingsplusinit	savings+ ^b	first-fail		incr	AP	TW, AE	opt	–
basic_tabusearchinit	tabu ^c	first-fail		incr	AP	TW, AE	opt	–
basic_trivialinit	trivial ^d	first-fail		incr	AP	TW, AE	opt	–
max_regret	multi+		max-regret ^e		AP	TW, AE	rc < 0	–
only_apbounding	trivial	first-fail		incr	AP	–	opt	–
only_ancelimination	trivial	first-fail		incr	–	AE	opt	–
only_timepruning	trivial	first-fail		incr	–	TW	opt	–
raw	trivial	first-fail		incr	–	–	opt	–
routebuild_relaxguided	multi+	route ^f		relax	AP	TW, AE	opt	buffer
routebuild_relaxguided_feasible	multi+	route		relax	AP	TW, AE	rc < 0	buffer
routebuild_relaxguided_feasible_nobuffer	multi+	route		relax	AP	TW, AE	rc < 0	–
routebuild_relax_regret_feasible	multi+	route		relax+regret	AP	TW, AE	rc < 0	buffer
routebuild_relax_regret_feasible_nobuffer	multi+	route		relax+regret	AP	TW, AE	rc < 0	–

^aEinfache Konstruktionsheuristik.^bEinfache Konstruktionsheuristik und anschließende lokale Suche.^cEinfache Konstruktionsheuristik und Tabu Search mit max. 100 Iterationen.^dBei dieser Initialisierung wird jeder Knoten durch ein Fahrzeug besucht.^eVariablen- und Wertauswahl entsprechend der *max-Regret*-Heuristik.^fDie Variableninstanziierung erfolgt durch fortgesetzten Routenaufbau.

7.1.2 Experimentelle Untersuchung

Das Optimierungsverfahren entsteht aus der Kombination einzelner Elemente. Die experimentelle Untersuchung geht bezüglich der Gestaltung den folgenden Fragen nach:

- Wie sollte die Initialphase des Verfahrens gestaltet werden: Sollte bevorzugt eine große Anzahl oder eine gute Qualität der Routen angestrebt werden?
- Wie wirkungsvoll sind die einzelnen Bounding- und Pruning-Techniken?
- Wie wirkt sich eine Optimierung im Pricing-Problem gegenüber der einfachen Erzeugung von Routen mit negativen reduzierten Kosten aus?
- Welche Auswirkung hat die Einführung eines Spaltenmanagements?
- Welche Instanziierungsstrategien erweisen sich als vorteilhaft?
- Welcher Zusammenhang besteht zwischen der Problemstruktur und dem Verhalten des Verfahrens?

Im Rahmen von Vorstudien hat sich gezeigt, daß neben der Kundenzahl als Kenngröße des Problemumfangs, der Grad der Clusterung und der Alternativenanteil die maßgeblichen externen Einflußfaktoren sind. Aus diesem Grund findet die Darstellung der Ergebnisse nicht auf der Ebene einzelner Testinstanzen, sondern aggregiert statt. Konkret werden die Auswertungen getrennt nach Problem- und Größenklasse und in Abhängigkeit des Alternativenanteils dargestellt. Die in der jeweiligen Versuchsreihe betrachtete Größe wird dabei als Durchschnittswert über die betreffenden Testinstanzen berechnet.

In den folgenden Abschnitten werden die einzelnen Untersuchungen mit ihren Resultaten vorgestellt.

7.1.3 Lösungsgüte in der Initialphase

Applegate u. a. veröffentlichten im Mai 2004 die optimale Lösung eines 24.978-Städte-Rundreise-Problem für alle Städte Schwedens. Nach Angaben in (Applegate, Bixby, Chvátal, Cook und Helsgaun 2004) handelt es sich dabei um das bis dahin größte optimal gelöste Handlungsreisendenproblem. Bemerkenswert an der Art, wie die Lösung gefunden wurde, ist die Rolle der Eröffnungsheuristik. Die verwendete erweiterte Lin-Kernighan-Heuristik aus (Helsgaun 2000) findet bereits die optimale Lösung, den weitaus größten Teil der Laufzeit benötigt anschließend jedoch der Branch-and-Cut-Teil des Verfahrens, um die Optimalität zu bestätigen.

Es liegt daher nahe zu untersuchen, inwiefern bei der Lösung des VRPA ein vergleichbares Verhalten erkennbar ist. Die Untersuchung soll dazu dienen, eine Empfehlung für die Art der in der Initialphase zu verwendenden Heuristiken abzugeben. Zu diesem Zweck werden die Varianten

- `basic_trivialinit`,
- `basic_savingsinit`,
- `basic_savingsplusinit`,
- `basic_multiinit`,
- `basic_multiplusinit` und
- `basic_tabusearchinit`

verwendet und für alle mit diesen Verfahren optimal gelösten Testinstanzen bestimmt, welcher Anteil der Routen aus der Optimallösung bereits während der Initialphase bestimmt wurde. Die Ergebnisse für die 25-Kunden-Probleme werden in den Abbildungen 7.3 bis 7.5 wiedergegeben.

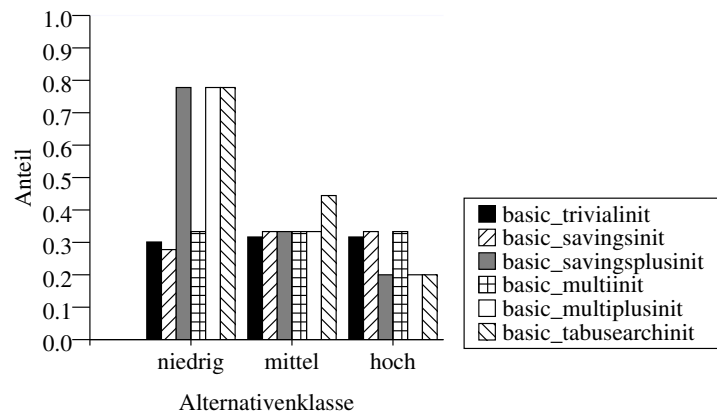


Abbildung 7.3: Anteil der in der Initialphase gefundenen Routen aus der Optimallösung (C1, 25 Kunden)

In Abbildung 7.3 wird deutlich, daß die Verwendung von Verbesserungsverfahren gegenüber reinen Eröffnungslösungen bei einem niedrigen Alternativenanteil vorzuziehen ist. Bei mittlerem und hohem Alternativenanteil schwindet der Vorteil, was darauf zurückzuführen ist, daß sich kein lokales Suchverfahren explizit den alternativen Liefermöglichkeiten widmet. Verbesserungsverfahren sind bei den C1-Problemen auch deswegen besser, weil diese Probleme von Solomon nach Konstruktion auf derartige Verfahren zugeschnitten worden sind. Bei zufälligen Standorten (s. Abbildung 7.4) zeigt sich ein deutlicher Vorteil für die Kombination mehrerer Eröffnungs- mit Verbesserungsverfahren. Im Vergleich zu Tabu-Search erweist sich die Kombination als durchgängig besser, wobei der Vorteil mit steigendem Alternativenanteil sinkt. Aus der Betrachtung der Ergebnisse für das RC1-Problem in Abbildung 7.5 ergibt sich

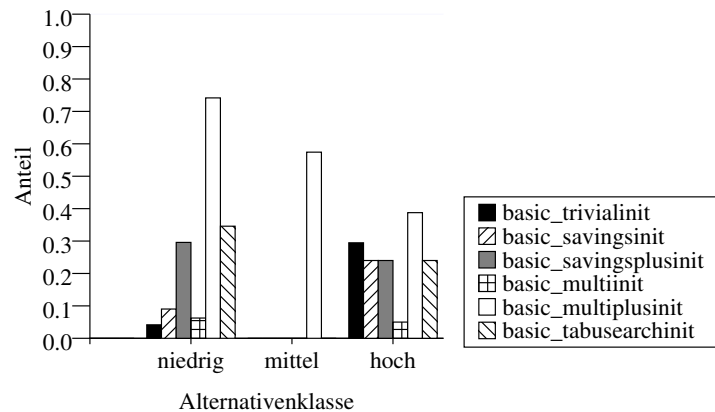


Abbildung 7.4: Anteil der in der Initialphase gefundenen Routen aus der Optimallösung (R1, 25 Kunden)

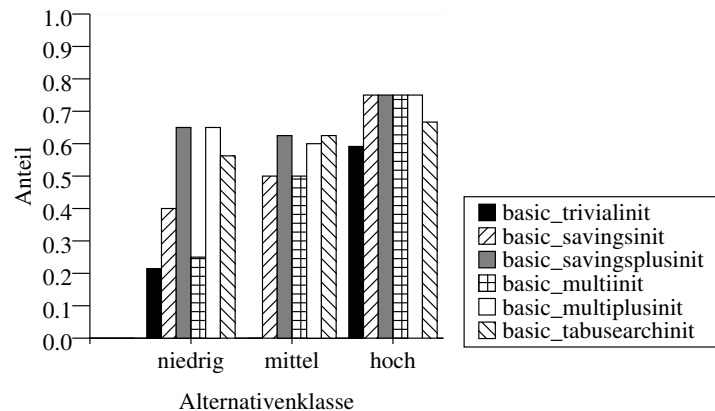


Abbildung 7.5: Anteil der in der Initialphase gefundenen Routen aus der Optimallösung (RC1, 25 Kunden)

kein eindeutiges Bild, die folgenden Schlußfolgerungen lassen sich jedoch insgesamt ziehen:

- Die Verwendung von Verbesserungsverfahren erweist sich gegenüber dem gleichen Verfahren ohne lokale Suche als besser,
- Anstelle der Verwendung von Metaheuristiken erscheint es besser, eine größere Zahl von Eröffnungslösungen zu bestimmen.

Daß die Verfahrensvariante `basic_multiplusinit` in der Tat eine größere Anzahl von Routen erzeugt, wird in den Abbildungen 7.6 bis 7.8 deutlich, in denen der Anteil

der in der Initialphase erzeugten Routen in Relation zur Gesamtzahl der erzeugten Routen gesetzt wird.

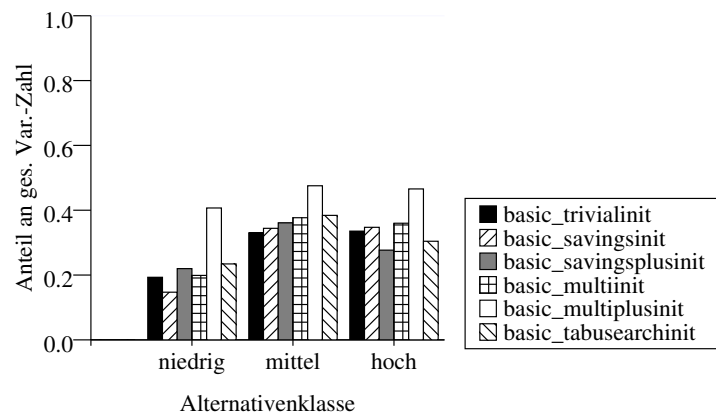


Abbildung 7.6: Anteil der in der Initialphase erzeugten Routen an der Gesamttroutenzahl (C1, 25 Kunden)

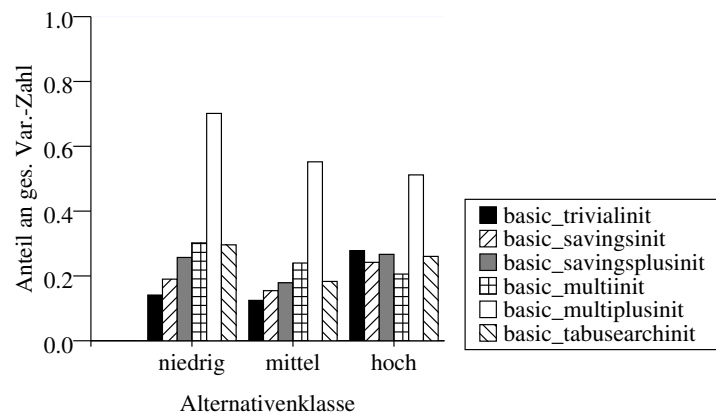


Abbildung 7.7: Anteil der in der Initialphase erzeugten Routen an der Gesamttroutenzahl (R1, 25 Kunden)

Aus den Diagrammen ist ersichtlich, daß die Tabu-Search-Variante vergleichsweise ökonomisch zu Werke geht: Während die Anzahl der in der Initialphase erzeugten Routen aus der Optimallösung oftmals mit derjenigen der `basic_multiplusinit`-Variante vergleichbar ist, werden relativ weniger Routen erzeugt. Insbesondere für hohe Alternativenanteile könnte Tabu-Search daher von einem lokalen Suchverfahren zur Auswahl der zu besuchenden Alternativen profitieren, weil die erhöhte Qualität der innerhalb der begrenzten Iterationszahl erzeugbaren Routen die geringere Anzahl kompensieren könnte.

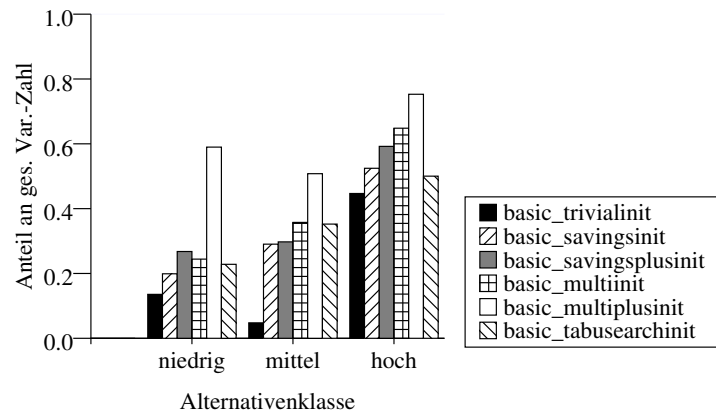


Abbildung 7.8: Anteil der in der Initialphase erzeugten Routen an der Gesamtroutenzahl (RC1, 25 Kunden)

7.1.4 Einfluß der Initialphase auf die Gesamtlaufzeit

Der vorangegangene Abschnitt zeigte, daß sich die Güte und Anzahl der in der Initialphase erzeugten Routen, über die verschiedenen Varianten betrachtet, unterscheiden. Es stellt sich daraufhin die Frage, inwiefern dies auch Einfluß auf die Gesamtlaufzeit des Verfahrens hat. Um diese Frage zu klären ist in den Abbildungen 7.9 bis 7.11 jeweils die Gesamtlaufzeit der Verfahren wiedergegeben.

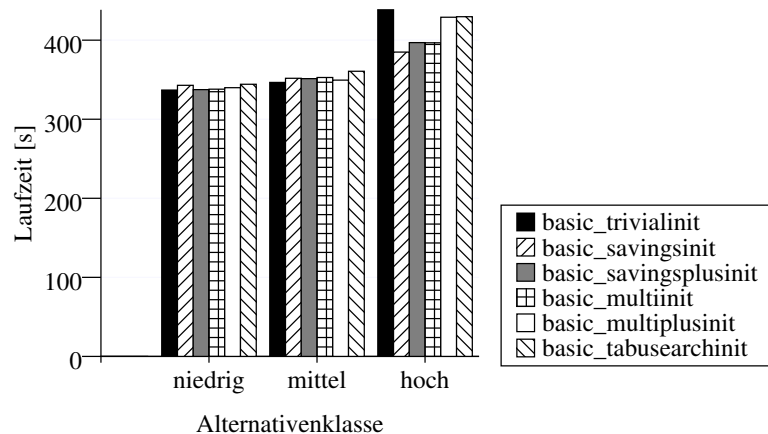


Abbildung 7.9: Gesamtlaufzeit in Abhängigkeit der Initialphase (C1, 25 Kunden)

Die Diagramme zeigen, daß die Laufzeitunterschiede minimal sind und eher auf die Problemstruktur denn auf die verwendete Initialisierungsstrategie zurückzuführen sind. Bei den C1-Problemen ist dies sehr wahrscheinlich auf das Konstruktionsprinzip

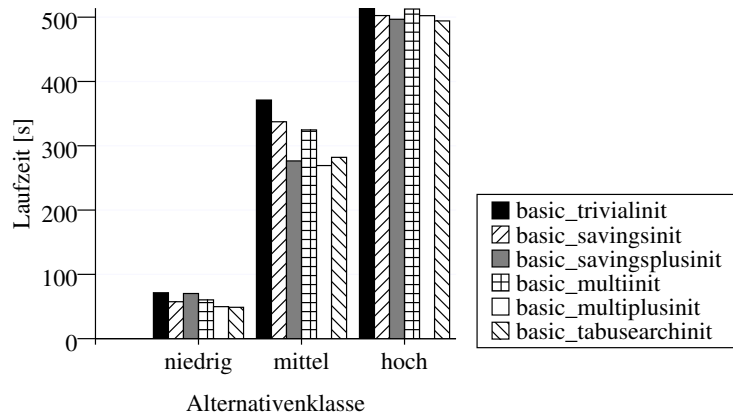


Abbildung 7.10: Gesamtlaufzeit in Abhängigkeit der Initialphase (R1, 25 Kunden)

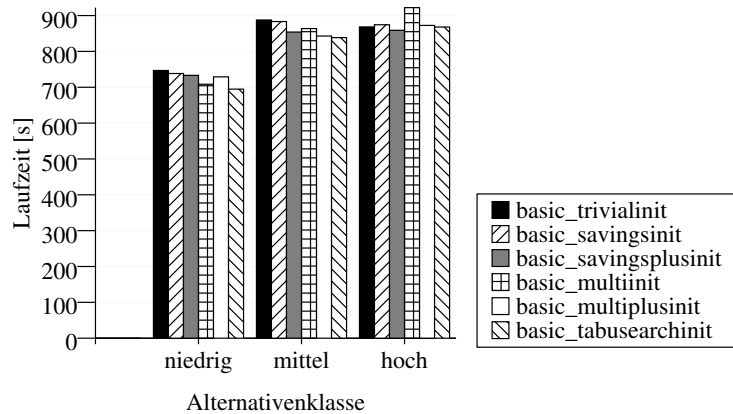


Abbildung 7.11: Gesamtlaufzeit in Abhängigkeit der Initialphase (RC1, 25 Kunden)

zurückzuführen, das es auch einfachen Heuristiken erlaubt, gute und sogar optimale Lösungen zu bestimmen. Die durchschnittliche Laufzeit in der Nähe der Abbruchgrenze von 1000 Sekunden deutet für RC1-Probleme darauf hin, daß hier nicht die Initialphase der bestimmende Faktor ist, sondern die eigentliche Optimierung, die viele Iterationen benötigt, um die Optimalität zu bestätigen. Lediglich die R1-Probleme zeigen für Instanzen der mittleren Alternativenanteilklasse, daß eine gute Initia llösung auch hinsichtlich der Gesamtlaufzeit vorteilhaft ist. Daß die Laufzeiten der verschiedenen Varianten für Probleme mit hohem Alternativenanteil ähnlich sind, kann wieder auf das Fehlen guter lokaler Suchverfahren für das VRPA zurückgeführt werden.

7.1.5 Wirkung der Bounding und Pruning-Techniken

Um die Wirkung von Timewindow-Pruning, Kanteneliminierung und Bounding einzuschätzen, wird die Anzahl der Backtracking-Schritte gezählt. Eine niedrige Zahl von Backtracking-Schritten bedeutete eine niedrige Zahl von Fehlentscheidungen im Suchprozess des Constraint Programms zur Routensuche. Von besonderem Interesse ist der Effekt der Kombination mehrerer Techniken. Zur Veranschaulichung der Analyse sind in Abbildung 7.12 bis 7.14 drei Messungen für kleine Problemumfänge dargestellt.

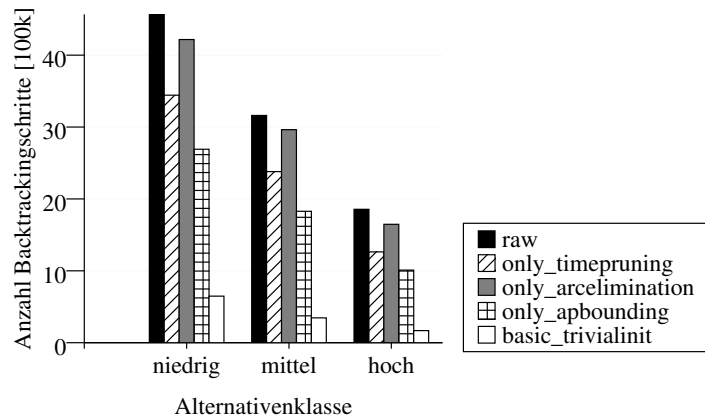


Abbildung 7.12: Backtrackingverhalten (C1, 25 Kunden)

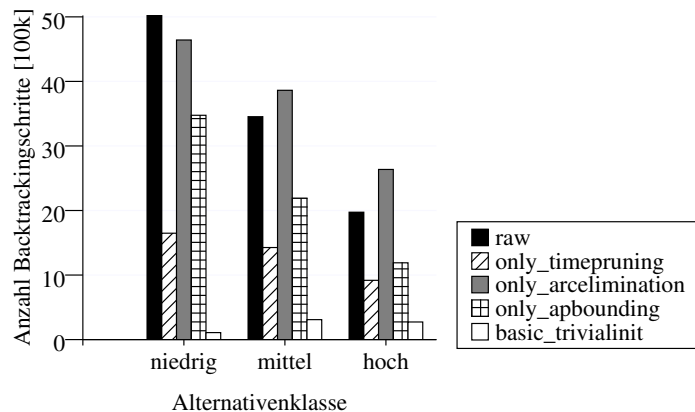


Abbildung 7.13: Backtrackingverhalten (R1, 25 Kunden)

Zunächst einmal wird deutlich, daß eine Erhöhung des Alternativenanteils zu einer absoluten Reduktion der Anzahl von Backtracking-Schritten führt. Eine Erklärung kann

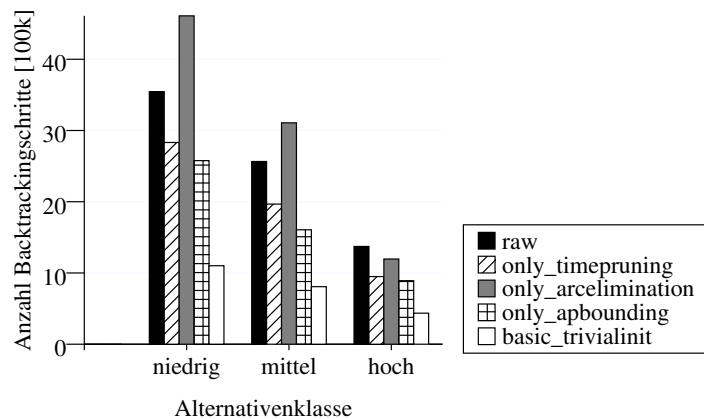


Abbildung 7.14: Backtrackingverhalten (RC1, 25 Kunden)

darin liegen, daß eine größere Anzahl von Alternativen den Lösungsraum um offensichtlich gute Standorte erweitert, bzw. daß ungünstige Standorte früh erkannt und von der weiteren Betrachtung ausgeschlossen werden können (bspw. anhand der Gradienteninformation im Bounding).

Die Kombination mehrerer Techniken (Variante `basic_trivialinit`) erreicht eine sehr starke Reduktion der Anzahl von Backtracking-Schritten (in den Diagrammen gemessen in Einheiten von 100.000) im Vergleich zu einer Verfahrensvariante ohne derartige Verfahrenselemente (Variante `raw`). Dieses Verhalten ist nachvollziehbar, da jede Verkleinerung des Suchbaums den anderen Constraints über den Propagationsmechanismus mitgeteilt wird und somit eine weitere Reduktion anstoßen kann.

Das Verhalten der `ArcElimination`-Constraint muß genauer analysiert werden, was jedoch für weitere Forschungsarbeiten über diese Arbeit hinaus vorgemerkt wird. Es kann jedoch festgestellt werden, daß zuweilen ein kontraproduktives Verhalten zu beobachten ist. Die Messergebnisse deuten darauf hin, daß die Constraint wahrscheinlich zu häufig aufgerufen wird und in verschiedenen Suchknoten die gleiche zu einem Backtracking-Schritt führende Reduktion wiederholt, während bei weniger aggressiver Vorgehensweise dies seltener geschieht. Die Diagramme in Abbildung 7.13 und 7.14 lassen vermuten, daß insbesondere eine Erhöhung des Alternativenanteils sich negativ auswirkt.

Die Reduktion der Kanten des Netzwerks durch die Entfernung zeitlich ungültiger Verbindungen, wie dies die `Timewindow-Pruning-Constraint` durchführt, erweist sich bei zufälligen Standorten als besonders hilfreich. Für niedrige Alternativenanteile ist damit eine praktisch backtrackingfreie Suche möglich.

Die Verwendung der `Additive Bounding`-Technik ist nicht vorteilhaft. Wie Abbildung 7.15 stellvertretend zeigt, wird die Laufzeit nicht gesenkt, d.h. die erhöhte Qualität der unteren Schranke setzt sich gegenüber dem erhöhten Berechnungsaufwand nicht

durch.

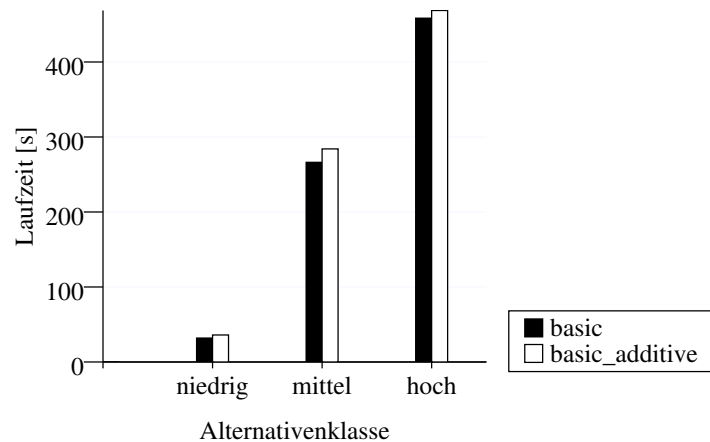


Abbildung 7.15: Vergleich der Laufzeiten für Bounding und Additive Bounding (R1, 25 Kunden)

Aus diesem Teil der Leistungsanalyse können insgesamt folgende Schlüsse gezogen werden:

- Die Wirkung verschiedener Bounding- und Pruningtechniken verstärkt sich gegenseitig,
- im Fall der ArcElimination-Constraint sollte eine strukturabhängige Gestaltung in Betracht gezogen werden,
- die Wirkung der Bounding- und Pruningtechniken ist abhängig von der Problemstruktur und dem Alternativenanteil,
- eine Erhöhung des Alternativenanteils senkt die Anzahl der Backtracking-Schritte.

7.1.6 Zielkriterium des Pricing-Problems

Zwei mögliche Zielkriterien im Pricing-Problem werden unterschieden:

- Die Optimierung, d.h. Minimierung der Routenlänge, oder
- die Beschränkung auf negative reduzierte Kosten.

Mit dem Ziel festzustellen, welchem Zielkriterium generell der Vorzug zu geben ist, sind drei Verfahrensvarianten (jeweils in zwei Untervarianten entsprechend der Zielkriterien) anhand der Gesamtlaufzeit verglichen worden. Abbildungen 7.16 bis 7.18 geben die Resultate wieder.

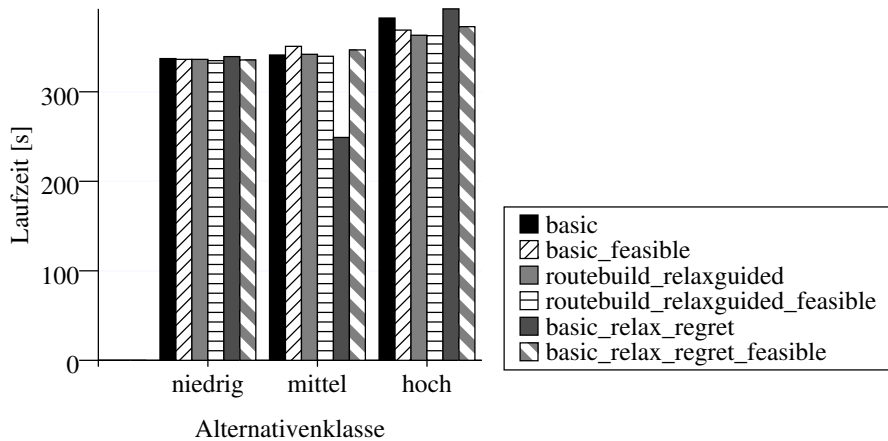


Abbildung 7.16: Vergleich verschiedener Varianten mit bzw. ohne Zielfunktion im Pricing-Problem (C1, 25 Kunden)

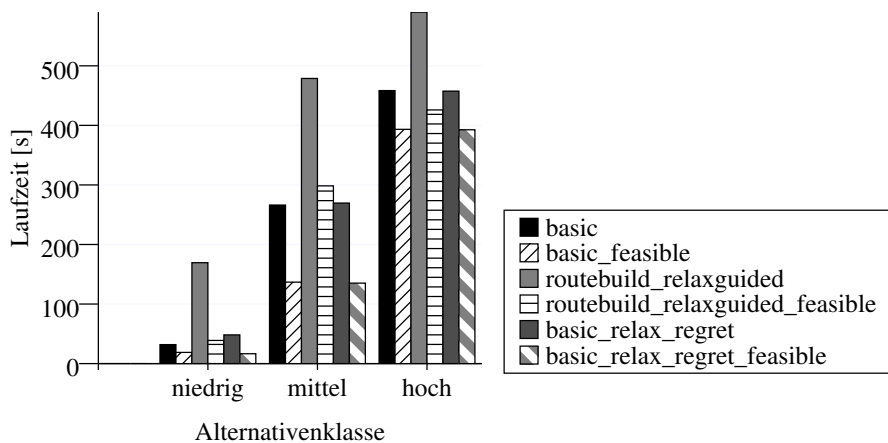


Abbildung 7.17: Vergleich verschiedener Varianten mit bzw. ohne Zielfunktion im Pricing-Problem (R1, 25 Kunden)

Die Ergebnisse unterscheiden sich deutlich je nach Problemstruktur. Während für die C1-Probleme unabhängig von der Alternativenzahl fast keine Veränderungen feststellbar sind, wächst die Laufzeit aller Verfahren für die R1-Probleme mit zunehmender Alternativenzahl. Auch für RC1-Probleme steigt die Laufzeit entsprechend, hier sind jedoch die Begrenzungen der Rechenzeit für die nicht so starke Zunahme der Laufzeit bei hohem Alternativenanteil verantwortlich.

Die Laufzeit der Varianten ohne Optimierung ist mit einer Ausnahme durchgängig niedriger und somit besser. Für Probleme, die keine ausgesprochene Clusterstruktur aufweisen, verringert sich tendenziell der Unterschied zwischen den Varianten mit bzw. ohne Zielfunktion mit zunehmender Alternativenzahl. Als Ursache dafür kann

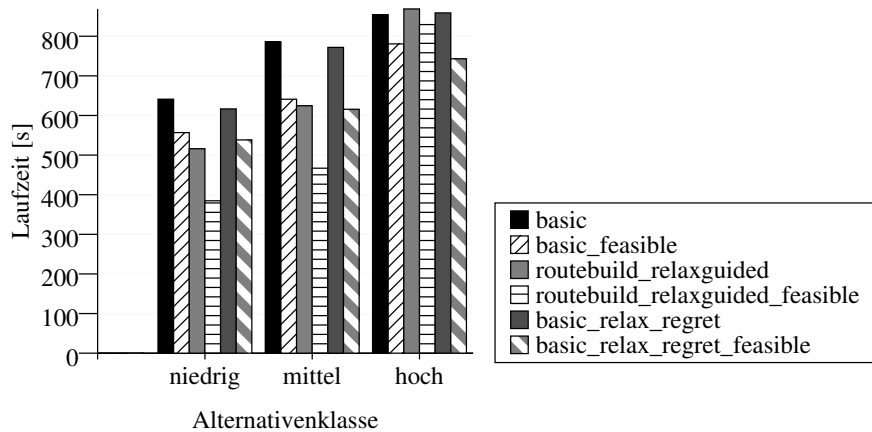


Abbildung 7.18: Vergleich verschiedener Varianten mit bzw. ohne Zielfunktion im Pricing-Problem (RC1, 25 Kunden)

die höhere Routenqualität der Optimierungsvarianten gesehen werden, die bei größerer Alternativenzahl zwar wichtiger wird, den Geschwindigkeitsnachteil allerdings nicht kompensieren kann.

Im Ergebnis bleibt festzuhalten, daß die untersuchten Verfahrensvarianten von einer Optimierung im Pricing-Problem nicht profitieren, sondern sich vielmehr die vergleichsweise schnelle Erzeugung vieler Routen als besser erweist.

7.1.7 Beitrag des Spaltenmanagements

An der Schnittstelle zwischen Master- und Pricing-Problem kann ein Spaltenpuffer die Zwischenspeicherung einmal erzeugter Routen übernehmen, um diese im Pricing-Schritt falls zulässig und potentiell verbessernd wieder zur Verfügung zu stellen. Analog zu Abschnitt 7.1.6 werden drei Verfahrensvarianten in jeweils zwei Untervarianten – mit bzw. ohne Spaltenpufferung – miteinander verglichen. Die Ergebnisse finden sich in den Abbildungen 7.19 bis 7.21.

Die Wirkung des Spaltenpuffers ist im allgemeinen nur gering. Der Verwaltungsaufwand für die Aufnahme neuer Routen und das Finden zulässiger und verbessernder Routen kann eine Ursache für den geringen zu beobachtenden Effekt sein. Da der Verwaltungsaufwand von der Anzahl der zu behandelnden Routen abhängt, ist es naheliegend an dieser Stelle kurz zu beleuchten, welchen Beitrag der Spaltenpuffer im Fall der Optimierung des Pricing-Problems leistet. Zu diesem Zweck werden die entsprechenden Untervarianten des `basic_relax_regret`-Verfahrens miteinander verglichen. Abbildung 7.22 zeigt stellvertretend für andere Problemkonstellationen, daß der Nutzen des Spaltenpuffers uneinheitlich ist. Bei einer hohen Alternativenzahl ist der Beitrag des Spaltenpuffers bei einer Optimierung im Pricing-Problem deutlich grö-

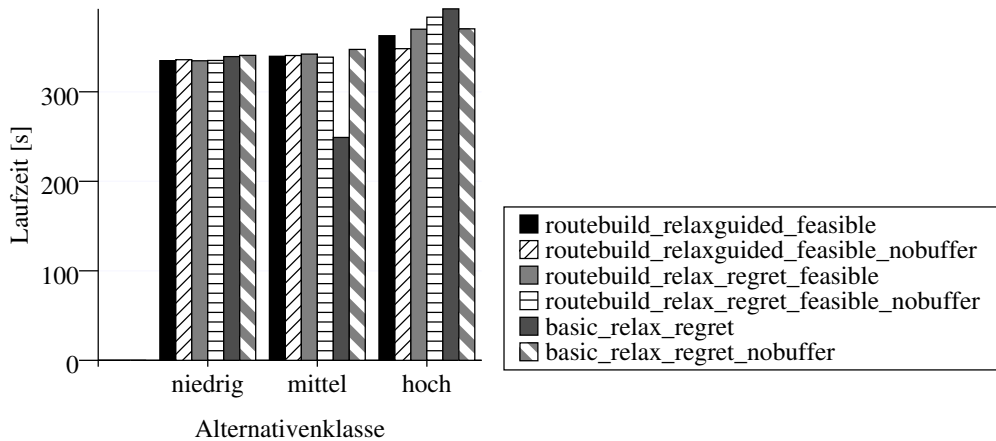


Abbildung 7.19: Vergleich der Laufzeit für Varianten mit bzw. ohne Spaltenpuffer (C1, 25 Kunden)

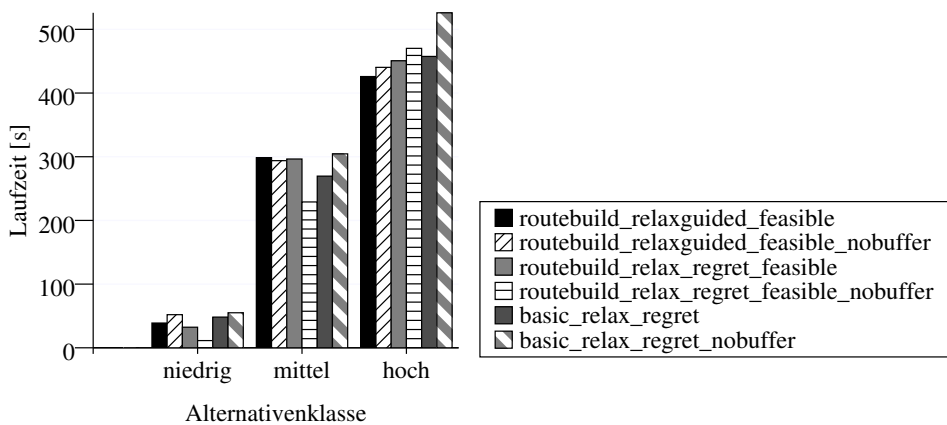


Abbildung 7.20: Vergleich der Laufzeit für Varianten mit bzw. ohne Spaltenpuffer (R1, 25 Kunden)

ßer als bei nur zulässigen Routen. Dies kann als Indiz für die obige Argumentation anhand des erhöhten Verwaltungsaufwands angesehen werden. Bei einer mittleren Alternativenanteil kann hingegen eine geringfügig längere Laufzeit der Variante mit Pufferung bei Optimierung und eine kürzere Laufzeit ohne Optimierung festgestellt werden.

Dieses Ergebnis spricht dafür, daß im Fall der Optimierung der Puffer erfolglos durchsucht wird. Dabei fällt der Suchaufwand an, ein Ertrag in Form von wiederverwendbaren Routen jedoch nicht. Anders bei nur zulässigen Routen. Der Umfang des Puffers ist in diesem Fall derart, daß die durch Wiederverwendung ersparte Rechenzeit den Suchaufwand positiv überwiegt.

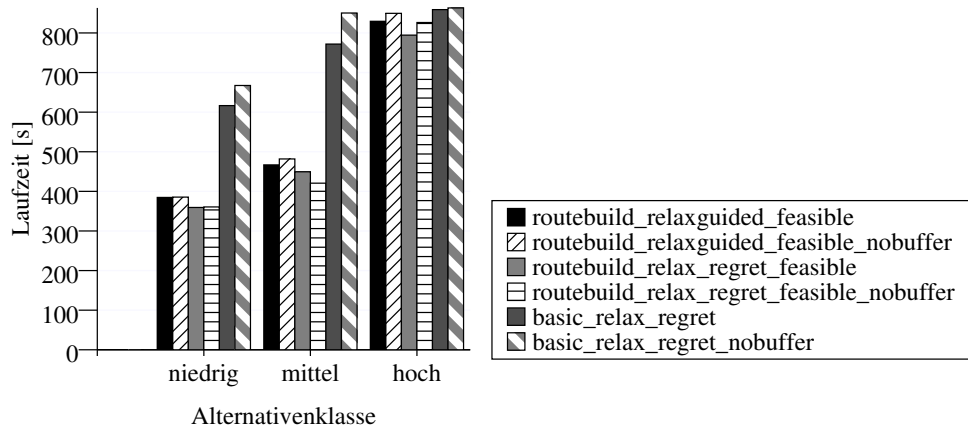


Abbildung 7.21: Vergleich der Laufzeit für Varianten mit bzw. ohne Spaltenpuffer (RC1, 25 Kunden)

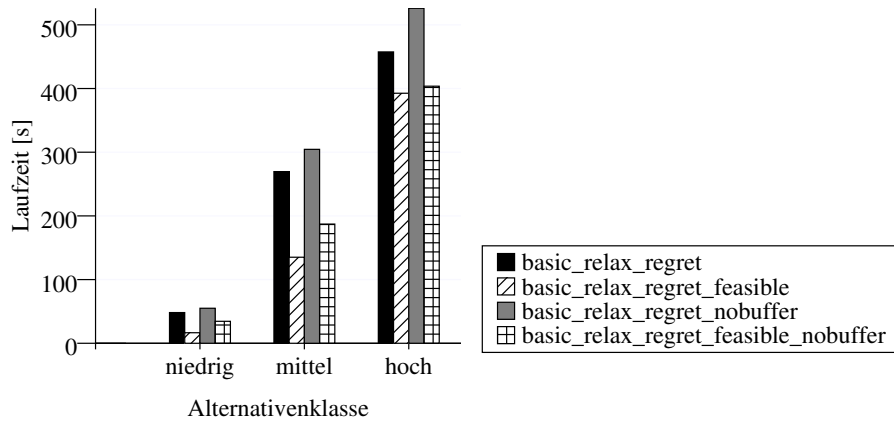


Abbildung 7.22: Zusammenhang zwischen Beitrag des Spaltenpuffers und Zielkriterium des Pricing-Problems (R1, 25 Kunden)

Die Quintessenz der Analyse des Spaltenmanagements geht dahin, daß die Skalierbarkeit noch verbesserungswürdig ist. Für kleinere Problemumfänge ist ein durchaus nennenswerter positiver Beitrag zu erkennen. Sobald der Umfang der gespeicherten Routen jedoch wächst, ist die Verwaltung des Spaltenpuffers ein bremsender Engpaß. Auch aus dieser Erkenntnis soll für weitere Arbeiten Forschungsbedarf abgeleitet werden.

7.1.8 Vergleich von Instanziierungsstrategien

Ein wichtiges Gestaltungselement im Pricing-Problem ist die Wahl der Instanziierungsstrategie. Fünf Verfahrensvarianten in einer Form ohne die Verwendung von Spaltenpuffern und mit optimierendem Pricing-Problem werden verglichen. Die Resultate zeigen die Abbildungen 7.23 bis 7.25.

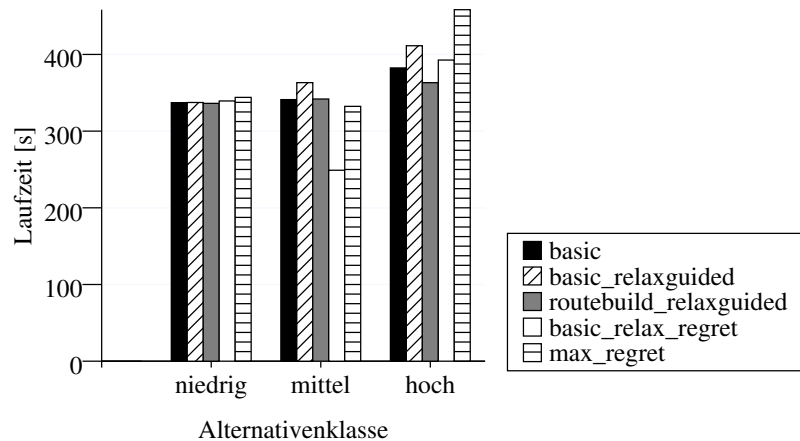


Abbildung 7.23: Vergleich der Instanziierungsstrategien (C1, 25 Kunden)

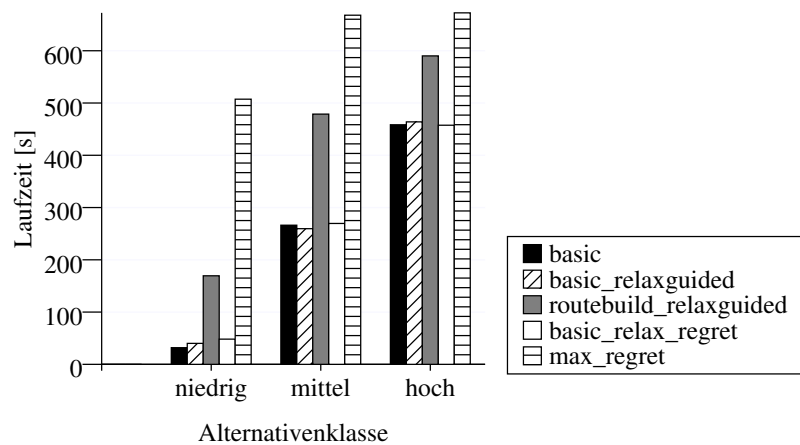


Abbildung 7.24: Vergleich der Instanziierungsstrategien (R1, 25 Kunden)

Während die Ergebnisse für stark geclusterte Probleme in etwa vergleichbar und auch größtenteils unabhängig von der Alternativenanzahl sind (s. Abbildung 7.23), variieren die Resultate bei stärker zufälliger Struktur. Weiterhin gleichen sich die Laufzeiten der Verfahren für höhere Alternativenzahlen einander wieder mehr an, nachdem sich für mittlere Alternativenzahlen deutliche Unterschiede zeigen.

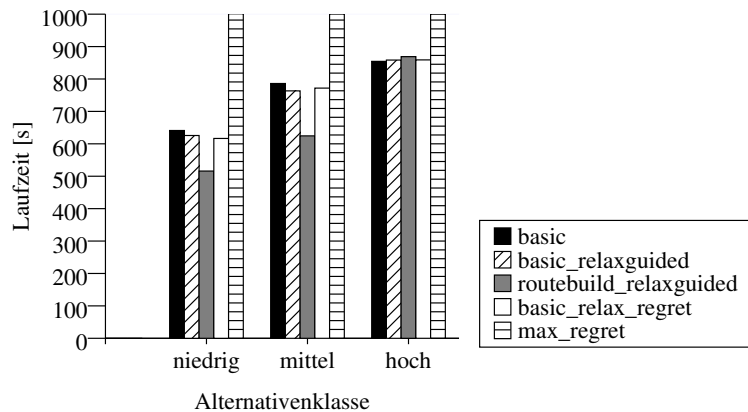


Abbildung 7.25: Vergleich der Instanziierungsstrategien (RC1, 25 Kunden)

Aus Abbildungen 7.24 und 7.25 wird ersichtlich, daß die max-Regret-Heuristik im Vergleich zu den anderen Strategien deutlich schlechter ist.

Im Vergleich der Verfahren `basic_relaxguided` und `basic_relax_regret` ist (bis auf den Sonderfall der C1-Probleme mit mittlerer Alternativenzahl) kein spürbarer Unterschied in der Laufzeit feststellbar. Beide Verfahren sind darüberhinaus gleichwertig zum klassischen Ansatz der `basic`-Variante.

Die Instanziierungsstrategie des Routenaufbaus verhält sich in den Versuchen erkennbar anders als die first-fail-Varianten. Während für zufällige Probleme die Laufzeit deutlich länger ist, ist sie für die Klasse der RC1-Probleme kürzer und wächst beim Übergang von niedrigen auf mittlere Alternativenanteile kaum an.

Aufgrund dieses unterschiedlichen Verhaltens im Vergleich zu den first-fail-Verfahrensvarianten, sollen an dieser Stelle weitere Varianten der Routenaufbaustrategie betrachtet werden. Abbildung 7.26 gibt einen Überblick, der auch analog für andere Probleme gilt.

Die Messungen zeigen, daß es – wie im Fall der first-fail-Varianten – keinen großen Einfluß auf die Laufzeit hat, ob die beste oder zweitbeste Lösung der Relaxation für die Wertzuweisung verwendet wird. In der Tendenz ist allenfalls ein kleiner Vorteil für die vorsichtigeren Strategie des minimalen Bedauerns erkennbar. In diesem Zusammenhang ist es auch interessant festzustellen, daß die Strategie des maximalen Bedauerns sich fast durchgängig als schwächer im Vergleich zu den Entwicklungen aus dieser Arbeit gezeigt hat.

7.1.9 Analyse des Branching-Verhaltens

Um ganzzahlige Lösungen zu erhalten, wird in der ersten Phase des Branchings auf Alternativen verzweigt. In diesem Abschnitt wird ausgewertet, welches Verzweigungs-

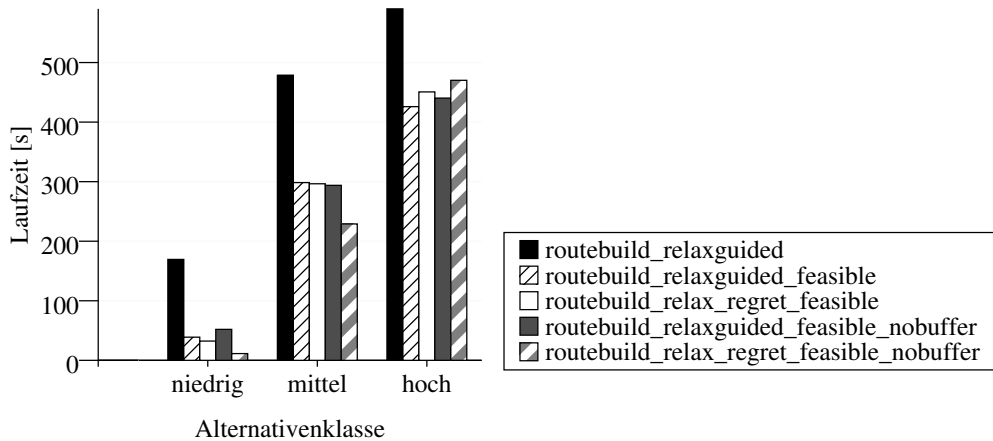


Abbildung 7.26: Varianten der Routenaufbau-Strategie (R1, 25 Kunden)

verhalten sich insgesamt einstellt. Dazu wird die Häufigkeit der Verwendung der einzelnen Branching-Regeln untersucht.

Während für stark geclusterte Problemstrukturen ein Branching für die 25- bzw. 50-Kunden-Probleme nie erforderlich ist, weisen die R1- bzw. RC1-Probleme ein interessantes Verhalten auf.

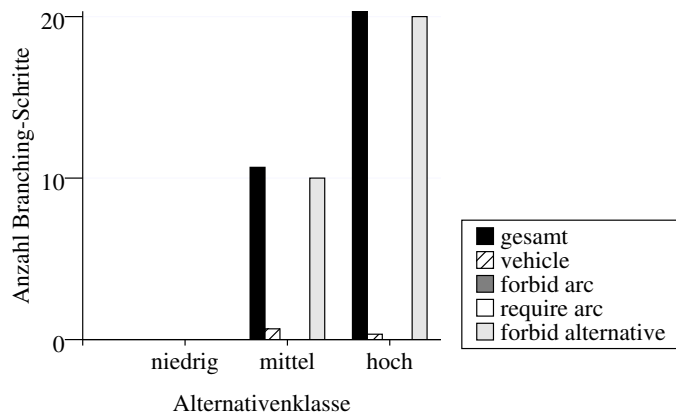


Abbildung 7.27: Häufigkeit der verwendeten Branching-Regeln (R1, 25 Kunden)

Abbildungen 7.27 und 7.28 zeigen, daß ab einer gewissen Mindestmenge von Alternativen keine Verzweigungen auf Kanten des Netzwerkes erforderlich sind und die speziell eingeführte Verzweigungsstufe auf Alternativen und die bekannte Stufe der Verzweigung auf die Fahrzeuganzahl für den Erhalt ganzzahliger Lösungen ausreichend sind. Der Grund dafür liegt darin, daß, sobald die Entscheidung für bestimmte

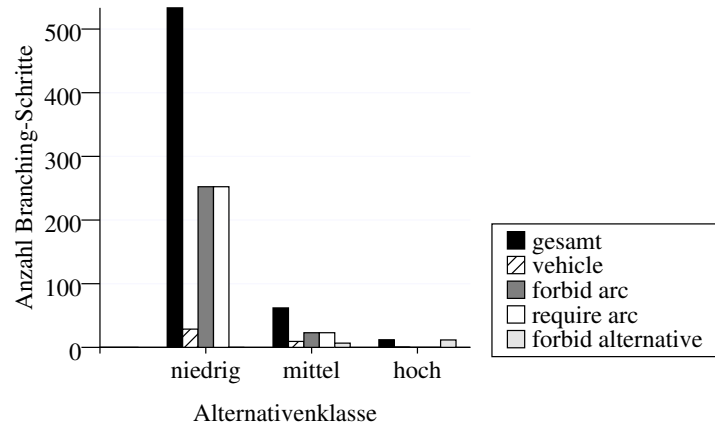


Abbildung 7.28: Häufigkeit der verwendeten Branching-Regeln (RC1, 25 Kunden)

alternative Liefermöglichkeiten getroffen worden ist, sich deutliche Clusterstrukturen herausbilden und damit die einzelnen Touren feststehen. Wenn jeder Kunde somit nur in einer Tour auftritt, ist die ganzzahlige Lösung des Master-Problems gesichert. Durch die bereits teilweise vorhandene Clusterstruktur in der RC1-Klasse sind hier bei hoher Alternativenzahl insgesamt weniger Verzweigungsschritte als bei den R1-Problemen erforderlich.

7.1.10 Zusammenhang zwischen Problemstruktur und Laufzeit

Die zentrale Frage bei der Lösung des VRPA ist die nach dem Einfluß der Alternativen auf die Laufzeit. In Abschnitt 2.5 wurde angeführt, daß sich der Lösungsraum bei zwei Alternativen für p Kunden bereits um einen Faktor 2^p vergrößert. Die in den Abbildungen 7.23 bis 7.25 wiedergegebenen Laufzeiten lassen jedoch erkennen, daß diese Steigerung der Lösungsraumgröße sich nicht voll auf die Laufzeit auswirkt. Stattdessen ist eine Abhängigkeit der Laufzeitzunahme von der räumlich/zeitlichen Struktur erkennbar.

Weist ein Planungsproblem bereits einen großen Grad an Clusterbildung auf, sind hinzukommende alternative Liefermöglichkeiten nur in seltenen Fällen so attraktiv, daß eine neue Tourstruktur zu geringeren Kosten möglich ist. Die hinzugekommenen Alternativen werden demnach vergleichsweise schnell verworfen, so daß sich ein Gesamtaufwand vergleichbar der Situation ohne Alternativen einstellt. Im anderen Extremfall – bei Vorliegen einer zufälligen Kundenstruktur – können die Alternativen nicht bzw. erst später verworfen werden, was die Laufzeit verlängert.

Ein Vergleich von Abbildung 7.24 und 7.29 zeigt, daß die Laufzeit des kleineren Problems mit hohem Alternativenanteil noch unter derjenigen des größeren Problems mit niedrigem Alternativenanteil liegt. Obwohl die beiden Problemen zugrundeliegenden

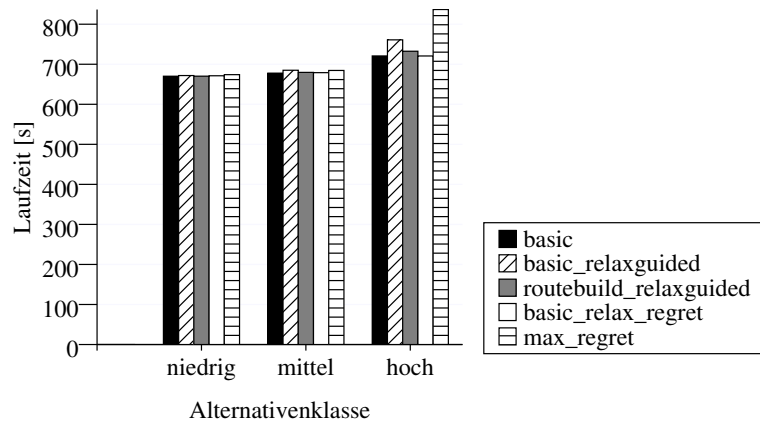


Abbildung 7.29: Vergleich der Instanziierungsstrategien (R1, 50 Kunden)

Graphen in etwa die gleiche Knotenzahl aufweisen, führt die durch die Alternativenbeziehung induzierte Netzwerkstruktur zu im Vergleich niedrigeren Laufzeiten.

Der Alternativenanteil hat einen erkennbaren Einfluß auf die Laufzeit, wobei in der Tendenz eine Erhöhung des Alternativenanteils erwartungsgemäß auch zu einer Erhöhung der Laufzeit führt. Das Maß der Laufzeitverlängerung ist dabei jedoch abhängig von der räumlich/zeitlichen Problemstruktur. Weisen die Standorte einen niedrigen Clusterungsgrad auf, verlängert sich die Laufzeit stärker gegenüber Probleminstanzen mit ausgeprägter Clusterstruktur.

Die vorliegenden Daten der Experimente weisen zudem darauf hin, daß bei gleicher räumlich/zeitlicher Problemstruktur und Knotenzahl das VRPA schneller als das VRPTW (mit gleicher Knotenzahl, d.h. größerer Kundenzahl) gelöst werden kann. Die zu einer größeren Knotenzahl führende Erhöhung des Alternativenanteils wirkt sich demnach schwächer aus als eine zahlenmäßig gleich große Erhöhung der Kundenzahl.

7.1.11 Schlußfolgerungen aus der algorithmischen Bewertung

In den vorangegangenen Abschnitten sind die entwickelten Verfahren aus mehreren Perspektiven verglichen worden, wobei als Bewertungskriterium die Laufzeit im Vordergrund stand. Betrachtet man den Beitrag der Heuristiken in der Initialphase, die Bounding-Varianten und die Instanziierungsstrategien, können einige Grundsätze und Leitlinien identifiziert werden.

Es ist nicht immer erforderlich, ein einzelnes Verfahrenselement bestmöglich auszuführen. Heuristiken und Bounding haben gezeigt, daß die einfacheren Strategien (Mehrfacheröffnung mit lokaler Suche bzw. einfaches Zuordnungsproblem) effizienter als die ausgefeilteren Varianten waren (Metaheuristik bzw. Additive Bounding).

Der zusätzliche Qualitätsvorteil letzterer wurde quasi „zu teuer“ erkaufte und wirkt sich damit im Endeffekt nicht leistungssteigernd aus.

Die zweite wesentliche Erkenntnis betrifft den Bereich der Problemstruktur. Letztlich hat diese sich als entscheidend für die Verfahrensleistung erwiesen und es hat sich – bspw. bei den Instanzierungsstrategien – gezeigt, daß einzelne Verfahrensvarianten mit bestimmten Problemstrukturen besonders gut umgehen können. Die strukturabhängige Konfiguration von Verfahren, evtl. auch eine dynamische Anpassung an den Suchfortschritt, erscheinen daher als vielversprechender Ansatz für die Zukunft.

Eine dritte Schlußfolgerung betrifft das Spaltenmanagement. Für kleinere Problemumfänge hat sich die Pufferung als erfolgreich erwiesen, weshalb entweder die Suche innerhalb des Puffers effizienter gestaltet werden muß, die Zahl der Suchvorgänge verringert oder der Pufferumfang verkleinert werden sollte.

Eine Verringerung der Laufzeit sollte das primäre Ziel für Fortsetzungsarbeiten sein. Die folgenden Ansatzpunkte sind dazu erkannt worden:

- Initialisierungsphase: Höhere Qualität der Lösungen für Probleme mit Alternativen, indem neue lokale Suchverfahren für die gezielte Auswahl von Alternativen entwickelt werden;
- Schnittstelle zwischen Master- und Pricing-Problem: Ausdehnung des Spaltenmanagements auf die Auswahl der aufzunehmenden Routen. Eine Möglichkeit dazu besteht darin, die aufzunehmenden Routen so auszuwählen, daß sie möglichst verschieden zu den bereits vorhandenen sind, um damit den Lösungsraum gut auszuloten und das Pricing-Problem mit qualitativ guten Dualvariablen (Strukturinformationen) zu versorgen;
- Pricing-Problem: Stärkere Bounding-Methoden, die ebenfalls explizit alternative Liefermöglichkeiten einbeziehen – bspw. über ein eigenes Branch-and-Bound innerhalb der globalen Constraint – und zu einem starken Pruning des Netzwerks führen;
- Branching: Verzweigen auf Gruppen von Alternativen, die mit hoher Wahrscheinlichkeit gemeinsam in der optimalen Lösung enthalten sind. Gerade für Probleme mit ausgeprägter Clusterstruktur ist zu vermuten, daß sich solche Gruppen bestimmen lassen und somit schnell obere Schranken ermittelt werden können.

Mit der Identifikation der Problemstruktur als treibende Größe, wurde bereits eine nicht ausschließlich algorithmische Feststellung über das VRPA getroffen. Im folgenden Unterkapitel soll der strukturelle Aspekt verstärkt betrachtet und in Beziehung zum logistischen Potential alternativer Liefermöglichkeiten gesetzt werden.

7.2 Untersuchung des logistischen Potentials alternativer Liefermöglichkeiten

Das in dieser Arbeit entwickelte Optimierungsverfahren bietet die Grundlage für ein Werkzeug, mit dem Logistikdienstleister eine neue Qualität von Lieferservice anbieten können. Neben den im vorangegangenen Unterkapitel untersuchten technischen Aspekten, ist es für die konzeptionelle Bewertung relevant der Frage nachzugehen, inwiefern das Angebot von alternativen Liefermöglichkeiten auch eine Kostenreduktion erlaubt und von welchen Einflußgrößen diese Reduktion gegebenenfalls abhängig ist. Für eine systematische Untersuchung wird ein szenariobasierter Ansatz verfolgt: Es wird eine Sammlung von Probleminstanzen erzeugt, die in ihrer räumlichen und zeitlichen Struktur erwarteten Anwendungsfällen entspricht. Von Interesse ist dabei weniger die Gesamtzahl der Kunden als deren Zusammensetzung aus unterschiedlichen Kundentypen mit und ohne Zustellalternativen, weil der Beitrag der alternativen Liefermöglichkeiten zur Kostenreduktion im Vordergrund der Untersuchungen steht. In den folgenden Abschnitten wird zunächst der Aufbau der Anwendungsszenarien erläutert, anschließend wird die Struktur der Testinstanzen beschrieben. Die Ergebnisse der Planungsläufe für diese Instanzen dienen dann zur Analyse des Konzepts der alternativen Liefermöglichkeiten.

7.2.1 Anwendungsszenario

Ein KEP-Dienstleister steht vor der Aufgabe, einer Gruppe von Kunden Sendungen zuzustellen. Die Kunden werden – der Idee der Kundentypisierung aus dem Marketing (s. bspw. (Nieschlag, Dichtl und Hörschgen 1994)) folgend – in Gruppen unterschiedlichen Kundentyps unterteilt und in unterschiedlicher Häufigkeit in einem geographischen Modellraum angeordnet.

Der Modellraum entspricht einer imaginären Stadt mit unterschiedlich großen Stadtteilen und stadtteilindividuellen Besiedlungsstrukturen und Bevölkerungsverteilungen. Die Kundentypen unterscheiden sich hinsichtlich der Tageszeiten, zu denen eine Sendungszustellung erfolgen kann und der möglichen Zustellorte.

In den folgenden Abschnitten werden die Kundentypen charakterisiert und der Aufbau der imaginären Stadt geschildert.

Kundentypen

Die verschiedenen Konsumententypen sollen unterschiedliche Verhaltensmuster hinsichtlich der urbanen Mobilität und häuslichen Erreichbarkeit widerspiegeln. Die einzelnen Kundentypen werden mit prägnanten Namen bezeichnet und weisen überwiegend zwei Liefermöglichkeiten auf. Die Liefermöglichkeiten unterscheiden sich hin-

sichtlich Ort und Zeitfenster, wobei letztere zunächst nur qualitativ beschrieben werden. Tabelle 7.3 zeigt die Kundentypen.

Nr.	Beschreibung	Erste Liefermöglichkeit	Zweite Liefermöglichkeit
1	Rentner	Zustellung zu Hause, lange Zeitfenster, ganztags	keine
2	Angestellter, Teilzeit	Zustellung zu Hause, mittlere Zeitfenster, vormittags oder nachmittags	keine
3	Hausfrau	Zustellung zu Hause, mittlere Zeitfenster, vormittags	Zustellung zu Hause, mittlere Zeitfenster, nachmittags
4	Angestellter, Vollzeit	Zustellung am Arbeitsplatz, kurze Zeitfenster, mittags	Abholung, abends
5	Selbständiger	Zustellung am Arbeitsplatz, ganztägig	Zustellung zu Hause, mittlere Zeitfenster, abends

Tabelle 7.3: Übersicht über Kundentypen

Bei dieser Charakterisierung von Kundentypen erscheint es wichtig darauf hinzuweisen, daß die plakativen Beschreibungen lediglich eine grobe Typisierung darstellen und stellvertretend für eine Zielgruppe mit dem zugehörigen Belieferungsmuster stehen.

In der experimentellen Untersuchung werden die Anteile der Kundengruppen an der Gesamtkundenanzahl innerhalb festgelegter Intervallgrenzen zufällig gewählt. Die Grenzen sind in Tabelle 7.4 wiedergegeben.

Nr.	Beschreibung	Mindestanteil	Maximalanteil
1	Rentner	20%	50%
2	Angestellter, Teilzeit	10%	50%
3	Hausfrau	10%	50%
4	Angestellter, Vollzeit	10%	40%
5	Selbständiger	10%	40%

Tabelle 7.4: Häufigkeiten einzelner Kundentypen

Zeitliche Kundenverteilung

Ein charakteristisches Unterscheidungsmerkmal der Kundentypen besteht in den bevorzugten Lieferzeiten, die durch die zeitliche Lage und Länge bestimmt werden. Für die Szenarienerzeugung werden die Zeitfensterlängen nach Tabelle 7.5 gewählt. Diese Vorgaben sind verglichen mit den zum Zeitpunkt der Erstellung dieser Arbeit am Markt

Bezeichnung	Dauer
kurz	$\frac{1}{2}$ Stunde
mittel	1 Stunde
lang	2 Stunden

Tabelle 7.5: Zeitfensterlängen

Zeitraum	Uhrzeit
morgens	8:00 – 10:00
vormittags	9:00 – 12:00
mittags	12:00 – 14:00
nachmittags	14:00 – 18:00
abends	18:00 – 21:00
ganztags	9:00 – 18:00

Tabelle 7.6: Unterschiedene Zeiträume

angebotenen Zeitfensterlängen vergleichsweise restriktiv. Infolge dieser Wahl wird einerseits der Lösungsraum verkleinert und damit die Rechenzeit verringert, andererseits wird auf diese Weise bereits eine mögliche zukünftige stärkere Serviceorientierung der Zustellangebote abgebildet.

Grundsätzlich wird in den Szenarien von Zustellungen im Zeitraum von 8:00 Uhr morgens bis 21:00 Uhr abends ausgegangen. Für eine feinere Differenzierung werden die Zeiträume nach Tabelle 7.6 begrifflich unterschieden. Das für einen Zustellort konkret gewählte Zeitfenster muß dabei voll in den entsprechenden Zeitraum fallen: ein langes Zeitfenster am Nachmittag muß demnach zwangsläufig spätestens um 16:00 Uhr beginnen. Um die Erzeugung der Szenariodaten zu vereinfachen, wird weiterhin die Vereinbarung getroffen, daß die Kundenzeitfenster nur zur vollen Stunde und 20 Minuten vor oder nach der vollen Stunde beginnen können.

In der Realität werden einige Kunden keine Zeitfenstervorgaben machen. Deren Berücksichtigung erfordert in der Optimierung eine längere Rechenzeit, da der Lösungsraum deutlich erweitert wird. Der Besuch dieser Kunden legt dem Dienstleister andererseits keine Restriktionen auf (abgesehen von den in dieser Anwendung kaum bindenden Kapazitätsrestriktionen), so daß diese Kunden nach erfolgter Tourenplanung noch an passender Stelle in den Tourenplan eingefügt werden können. Die erzeugten Szenarien gehen daher davon aus, daß alle Standorte Zeitfenster aufweisen. Die Bedeutung der Abholzeitfenster unterscheidet sich dabei von derjenigen der Zustellorte, weil die abendliche Abholung durch den Kunden für den Dienstleister lediglich bedeutet, daß die entsprechende Sendung bis zum Abend am Abholort hinterlegt worden sein muß.

Für Abholorte wird die Standzeit, d.h. die Zeit für die Beschickung, auf eine Minute pro Sendung festgelegt. Für Zustellorte werden verschiedene Ausprägungen der Stand-

zeit experimentell verglichen.

Gestaltung der Modellstadt

Die Gestalt der Modellstadt wird im Rahmen der Untersuchungen des Konzepts im wesentlichen durch das Stadtgebiet und die Besiedlungsstruktur festgelegt. Letztere bestimmt die räumliche Verteilung der Zustell- und Abholstandorte entsprechend der Kundentypen im Stadtgebiet. Im folgenden wird auf diese einzelnen Elementen näher eingegangen.

Stadtgebiet

Das Stadtgebiet der Modellstadt hat eine Ausdehnung von $20\text{km} \times 20\text{km}$ und entspricht damit in etwa einer mittleren deutschen Großstadt wie Karlsruhe. Das Gebiet ist aufgeteilt in fünf entsprechend Abbildung 7.30 angeordneten Stadtteilen.

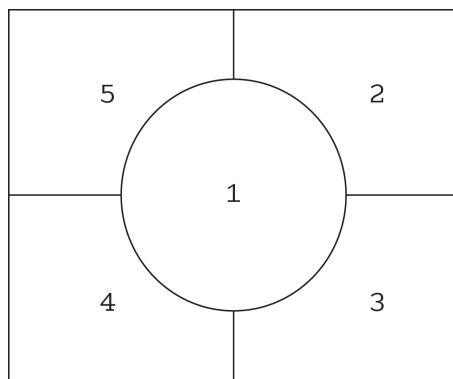


Abbildung 7.30: Anordnung der Stadtteile der Modellstadt

Die einzelnen Stadtteile haben bei annähernd gleicher Fläche einen unterschiedlichen Anteil an der Gesamtbevölkerung. Für die Szenarienerzeugung wird der Bevölkerungsanteil nach *Benfords Gesetz* bestimmt, so daß sich Werte entsprechend Tabelle 7.7 ergeben.

Benfords Gesetz (auch bekannt als *Gesetz der ersten Ziffer*) wurde bereits 1881 von Newcomb erkannt und 1938 von (Benford 1938) wiederentdeckt. Die empirische Beobachtung besagt, daß die erste signifikante Ziffer von vielen in der Natur gemessenen Größen nicht gleichverteilt ist, sondern sich nach

$$P(\text{erste signifikante Ziffer} = z) = \log_{10}\left(1 + \frac{1}{z}\right), z = 1, \dots, 9$$

Rangziffer	Bevölkerungsanteil
1	39%
2	23%
3	16%
4	12%
5	10%

Tabelle 7.7: Verteilung der Bevölkerungszahl der Stadtteile entsprechend Benfords Gesetz

Rangziffer	Modellannahme	Realdaten			
		Karlsruhe	Düsseldorf	Hamburg	Köln
1	39%	36,1%	45,5%	48,7%	43,7%
2	23%	23,5%	27,8%	26,9%	22,6%
3	16%	17,9%	14,8%	15,6%	16,2%
4	12%	12,3%	8,9%	7,6%	11,6%
5	10%	9,3%	3,1%	1%	5,8%

Tabelle 7.8: Übersicht über die Bevölkerungsverteilung einiger deutscher Städte im Vergleich mit Benfords Gesetz.

ergibt. Voraussetzung ist, daß eine ausreichend große Grundgesamtheit vorliegt und die Messung skaleninvariant, d.h. unabhängig vom verwendeten Maßstab, ist. An dieser Stelle soll nicht weiter auf die Theorie eingegangen werden, sondern eine kurze empirische Überprüfung der Verteilungsannahme anhand einiger deutscher Städte gegeben werden. Für diese Überprüfung wurden die Einwohnerzahlen aus der offiziellen Stadtstatistik von Karlsruhe, Düsseldorf, Hamburg und Köln entnommen, absteigend der Größe nach sortiert und in fünf Gruppen eingeteilt. Die Ergebnisse in Tabelle 7.8 lassen die Verteilungsannahme

$$P(\text{Bevölkerungsanteil des Stadtteils auf Rang } z) = \log_6\left(1 + \frac{1}{z}\right), z = 1, \dots, 5, \quad (7.1)$$

plausibel erscheinen.

Im Rahmen der Szenarioerzeugung werden die relativen Größen der Stadtteile verwendet, um die Anzahl der Standorte pro Stadtteil festzulegen. Allerdings ist bei dieser Erzeugung ein Kompromiss zwischen theoretisch angestrebter und praktisch erreichbarer Anteile zu schließen. Die Laufzeit des Optimierungsverfahrens erlaubt es nicht, Szenarien mit 100 Standorten, wie sie zur exakten Nachbildung der Modellannahmen erforderlich wären, zu untersuchen. Aus diesem Grund – und um die Zeit zur Erzeugung der Szenarien selbst in akzeptablen Grenzen zu halten – wird pro Größenklasse

eine Abweichung von bis zu 40% von den Modellvorgaben erlaubt. Angesichts der in der in Realität zu beobachtenden Abweichungen (s. Tabelle 7.8) erscheint dies vertretbar.

Besiedlungsstruktur

Bei gegebener Anordnung der Stadtteile und bekannten Bevölkerungsanteilen, muß noch die räumliche Zuordnung getroffen werden, da nicht unbedingt der zentrale Stadtteil auch den größten Bevölkerungsanteil aufweist. Weiterhin weisen die einzelnen Stadtteile in unterschiedlichem Maße Wohnungen und Arbeitsplätze auf. Drei Typen von Siedlungsstruktur sollen unterschieden werden:

- überwiegend Wohngegend,
- überwiegend Gewerbe,
- gemischte Besiedlung.

Per Definition ist der Bevölkerungsanteil in einer Wohngegend sehr viel größer als in einem Gewerbegebiet und umgekehrt die Zahl der Arbeitsplätze in einem Gewerbegebiet größer als in einer reinen Wohngegend. Dieser Sachverhalt wird für die Modellstadt durch eine Spezifikation der möglichen Besiedlungsstruktur entsprechend Tabelle 7.2.1 nachgebildet.

Rangziffer Bevölkerung	Stadtteiltyp		
	Wohngegend	gemischt	Gewerbegebiet
1	✓		
2	✓	✓	
3	✓	✓	
4		✓	✓
5			✓

Tabelle 7.9: Mögliche Beziehungen zwischen Bevölkerungsanteil und Stadtteiltyp

Um eine realistische Verteilung der Kunden über das Stadtgebiet nachzubilden, die nicht nur quantitative, sondern auch qualitative Aspekte umfaßt, ist eine Zuordnung von Kundentypen zu Stadtteilen erforderlich. Zwei Beispiele zeigen die Notwendigkeit dafür:

- Rentner- und Hausfrauen leben typischerweise in Wohngebieten oder gemischten Stadtteilen,
- Arbeitsplätze finden sich hauptsächlich in Gewerbegebieten aber auch in gemischten Stadtteilen.

In Tabelle 7.10 markiert „✓“ die in der Szenarienerzeugung erlaubten Zuordnungen der Standorte von Kunden entsprechenden Typs zu Stadtteiltypen.

Kudentyp	Stadtteiltyp		
	Wohngegend	gemischt	Gewerbegebiet
Rentner	✓	✓	–
Angestellter, Teilzeit	✓	✓	✓
Hausfrau	✓	✓	–
Angestellter, Vollzeit	✓	✓	✓
Selbständiger	✓	✓	✓

Tabelle 7.10: Mögliche Zuordnungen von Kundentypen zu Stadtteiltypen

Das Depot des KEP-Dienstleisters, von dem aus die Belieferung stattfinden soll, wird ebenfalls in einem Gewerbegebiet angesiedelt. Es werden zwei Abholorte in der Modellstadt platziert. Weil Abholorte entweder an stark frequentierten oder bequem erreichbaren Orten errichtet werden (s. Kapitel 2) und es sich dabei im Modell entweder um ausgeprägte Wohngegenden oder Gewerbegebiete handelt, befinden sich die Abholorte jeweils im Zentrum des größten bzw. kleinsten Stadtteils (Rangziffer 1 bzw. 5). Den Kunden mit abendlicher Abholung wird derjenige Abholort zugeordnet, der dem ersten Kundenstandort am nächsten liegt.

Für die experimentelle Untersuchung werden die Entfernungen der Kundenstandorte untereinander per Luftlinie bestimmt. Dazu müssen zunächst die Koordinaten der Kundenstandorte erzeugt werden. Eine Genauigkeit von 100m wird in diesem Modell als ausreichend angesehen, das Stadtgebiet läßt sich daher als quadratische Fläche mit einer Seitenlänge von 200 Längeneinheiten (LE) abbilden.

Für die Koordinatenerzeugung müssen die Gebiete der Stadtteile berücksichtigt werden. Der Stadtteil 1 hat einen Radius von 50LE mit Mittelpunkt (100, 100). Definiert man die linke obere Ecke des Stadtgebiets als Ursprung des Koordinatensystems, liegt ein Kundenstandort i mit Koordinaten $(x_i, y_i) \in [0 \dots 199] \times [0 \dots 199]$ im Stadtteil 1, wenn

$$(x_i - 100)^2 + (y_i - 100)^2 \leq 50^2$$

gilt. Auf ähnliche Weise lassen sich die Koordinatenbereiche der anderen Stadtteile beschreiben. Tabelle 7.11 gibt eine vollständige Übersicht.

7.2.2 Erzeugung der Szenarien

Testinstanzen werden im Operations Research oftmals zufällig erzeugt, d.h. die Parameter der Instanzen werden mit zufällig ausgewählten Werten belegt. Diese Vorge-

Stadtteil	Koordinatenbereich
1	$(x_i - 100)^2 + (y_i - 100)^2 \leq 50^2$
2	$(x_i - 100)^2 + (y_i - 100)^2 > 50^2$ und $x_i \geq 100$ und $y_i < 100$
3	$(x_i - 100)^2 + (y_i - 100)^2 > 50^2$ und $x_i \geq 100$ und $y_i \geq 100$
4	$(x_i - 100)^2 + (y_i - 100)^2 > 50^2$ und $x_i < 100$ und $y_i \geq 100$
5	$(x_i - 100)^2 + (y_i - 100)^2 > 50^2$ und $x_i < 100$ und $y_i < 100$

Tabelle 7.11: Koordinatenbereiche der Stadtteile

hensweise bietet sich immer dann an, wenn es keine oder nur geringe Abhängigkeiten zwischen verschiedenen Parametern gibt. Gibt es jedoch wechselseitige Abhängigkeiten, stößt diese Vorgehensweise an Grenzen, weil zufällige Wertzuweisungen die Abhängigkeiten nicht berücksichtigen und viele Zuweisungsversuche notwendig sind, um gültige Szenarien zu erhalten. Für den vorliegenden Fall finden sich derartige Abhängigkeiten bspw. zwischen der Anzahl von Kunden eines bestimmten Typs, der Anzahl von Standorten pro Stadtteil und der Besiedlungsstruktur.

Die Beschreibung der Szenarien kann als Spezifikation von Anforderungen aufgefasst werden, der ein gültiges Szenario genügen muß. Die deklarative Form legt es nahe, die Anforderungen als Constraint Programm zu formulieren. Jede zulässige Lösung des Constraint Programms entspricht dann einem gültigen Szenario. Ein ähnlicher Ansatz wurde von (Bin, Emek, Shurek und Ziv 2002) für die Erzeugung von Testprogrammen für komplexe Informationssysteme umgesetzt.

Die Szenarienerzeugung unterscheidet sich von üblichen CP-Problemen, wie sie auch in dieser Arbeit beschrieben wurden, dadurch, daß die Ergebnisse ein bestimmtes Maß von „Zufälligkeit“ aufweisen müssen. Die Lösungssuche in Constraint Programmen geschieht üblicherweise systematisch, so daß aufeinanderfolgende Lösungen sehr ähnlich sind. Beispielsweise gilt für die ersten Lösungen typischerweise, daß sich die Werte der Variablen sämtlich am unteren Rand der Wertebereiche befinden. Eine systematische Erzeugung von Lösungen führt daher nicht zu einer repräsentativen Stichprobe des gesamten Lösungsraums. Um dies dennoch zu erreichen, sind Suchverfahren implementiert worden, die eine zufallsgesteuerte Instanziierung durchführen und im Ergebnis strukturell gleiche aber unähnliche Szenarien liefern.

Die Suche nach gültigen Lösungen des CP-Szenarienmodells hat sich im Rahmen der Entwicklung des Szenariengenerators als vergleichsweise zeitaufwendig erwiesen. Eine Möglichkeit, das Verfahren zu beschleunigen liegt darin, die konkrete Bestimmung der Zeitfenster von der restlichen Erzeugung zu trennen. Dies ist möglich, weil die Zeitfenster – sobald der Kundentyp und der Zeitraum festgelegt worden sind – in den gegebenen Grenzen unabhängig gewählt werden können.

Die Beschreibung von Testszenarien mit Constraint Programming sichert nur die Einhaltung einer vorgegebenen Problemstruktur; die Lösbarkeit der erzeugten Szenarien ist nicht gewährleistet, ebensowenig wie garantiert ist, daß die Optimallösung inner-

halb einer vorgegebenen Zeit gefunden werden kann. Aus diesem Grund müssen eine Vielzahl von Szenarien erzeugt und berechnet werden, um für die Ableitung von Erkenntnissen über eine ausreichende Anzahl von optimal gelösten Fällen zu verfügen. Der folgende Abschnitt gibt eine Übersicht über die erzeugten und berechneten Szenarien.

7.2.3 Szenarienübersicht

Insgesamt wurden vier Szenarienreihen erstellt, die sich im Hinblick auf die Kundenzahl und die Standzeit unterscheiden. Jede Szenarienreihe umfaßt 200 Instanzen, wobei die ungeraden Reihen alternative Standorte aufweisen und die geraden Szenarienreihen nicht. Lage und Zeitfenster der Standorte wurden zwischen den Szenarienreihen mit unterschiedlicher Standzeit nicht verändert. Für die Standzeiten sind Werte von fünf Minuten für Zustellungen an Kunden und von einer Minute für Zustellungen an Abholstandorten angenommen worden.

Die Instanzen der Szenarienreihen ohne Alternativen wurden durch zufällige Standortauswahl aus den korrespondierenden Instanzen mit Alternativen abgeleitet. Konkret wurden alle Kundenstandorte ohne alternative Liefermöglichkeit übernommen, während für alle anderen Kunden eine (gleichverteilte) zufällige Auswahl aus den zwei verfügbaren Alternativen getroffen wurde. Damit handelt es sich bei den Instanzen ohne Alternativen um echte Teilmengen der gleichen Instanz mit Alternativen. Zwei infolge dieses Erzeugungsschemas zusammengehörige Instanzen werden im weiteren Verlauf als Instanzenpaar bezeichnet.

Tabelle 7.12 gibt einen Überblick über die erstellten Szenarien.

Szenarienreihe	Kundenzahl	Alt.	ØAnzahl Standorte	ØAnzahl Kundentypen				
				Rentner	Teilzeit	Hausfrau	Vollzeit	Selbstständig
1	36	✓	55,9	9,9	6,1	4,8	10,3	4,9
2		–	36					
3	60	✓	93,6	16,7	9,7	8,6	16,5	8,4
4		–	60					

Tabelle 7.12: Übersicht über die erzeugten Szenarien

Nicht alle Instanzen sind innerhalb der auf 1000 Sekunden begrenzten Rechenzeit optimal lösbar. Bei der Betrachtung von Instanzenpaaren tritt zudem die Situation auf, daß nur eine Instanz optimal gelöst wurde, während für die zweite nur suboptimale

oder keine Lösungen gefunden wurden. Da eine Auswertung – bspw. die erreichbare Verbesserung – jedoch nur dann sinnvoll erfolgen kann, wenn ein Instanzenpaar vollständig optimal gelöst wurde, werden in die Auswertung nur diejenigen Paare aufgenommen, für die das zutrifft. Tabelle 7.13 zeigt die Strukturdaten der diesem Kriterium entsprechenden Szenarien.

Szenarienreihe	Kundenzahl	Alt.	ØAnzahl Standorte	ØAnzahl Kundentypen					optimallöste Szenarien
				Rentner	Teilzeit	Hausfrau	Vollzeit	Selbstständig	
1	36	✓	55,8	10,1	6,1	4,5	10,6	4,7	98
2		–	36						
3	60	✓	93,1	17,3	9,6	8,5	16,0	8,6	64
4		–	60						

Tabelle 7.13: Übersicht über die Szenarien für die Auswertung

Die Untersuchungen fanden mit der Verfahrensvariante `routebuild_relax_regret_feasible_nobuffer` statt, weil diese bei der Lösung der Solomon-Instanzen für Probleme der RC-Klasse gute Ergebnisse zeigte und eine ähnliche Struktur auch in den Szenarientests vorliegt. Abbildungen 7.31 und 7.32 zeigen ein Beispiel ohne bzw. mit Alternativen mit eingezeichnetem Tourverlauf. Es wird deutlich, daß im Fall von alternativen Liefermöglichkeiten es in geringerem Maß zu räumlich-/zeitlichen Überschneidungen kommt.

In den folgenden Abschnitten werden die Analyseergebnisse vorgestellt.

7.2.4 Wegersparnis durch alternative Liefermöglichkeiten

Neben dem eher „weichen“ Zielkriterium der Servicequalitätverbesserung ist das „harte“ Kriterium der Kostenreduktion für den Logistikdienstleister der entscheidende Qualitätsmaßstab. In den durchgeführten Experimenten bestand das Optimierungsziel in der Minimierung des zurückgelegten Weges. Für die Bewertung wird die relative Verringerung der Wegelänge bezogen auf die jeweiligen Instanzen ohne Alternativen bestimmt. Bezeichnen z_i^* die optimale Wegelänge der Instanz i ohne Alternativen und $z_{A,i}^*$ die Länge der gleichen Instanz mit alternativen Liefermöglichkeiten, wird die Ver-

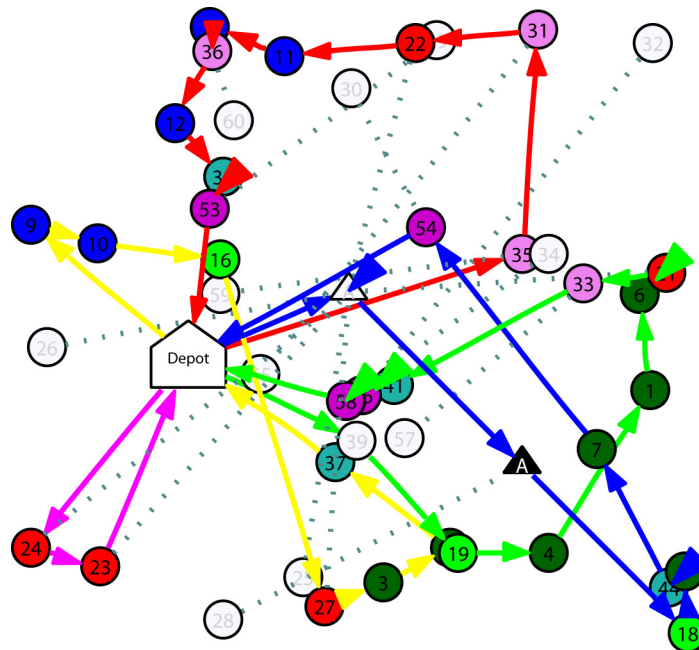


Abbildung 7.31: Beispiel eines Zustellszenarios ohne Alternativen

besserung v_i durch

$$v_i = \frac{z_i^* - z_{A,i}^*}{z_i^*} = 1 - \frac{z_{A,i}^*}{z_i^*} \quad (7.2)$$

bestimmt. Tabelle 7.14 gibt die Lageparameter der relativen Verbesserung an.

Lageparameter	Ausprägung
Mittelwert	20%
Standardabweichung	0,07
Minimum	1%
Maximum	35%
Median	20%

Tabelle 7.14: Lageparameter der relativen Verbesserung über alle Szenarien

Abbildung 7.33 zeigt das Histogramm der relativen Verbesserungen. Die intuitive Vermutung einer vorliegenden Normalverteilung konnte statistisch nicht bestätigt werden. Ein Vergleich der erreichbaren Wegelängenreduktion für 36 bzw. 60 Kunden ergibt ein ähnliches Bild unabhängig von Kundenzahl und Standzeit. Die entsprechenden Daten sind in Tabelle 7.15 angegeben

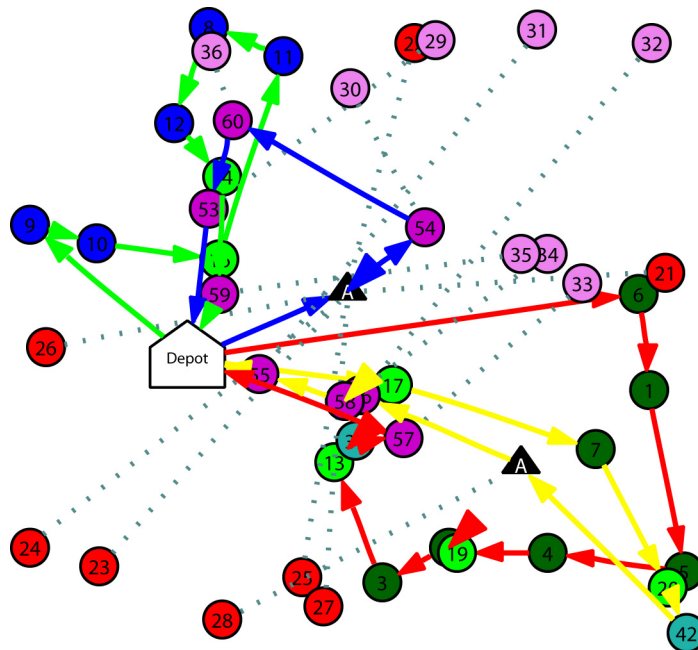


Abbildung 7.32: Beispiel eines Zustellszenarios mit Alternativen

Parameter	36 Kunden	60 Kunden
Mittelwert	19,8%	19,9%
Standardabweichung	0,07	0,07
Minimum	1,1%	2,3 %
Maximum	31,8%	35,5%
Median	20,4%	19,3 %

Tabelle 7.15: Lageparameter der relativen Verbesserung einzelner Szenarienreihen

7.2.5 Reduktion der Fahrzeugzahl

Neben der Verringerung der Weglänge liegt die Fahrzeuganzahl im Blickfeld des Dienstleisters. Zwar wurde in dieser Arbeit die Reduktion der Fahrzeugzahl nicht als Zielfunktion betrachtet³, es ist jedoch dennoch interessant, die Ergebnisse auf eine mögliche Wirkung diesbezüglich zu untersuchen.

Abbildungen 7.34 und 7.35 zeigen auch bei einer reinen Weglängenoptimierung eine Veränderung hinsichtlich der Flottengröße.

³Diese Zielfunktion läßt sich relativ einfach nachbilden, indem im kompakten Modell $c_{0i} = 1$ und $c_{ij} = 0$ für alle Knoten i bzw. j gesetzt wird. Ein effizientes Verfahren könnte jedoch die Tatsache nutzen, daß eine explizite Routenbestimmung nicht unbedingt erforderlich ist, solange die kapazitive und zeitliche Zulässigkeit gewährleistet sind.

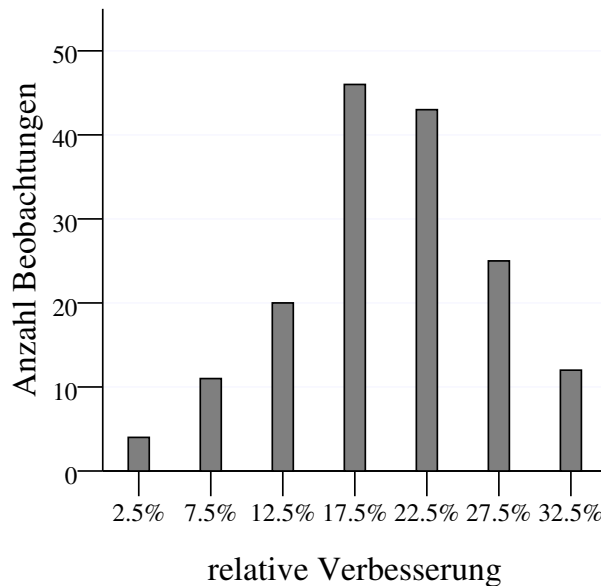


Abbildung 7.33: Histogramm der relativen Verbesserung

Die genauere Analyse der Daten zeigt, daß sich die durchschnittliche Fahrzeugzahl von 6,0 bei Instanzen ohne Alternativen zu 5,7 bei Instanzen mit Alternativen schwach verringert. Ein approximativer Zweistichproben-Gaußtest liefert zum Signifikanzniveau $\alpha = 5\%$ die Aussage, daß die mittlere Fahrzeugzahl für Instanzen ohne Alternativen größer als diejenige bei Instanzen mit Alternativen ist. Dies bedeutet jedoch nicht, daß die Fahrzeugzahl grundsätzlich kleiner ist; vielmehr lassen sich Instanzen finden, bei denen trotz Alternativen mehr Fahrzeuge erforderlich sind. Der Grund hierfür ist in der Zielfunktion zu suchen, die keinen Anreiz zur Senkung der Fahrzeugzahlen gibt.

7.2.6 Strukturelle Ursachen der Verbesserung

Neben einer Quantifizierung der Wirkung von alternativen Liefermöglichkeiten ist für die Konzeption und Planung derartiger Angebote die Ursachenermittlung wichtig. Für den Dienstleister stellt sich die Frage, ob es bestimmte Merkmale der Kundenstruktur gibt, die in direkter Beziehung zum Verbesserungspotential stehen. Eine Analyse der Kundenstruktur könnte dann darüber Aufschluß geben, ob eine Ausweitung des Serviceangebots in Form alternativer Lieferorte und -zeiten ein Verbesserungspotential ergibt, das die möglicherweise erhöhten Organisationskosten kompensiert.

Ausgehend von dieser Überlegung soll zunächst der Zusammenhang zwischen dem Alternativenanteil r_A und der erreichbaren Verbesserung untersucht werden. Für jede Szenarieninstanz mit Alternativen wird der Anteil der alternativen Liefermöglichkeiten an der Gesamtstandortzahl bestimmt. Die statistischen Lageparameter des Alterna-

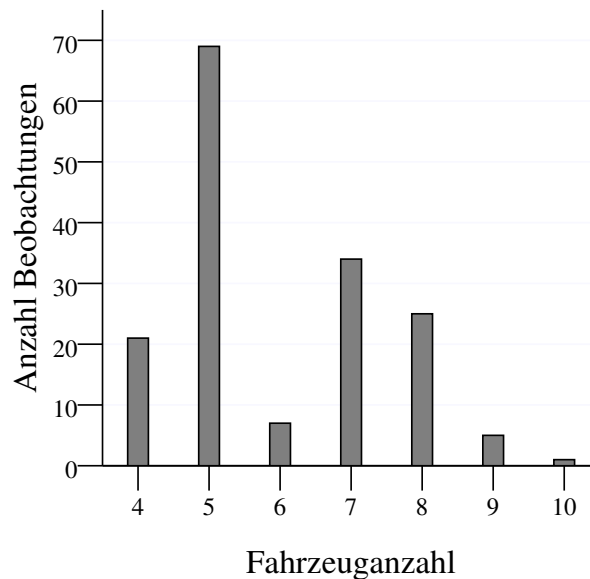


Abbildung 7.34: Eingesetzte Fahrzeuganzahl in Szenarien ohne Alternativen

tivenanteils sind in Tabelle 7.16 wiedergegeben.

Lageparameter	Ausprägung
Mittelwert	55%
Standardabweichung	0,08
Minimum	39%
Maximum	72%
Median	58%

Tabelle 7.16: Lageparameter des Alternativenanteils

Der relativen Größe Alternativenanteil wird die relative Verbesserung gegenübergestellt. Als Maß für den linearen Zusammenhang wird der Korrelationskoeffizient ρ_{v,r_a} berechnet. Für diesen ergibt sich

$$\rho_{v,r_a} = 0,39.$$

Der Grad des Zusammenhangs zwischen Verbesserung und Alternativenanteil kann daher nur als gering eingeschätzt werden.

Im nächsten Schritt wird der Effekt der Abholorte näher untersucht, indem die Korrelation $\rho_{v,v}$ zwischen der relativen Verbesserung und dem Anteil der Kunden vom Typ „Vollzeit“ an der Gesamtkundenanzahl berechnet wird. Über alle Szenarien ergibt sich

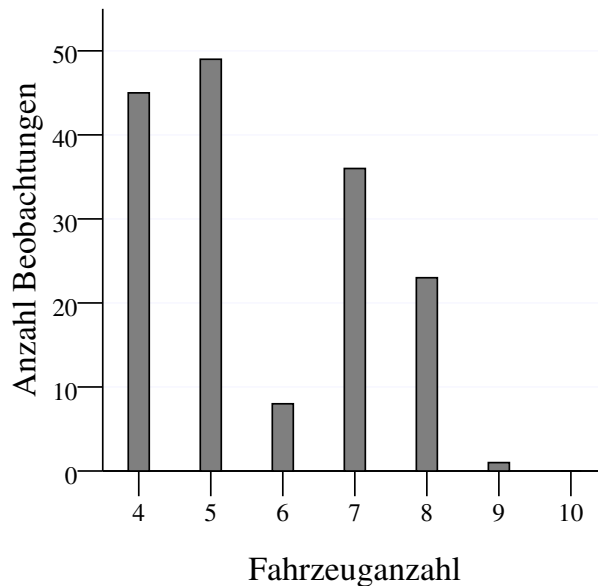


Abbildung 7.35: Eingesetzte Fahrzeuganzahl in Szenarien mit Alternativen

ein Wert von

$$\rho_{v,v} = 0,52.$$

Für die 36 bzw. 60-Kunden-Instanzen getrennt bestimmt ergeben sich Werte von 0,51 bzw. 0,57.

Die positive Korrelation bestätigt erwartungsgemäß den Gewinn an Einsparungspotential bei erhöhtem Abholungsanteil und belegt einen zumindest schwachen linearen Zusammenhang.

Dennoch ist auch diese Korrelation insgesamt als schwach anzusehen, obgleich sie noch deutlich größer ist als die (hier nicht angegebene) Korrelation zwischen den restlichen Kundentypen und der relativen Verbesserung. Dieses Ergebnis ist insofern nicht überraschend, als daß diese Kennzahlen die Kombinationen von Kunden bzw. deren Standorten nicht abdecken. Im folgenden Abschnitt wird daher eine Kennzahl entwickelt, die diese Kombinierbarkeit quantifiziert.

7.2.7 Kombinierbarkeit von Liefermöglichkeiten

Die vorangegangenen Analysen haben gezeigt, daß es erforderlich ist, die Kombinierbarkeit von Standorten zu Touren zu messen, um Planungsszenarien miteinander zu vergleichen.

Um das Verbesserungspotential durch Alternativen mittels einer Kennzahl auszudrücken, wird in dieser Arbeit vorgeschlagen, die Kombinierbarkeit von Standorten

zu Touren in Szenarien mit Alternativen derjenigen des korrespondierenden Szenarios ohne solche gegenüberzustellen. Zur Bestimmung der Kombinierbarkeit werden Ketten von drei Standorten betrachtet. Bezeichnen

$$c_{ijk} = c_{ij} + c_{jk} \quad (7.3)$$

die Kosten einer Kette im Szenario ohne Alternativen und

$$c_{ijk}^A = \min_{p \in \mathcal{A}_i, q \in \mathcal{A}_j, r \in \mathcal{A}_k} (c_{pq} + c_{qr}) \quad (7.4)$$

die minimalen Kosten einer Kette zwischen Alternativknoten von i, j und k , dann wird

$$\kappa_{ijk} = \frac{c_{ijk} - c_{ijk}^A}{c_{ijk}} = 1 - \frac{c_{ijk}^A}{c_{ijk}} \quad (7.5)$$

als die Verbesserung der Kombination der Kundenstandorte i, j und k infolge Alternativennutzung definiert, wobei jeweils nur zulässige Ketten betrachtet werden. Die Durchschnittsbildung über alle Tupel (i, j, k) liefert dann den Verbesserungsindex κ^+ :

$$\kappa^+ = \frac{1}{|\{(i, j, k)\}|} \sum_{(i, j, k)} \kappa_{ijk}. \quad (7.6)$$

Für die Berechnung durchlaufen i und k die Menge der Kunden zzgl. dem Depot (als Ausgangs- bzw. Enddepot), während j nur die Menge der Kunden durchläuft.

Das Maß des Zusammenhangs zwischen Verbesserungsindex und relativer Verbesserung wird erneut über die Korrelation ausgedrückt. Für den Korrelationskoeffizienten ρ_{v, κ^+} gilt dann

$$\rho_{v, \kappa^+} = 0,60.$$

Die Korrelation ist damit besser als für die im vorangegangenen Abschnitt untersuchten Kenngrößen. In Abbildung 7.36 wird die Regressionsgerade des linearen Modells abgebildet.

Die Abbildung läßt erkennen, daß der Verbesserungsindex zumindest eine Trendaussage erlaubt. In der Tat liegt der Wert des Index darin, daß es mit seiner Hilfe möglich ist, verschiedene gegebene Planungssituationen relativ schnell zu vergleichen, ohne tatsächlich eine Planung durchzuführen.

Die Güte des Verbesserungsindex ließe sich noch steigern, wenn längere Teilketten als in dieser Arbeit betrachtet würden. Allerdings zieht dies auch einen höheren Berechnungsaufwand nach sich, der sich dann bereits dem Aufwand einfacher aber oftmals dennoch guter Heuristiken nähert. Insofern muß hier ein Kompromiß geschlossen werden bzw. über andere Berechnungsalgorithmen nachgedacht werden.

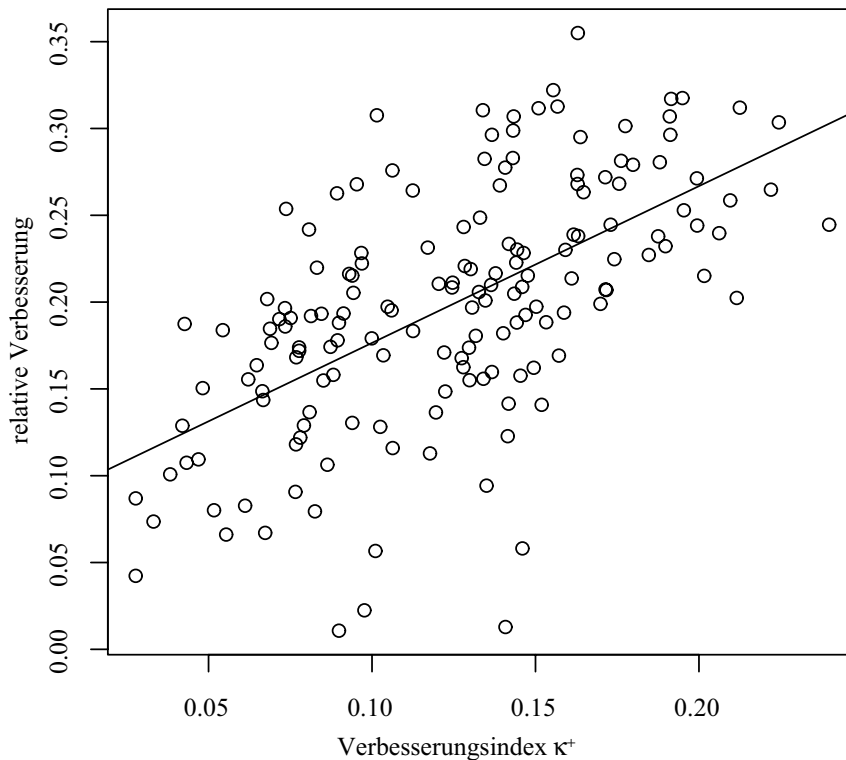


Abbildung 7.36: Zusammenhang zwischen Verbesserungsindex κ^+ und relativer Verbesserung

7.2.8 Bilanz und Ausblick der logistischen Bewertung

Die Bewertung des Konzepts alternativer Lieferorte und -zeitfenster hat ergeben, daß sich unter plausiblen Annahmen für Kundenstandorte und -zustelloptionen ein großes Verbesserungspotential – mit im Mittel knapp 20% der Wegelänge – ergibt. Quasi als Nebeneffekt stellt sich in der überwiegenden Zahl der Fälle eine Reduktion der Fahrzeugzahl ein. Ein einfache Abschätzung des Potentials erscheint nicht möglich, stattdessen müssen Maße wie der Verbesserungsindex herangezogen werden, die explizit den Aspekt der Kombinierbarkeit adressieren.

Die experimentelle Untersuchung hat jedoch auch einen Zusammenhang zwischen Lösungsmethode und Untersuchung aufgezeigt: Die Laufzeiten exakter Verfahren erlauben es heute noch nicht, Probleme von praktischer Größenordnung sicher innerhalb kurzer Zeit zu lösen. Zwar wurden in dieser Arbeit Probleme mit bis zu 100 Standorten optimal gelöst, dies gelingt jedoch nur bei „gutartiger“ Struktur.

Aus diesem Grund können Testumfänge und andere damit zusammenhängende Parameter wie Standortdichte, durchschnittliche Fahrzeiten, Standzeitvariationen usw. auch nicht in dem Maße verändert werden, wie dies erforderlich wäre, um den Anwendern aus der betrieblichen Praxis auf direkte Weise nachvollziehbare Planungsze-

narien zu präsentieren. Für exakte Verfahren führt der Weg heute noch über aggregierte und abstrahierte Szenarien.

Aus dieser Erkenntnis läßt sich Forschungsbedarf nach heuristischen Verfahren ableiten. Diese Verfahren könnten mehrere Zielsetzungen verfolgen:

- Höhere Lösungsgeschwindigkeit,
- Lösung größerer Probleme,
- andere Zielfunktionen, bspw. Reduktion der Flottengröße, Minimierung der Gesamtdauer oder Minimierung der spätesten Rückkehr zum Depot,
- Behandlung von mehr als zwei Alternativen pro Kunde,
- Berücksichtigung von weiteren Nebenbedingungen wie bspw. beschränkten Wartezeiten, Mindestauslastungsgraden.

Auch die Bewertung selbst kann sinnvoll erweitert werden. Die Stochastik von Fahrzeiten der Zustellfahrzeuge, Anwesenheitszeiten der Kunden und Standzeiten beim Kunden sollte für eine Evaluierung des gesamten Zustellkonzepts in geeigneten Modellen und Lösungsansätzen berücksichtigt werden. Methoden hierzu können die Szenariotechnik oder zeitdiskrete stochastische Modelle sein, die bereits in angrenzenden Problembereichen verwendet wurden (s. (Furmans 2003)).

8 Zusammenfassung

Die Zustellung von Warensendungen an private Endkunden – die *Business to Consumer*-Logistik (B2C-Logistik) – stellt Unternehmen vor besondere logistische Herausforderungen. Ende der 1990er Jahre machten zahlreiche Unternehmen die Erfahrung, daß die Organisation, physische Handhabung und Auslieferung einer Vielzahl von kleinen Sendungen gegenüber *Business to Business*-Prozessen eine andere Gestaltung der Logistik erfordern. Obwohl B2C-Logistik bereits seit langem aus dem Versandhandel bekannt ist, führte gerade die Unterschätzung der logistischen Probleme in Zeiten des eCommerce-Booms Mitte bis Ende der 1990er Jahre zum wirtschaftlichen Aus zahlreicher Unternehmen.

Neben den Kommissioniersystemen innerhalb von Distributionszentren erwies sich für die Transportlogistik die *Letzte Meile*, d.h. der letzte Abschnitt der Transportkette zwischen Ende des Hauptlaufs und Erreichen des Empfängers, als besondere Herausforderung. Ursächlich dafür sind die Vielzahl von Stops von vergleichsweise langer Dauer, bei denen zumeist nur eine Sendung ausgeliefert wird – sofern der Empfänger überhaupt angetroffen wird.

Mehrfache Zustellversuche sind unwirtschaftlich und pragmatische Lösungen, wie die Abgabe von Sendungen in der Nachbarschaft, rechtlich bedenklich und aus Servicegesichtspunkten nicht zufriedenstellend. Es wurden daher eine Reihe von technischen Lösungen entwickelt, um die Sendungszustellung von der Anwesenheit des Empfängers zu entkoppeln. In dieser Arbeit wird diesen Ansätzen ein alternatives Konzept zur Seite gestellt, das auf der Kooperation von Versender, Dienstleister und Endkunde aufbaut. Während die Entkopplungskonzepte ihren Ursprung primär in einer Kostenführer-Strategie haben, können die neu vorgestellten Abstimmungskonzepte als Instrumente zur Leistungsdifferenzierung im Wettbewerb gesehen werden.

Der Grundgedanke der Abstimmungskonzepte besteht darin, den Übergabepunkt der Sendungen zu einer verhandelbaren und damit wettbewerbliehen Größe zu machen. Technisch realisierbar ist ein derartiges Konzept über eine informationstechnische Plattform wie den *Personal Logistics Assistant* (PLA), der im Rahmen dieser Arbeit entworfen und implementiert wurde.

Den Übergabepunkt von Sendungen können Kunden nur dann wählen, wenn sie über alternative Lieferorte und -zeiten verfügen, die sie den Dienstleistern direkt oder indirekt mitteilen. Im täglichen operativen Betrieb müssen Dienstleister aus den ihnen gebotenen Alternativen eine Auswahl nach wirtschaftlichen Gesichtspunkten treffen. Für sie stellt sich ein Tourenplanungsproblem, das hinsichtlich seiner Komplexität über

die bisherigen Anforderungen hinausgeht. In der Tat erfordert ein System in der Art des PLA neue Planungsverfahren, um zu einem einsatzfähigen Werkzeug kooperativer Distributionsplanung zu werden.

Das Tourenplanungsproblem mit alternativen Lieferorten und -zeiten wird in dieser Arbeit als ein Kernproblem der kooperativen Distributionsplanung erkannt. Zu seiner Lösung wird ein exaktes Optimierungsverfahren konzipiert und implementiert. Die Verfahrensentwicklung basiert auf einer Modellierung des – in Anlehnung an die übliche Notation so genannten – *Vehicle Routing Problem with Alternatives* (VRPA) als gemischt-ganzzahliges lineares Programm (MIP). Es wird aufgezeigt, daß derartige kompakte MIP-Modelle einer direkten Lösung infolge der Größe des Lösungsraums nicht zugänglich sind und daher eine alternative Methode gefunden werden muß.

Die Wahl der Lösungsmethode stützt sich auf einen umfassenden Literaturüberblick. Aus diesem ergibt sich, daß Dekompositionsmodelle nach derzeitigem Wissensstand die Methode der Wahl zur Lösung großer und strukturierter gemischt-ganzzahliger linearer Programme sind. Weiterhin wird deutlich, daß sich das aus der Informatik stammende *Constraint Programming* zur deklarativen Modellierung und Lösung von kombinatorischen Optimierungsproblemen mit komplexen Nebenbedingungen eignet. Es wird dargestellt, wie sich Lineare Programmierung und Constraint Programming kombinieren lassen. In der Synthese der Erkenntnisse wird schließlich die Methode des Constraint Programming-basierten Column Generation, eingebettet in einen Branch-and-Bound-Ansatz zur Herstellung ganzzahliger Lösungen, ausgewählt. Die Lösungsmethode kann damit in die Klasse der *Branch-and-Price*-Verfahren eingeordnet werden.

Die methodischen Grundlagen von Constraint Programming, Branch-and-Price, sowie von heuristischen Verfahren der Tourenplanung, die im Gesamtkontext des Verfahrens ebenfalls benötigt werden, werden in dieser Arbeit vermittelt.

Die Festlegung der Lösungsmethode läßt im Detail noch zahlreiche Freiheitsgrade für die Implementierung. Dekompositionsansatz und Constraint Programming erlauben den Aufbau einer Familie von Optimierungsverfahren aus einzelnen Verfahrenselementen. Für die Untersuchung der Leistungsfähigkeit einzelner Verfahrenselemente und ihrer Kombinationen wurden in dieser Arbeit mehrere Verfahrensvarianten implementiert und anhand von Testdatensätzen empirisch verglichen. Eine grundlegende Erkenntnis der Untersuchungen ist, daß sich eine insgesamt bessere Leistung ergeben kann, wenn für einzelne Verfahrenselemente auf die bestmögliche Umsetzung verzichtet wird und stattdessen vermeintlich schwächere Methoden eingesetzt werden. Die Analyseergebnisse zeigen auch verfolgenswerte Forschungsperspektiven für nachfolgende Arbeiten auf. Beispielsweise zeigt sich, daß von einer Entwicklung von Heuristiken zur Alternativenauswahl auch das exakte Optimierungsverfahren profitieren könnte.

Neben der algorithmischen Bewertung interessiert das logistische Potential des Konzepts alternativer Lieferorte und -zeiten. Um dies zu untersuchen, werden Zustellsze-

narien in einer Modellstadt nachgebildet und analysiert. Nach Kenntnis des Verfassers wird in dieser Arbeit erstmals der Weg beschritten, für die Deklaration der Szenarienspezifikation und die Erzeugung der Instanzen Constraint Programming zu verwenden. Die Spezifikation der Modellstadt umfaßt die Aspekte Bevölkerung, Besiedelung und Geographie. Die Gesamtheit dieser Aspekte führt zu Szenarien, in denen Zustellungen zu verschiedenen Kundentypen mit entsprechenden zeitlichen und örtlichen Präferenzen geplant werden müssen. Die Zustellung an alternative Orte zu alternativen Zeiten umfaßt dabei auch die Hinterlegung von Sendungen an Abholpunkten.

Die Ergebnisse der Szenarienanalyse, der Optimallösungen von mehr als 300 Problemen mit bis zu 100 Kundenstandorte zugrunde liegen, zeigen, daß unter plausiblen Rahmenbedingungen alternative Liefermöglichkeiten im Mittel eine Kostenreduktion von knapp 20% bezogen auf den zurückgelegten Weg erlauben. Dabei kann festgestellt werden, daß weniger die Anteile einzelner Kundentypen als die Kombinierbarkeit, d.h. die Gesamtheit der angebotenen Liefermöglichkeiten, für den Erfolg ausschlaggebend ist. Für die Messung dieser Kombinierbarkeit wird erstmals ein Verbesserungsindex definiert und berechnet.

Motiviert durch die experimentellen Ergebnisse und die gewonnenen Erkenntnisse werden abschließend Themenstellungen für zukünftige Forschungsarbeiten aufgezeigt.

Literatur

- Agarwal, Y., K. Mathur und H. M. Salkin (1989). A Set-Partitioning-Based Exact Algorithm for the Vehicle Routing Problem. *Networks* 19, S. 731–749.
- Alicke, K. (2003). *Planung und Betrieb von Logistiknetzwerken*. VDI-Buch. Berlin, Heidelberg: Springer-Verlag.
- Applegate, D., R. Bixby, V. Chvátal, W. Cook und K. Helsgaun (2004). Optimal Tour of Sweden. <http://www.tsp.gatech.edu/sweden/index.html>. zuletzt besucht am 25. Februar 2005.
- Baptiste, P., C. L. Pape und W. Nuijten (1995). Incorporating Efficient Operations Research Algorithms in Constraint-Based Scheduling. In: *Proceedings of the First International Joint Workshop on Artificial Intelligence and Operations Research*, Timberline Lodge, Oregon.
- Bard, J. F., G. Kontoravdis und G. Yu (2002). An Branch-and-Cut Procedure for the Vehicle Routing Problem with Time Windows. *Transportation Science* 36(2), S. 250–269.
- Barnhart, C., C. Hane und P. H. Vance (2000). Using Branch-and-Price-and-cut to solve Origin-Destination Integer Multicommodity Flow Problems. *Operations Research* 48(2), S. 318–326.
- Barnhart, C., E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh und P. H. Vance (1998). Branch-and-price: column generation for solving huge integer programs. *Operations Research* 46(3), S. 316–329.
- Barták, R. (1998). On-line guide to Constraint Programming. <http://kti.ms.mff.cuni.cz/bartak/constraints/>. zuletzt besucht am 27. April 2005.
- Barták, R. (1999). Constraint programming: In pursuit of the holy grail.
- Battiti, R. und G. Tecchiolli (1994). The reactive tabu search. *ORSA Journal on Computing* 6(2), S. 126–140.
- Behringer, H. und B. de Backer (1995). Combinatorial problem solving in constraint logic programming with cooperating solvers. In: C. Beierle und L. Plümer (Hrsg.), *Logic Programming: Formal Methods and Practical Applications*, Studies in Computer Science and Artificial Intelligence. Elsevier.

- Benford, F. (1938). The Law of Anomalous Numbers. *Proceedings of the American Philosophical Society* 78(4), S. 551–572.
- BI-LOG (2004, April). Pressemitteilung: Premium – Lieferservice. http://www.bi-log.de/sites/presseartikel_intern.asp?ID=37, zuletzt besucht am 31. Oktober 2004.
- Bin, E., R. Emek, G. Shurek und A. Ziv (2002). Using a constraint satisfaction formulation and solution techniques for random test program generation. *IBM Systems Journal* 41(3), S. 386–402.
- Blunck, S. (2005). *Modellierung und Optimierung von Hub-and-Spoke-Netzen mit beschränkter Sortierkapazität*. Wissenschaftliche Berichte des Instituts für Fördertechnik und Logistiksysteme, Institut für Fördertechnik und Logistiksysteme, Universität Karlsruhe (TH).
- Bockmayr, A. und T. Kasper (1998). Branch and Infer: A Unifying Framework for Integer and Finite Domain Constraint Programming. *Journal on Computing* 10(3), S. 287–300.
- Bockmayr, A. und T. Kasper (2004). Branch-and-Infer: A Framework for combining CP and IP. In: M. Milano (Hrsg.), *Constraint and Integer Programming – Toward a Unified Methodology*, Volume 27 of *Operations Research/Computer Science Interface Series*, S. 58–87. Kluwer Academic Publishers.
- BTA (2002, Oktober). Wirtschaftliche Perspektiven des elektronischen Handels. Drucksache 14/10006: Bericht des Ausschusses für Bildung, Forschung und Technikfolgenabschätzung (19. Ausschuss) gemäß § 56a der Geschäftsordnung. Erstellt durch Büro für Technikfolgen-Abschätzung beim Deutschen Bundestag.
- Bundesverband des Deutschen Versandhandels e.V. (2004, Oktober). Entwicklung des E-Commerce in Deutschland. Statement des Präsidenten des Bundesverbandes des Deutschen Versandhandels (bvh e.V.), Herrn Rolf Schäfer, anlässlich der Pressekonferenz „Entwicklung des E-Commerce in Deutschland“ am 20. Oktober 2004 in Wiesbaden.
- Bundesverband Deutscher Postdienstleister e.V. (2004, April). Mitteilung: Deutsches Verkehrsforum: E-Commerce führt zu Verkehrsentlastung. http://www.bvdp.de/frontend/front_content.php?idcat=46&link=5907, zuletzt besucht am 31. Oktober 2004.
- Campbell, A. M. und M. Savelsbergh (2002). Decision Support for Consumer Direct Grocery Initiatives. *Eingereicht bei Transportation Science*.
- Campbell, A. M. und M. Savelsbergh (2003). Incentive Schemes for Attended Home Delivery Services. *Eingereicht bei Transportation Science*.
- Cardeneo, A. (2004). Personal Logistics Assistant: A platform and service for the optimization of B2C logistics. Forschungsbericht IFL–2004–2, Institut für För-

- dertechnik und Logistiksysteme, Universität Karlsruhe (TH), Karlsruhe. In Bearbeitung.
- Cardeneo, A. und C. Mohr (2003). Ein Internetportal zur Verbesserung der B2C-Distribution. *Distribution* (1-2).
- Carpaneto, G., S. Martello und P. Toth (1988). Algorithms and codes for the assignment problem. *Annals of Operations Research* 13, S. 193–223.
- Caseau, Y. und F. Laburthe (1997). Solving Small TSPs with Constraints. In: L. Naish (Hrsg.), *Proceedings of the Fourteenth International Conference on Logic Programming*, S. 316–330. MIT Press.
- Clarke, G. und J. W. Wright (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12(4), S. 568–581.
- Colmerauer, A. (1982, Oktober). PROLOG II reference manual and theoretical model. Forschungsbericht, Groupe Intelligence Artificielle, Université Aix-Marseille II.
- Cook, W. und J. L. Rich (1999). A parallel cutting-plane algorithm for the vehicle routing problem with time windows. Forschungsbericht, Computational and Applied Mathematics, Rice University.
- Cordeau, J.-F., M. Gendreau, G. Laporte, J.-Y. Potvin und F. Semet (2002, Mai). A guide to vehicle routing heuristics. *Journal of the Operational Research Society* 53(5), S. 512–522.
- Cormen, T. H., C. E. Leiserson und R. L. Rivest (1990). *Introduction to Algorithms*. The MIT electrical engineering and computer science series. Cambridge, Massachusetts: MIT Press.
- Daduna, J. R. (2000). Distribution bei Business-to-Consumer-Geschäften im Electronic Commerce. In: K. Inderfurth, M. Schenk, und D. Ziem (Hrsg.), *Logistik 2000plus: Herausforderungen, Trends, Konzepte/6. Magdeburger Logistik-Tagung*. Otto-von-Guericke-Universität Magdeburg, Lehrstuhl für Betriebswirtschaftslehre, insbesondere Produktion und Logistik; Institut für Förder- und Baumaschinentechnik, Stahlbau, Logistik.
- Dantzig, G. B. und P. Wolfe (1961). The decomposition algorithm for linear programs. *Econometrica* 29(4), S. 767–778.
- Danzas (2002). *Danzas Lotse 2002* (12. aktualisierte Auflage Aufl.). Danzas Deutschland Holding GmbH.
- Darby-Dowman, K. und J. Little (1998). Properties of some combinatorial optimization problems and their effect on the performance of integer programming and constraint logic programming. *INFORMS Journal on Computing* 10(3), S. 276–286.

- Darby-Dowman, K., J. Little, G. Mitra und M. Zaffalon (1997). Constraint logic programming and integer programming approaches and their collaboration in solving an assignment scheduling problem. *Constraints* 1(3), S. 245–264.
- de Backer, B., V. Furnon, P. Kilby, P. Prosser und P. Shaw (1997). Local Search in Constraint Programming: Application to the Vehicle Routing Problem. Datum?
- de Koster, R. B. (2002). The Logistics behind the Enter Click. In: A. Klose, M. G. Speranza, und L. N. V. Wassenhove (Hrsg.), *Quantitative approaches to distribution logistics and supply chain management*, Volume 519 of *Lecture Notes in Economics and Mathematical Systems*, S. 131 – 147. Springer-Verlag.
- Desrochers, M., J. Desrosiers und M. Solomon (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research* 40(2), S. 342–354.
- Desrosiers, J., Y. Dumas, M. M. Solomon und F. Soumis (1995). Time Constrained Routing and Scheduling. In: M. O. Ball, T. L. Magnanti, C. L. Monma, und G. L. Nemhauser (Hrsg.), *Network Routing*, Volume 8 of *Handbooks in Operations Research and Management Science*, S. 35 – 139. Amsterdam: North-Holland.
- Domschke, W. (1997). *Logistik: Rundreisen und Touren* (4. Auflage Aufl.). München, Wien: Oldenbourg-Verlag.
- Easton, K., G. Nemhauser und M. Trick (2004). CP Based Branch-and-Price. In: M. Milano (Hrsg.), *Constraint and Integer Programming – Toward a Unified Methodology*, Volume 27 of *Operations Research/Computer Science Interface Series*, S. 207–231. Kluwer Academic Publishers.
- EU (2004, März). European Union Public Opinion on Issues Relating to Business to Consumer E-Commerce. Executive Summary, Special Eurobarometer 60.0, erstellt durch European Opinion Research Group.
- Fahle, T., U. Junker, S. E. Karisch, N. Kohl, M. Sellmann und B. Vaaben (2002). Constraint Programming Based Column Generation for Crew Assignment. *Journal of Heuristics* 8(1), S. 59–81.
- Faißt, B. (2003). *Dynamische Effekte in Supply Chains – Der Bullwhip-Effekt als Ursache von Beständen bei Informationsdefiziten*. Wissenschaftliche Berichte des Instituts für Fördertechnik und Logistiksysteme, Institut für Fördertechnik und Logistiksysteme, Universität Karlsruhe (TH).
- Fichtner, C. und T. Engelsleben (2001, 7). B2C Logistik: Neue Zustellkonzepte und was die Kunden davon halten. Forschungsbericht, Simon-Kucher & Partners, Bonn.
- Fischetti, M. und P. Toth (1989). An Additive Bounding Procedure for Combinatorial Optimization Problems. *Operations Research* 37(2), S. 319–328.
- Fischetti, M. und P. Toth (1992). An additive bounding procedure for the asymmetric traveling salesman problem. *Mathematical Programming* 53, S. 173–197.

- Fleischmann, B. und M. Gietz (2002). Transport- und Tourenplanung. In: D. Arnold, H. Isermann, A. Kuhn, und H. Tempelmeier (Hrsg.), *Handbuch Logistik*, Kapitel Planung logistischer Systeme, S. A3–45 – A3–62. Berlin, Heidelberg: Springer-Verlag.
- Focacci, F., A. Lodi und M. Milano (1999). Cost-Based Domain Filtering. In: J. Jaffar (Hrsg.), *Principles and Practice of Constraint Programming – CP’99*, Lecture Notes in Computer Science, Berlin. Springer.
- Focacci, F., A. Lodi und M. Milano (2000). Cutting Planes in Constraint Programming: An Hybrid Approach. In: R. Dechter (Hrsg.), *Principles and Practice of Constraint Programming – CP 2000*, Lecture Notes in Computer Science, Berlin, S. 187 – 201. Springer.
- Focacci, F., A. Lodi und M. Milano (2004). Exploiting Relaxations in CP. In: M. Milano (Hrsg.), *Constraint and Integer Programming - Toward a Unified Methodology*, Volume 27 of *Operations Research/Computer Science Interfaces Series*, S. 137–167. Kluwer Academic Publishers.
- Focacci, F., A. Lodi, M. Milano und D. Vigo (1998). Solving TSP through the Integration of OR and CP techniques. Forschungsbericht DEIS-LIA-98-004, University of Bologna, Italy.
- Fourer, R., D. M. Gay und B. W. Kernighan (1993). *AMPL – A Modelling Language for Mathematical Programming*. San Francisco: The Scientific Press.
- Freuder, E. C. (1996). In Pursuit of the Holy Grail. *ACM Computing Surveys (CSUR)* 28(4es), S. 63.
- Funke, B. (2003, April). *Efiziente Lokale Suche für Vehicle Routing und Scheduling Probleme mit Ressourcenbeschränkungen*. Dissertation, Fakultät für Wirtschaftswissenschaften, RWTH Aachen, Aachen.
- Furmans, K. (2003). Zeitdiskrete stochastische Modelle von Logistiknetzwerken am Beispiel mehrstufiger Sammeltransporte. In: *9. Magdeburger Logistik-Tagung*. Magdeburg.
- Garey, M. R. und D. S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. New York: W. H. Freeman and Company.
- Gendreau, M., A. Hertz und G. Laporte (1992). New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research* 40(6), S. 1086–1094.
- Gietz, M. (1994). *Computergestützte Tourenplanung mit zeitkritischen Restriktionen*. Physica-Verlag. Zugleich Dissertation an der Universität Augsburg, 1993.
- Gillett, B. E. und L. R. Miller (1974). A heuristic algorithm for the vehicle routing problem. *Operations Research* 22(2), S. 340–349.
- Gilmore, P. C. und R. E. Gomory (1961). A linear programming approach to the cutting stock problem. *Operations Research* 9, S. 849–859.

- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* 5, S. 533–549.
- Glover, F. (1989). Tabu Search – Part I. *ORSA Journal on Computing* 1, S. 190–206.
- Glover, F. und M. Laguna (1997). *Tabu Search*. Boston, USA: Kluwer Academic Publishers.
- Gregori, G. (2003, Mai). E-Logistics: Last Mile Solutions. Vortrag an der Wirtschaftsuniversität Wien.
- Grönkvist, M. (2003). Using Constraint Propagation to Accelerate Column Generation in Aircraft Scheduling. Forschungsbericht, Department of Computing Science, Chalmers University of Technology, Göteborg, Schweden.
- Gudehus, T. (2002). Kommissioniersysteme. In: D. Arnold, H. Isermann, A. Kuhn, und H. Tempelmeier (Hrsg.), *Handbuch Logistik*, Kapitel Innerbetriebliche Logistik, S. C2–61 – C2–77. Berlin, Heidelberg: Springer-Verlag.
- Hahne, F. (2001, November). E-Logistik – Erfolgsfaktor für E-Commerce. Vortrag.
- Hall, R. (2004, Juni). On the Road to Recovery – 2004 survey of vehicle routing software. *OR/MS Today*.
- Harvey, W. D. und M. L. Ginsberg (1995). Limited discrepancy search. In: *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, Kanada, S. 607–615.
- Heipcke, S. (1999). *Combined Modelling and Problem Solving in Mathematical Programming and Constraint Programming*. Dissertation, School of Business, University of Buckingham.
- Helsgaun, K. (2000). An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. *European Journal of Operational Research* 126(1), S. 106–130.
- Hooker, J. (2000). *Logic-Based Methods for Optimization*. Series in Discrete Mathematics and Optimization. New York: Wiley-Interscience.
- Hooker, J. N. und M. A. Osorio (1999). Mixed logic/linear programming. *Discrete Applied Mathematics* 96-97, S. 395–442.
- Hooker, J. N., G. Ottosson, E. S. Thorsteinsson und H.-J. Kim (1999). On integrating constraint propagation and linear programming for combinatorial optimization. In: *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, S. 136–141. The AAAI Press/The MIT Press.
- ILOG (2001a). *ILOG CPLEX 8.0 User's Manual*. Gentilly, Frankreich: ILOG S.A.
- ILOG (2001b). *ILOG Dispatcher 3.3 User's Manual*. Gentilly, Frankreich: ILOG S.A.
- ILOG (2001c). *ILOG Solver 5.3 User's Manual*. Gentilly, Frankreich: ILOG S.A.

- Irnich, S. (2002). *Netzwerk-Design für zweistufige Transportsysteme und ein Branch-and-Price-Verfahren für das gemischte Direkt- und Hubflugproblem*. Dissertation, Fakultät für Wirtschaftswissenschaften, RWTH Aachen, Aachen.
- Jaffar, J. und J.-L. Lassez (1987). Constraint Logic Programming. In: *Proceedings of the 14th ACM Symposium on Principles of Programming Languages*, München, S. 111–119. ACM Press.
- Jain, V. und I. E. Grossmann (1999). Algorithms for hybrid MILP/CLP models for a class of optimization problems. Forschungsbericht, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA.
- Jonker, R. und A. Volgenant (1987). A Shortest Augmenting Path Algorithm for Dense and Sparse Assignment Problems. *Computing* 38, S. 325–340.
- Jordan, C. und A. Drexel (1995). A comparison of constraint and mixed-integer programming solvers for batch sequencing with sequence-dependent setups. *ORSA Journal on Computing* 7, S. 160–165.
- Jungnickel, D. (1999). *Graphs, Networks and Algorithms*, Volume 5 of *Algorithms and Computation in Mathematics*. Berlin, Heidelberg: Springer-Verlag.
- Junker, U., S. E. Karisch, N. Kohl, B. Vaaben, T. Fahle und M. Sellmann (1999). A framework for constraint programming based column generation. In: *Principles and Practice of Constraint Programming*, S. 261–274.
- Kaluza, B., H. Dullnig und F. Malle (2003). Principal-Agent-Probleme in der Supply Chain. Forschungsbericht 2003/03, Institut für Wirtschaftswissenschaften der Universität Klagenfurt.
- Kindervater, G. A. P. und M. W. P. Savelsbergh (1997). Vehicle routing: Handling edge exchanges. In: E. Aarts und J. K. Lenstra (Hrsg.), *Local Search in Combinatorial Optimization*. John Wiley & Sons.
- Kirkpatrick, S., J. C. D. Gelatt und M. P. Vecchi (1983). Optimization by simulated annealing. *Science* 220(4598), S. 671–680.
- Kämäräinen, V. (2003). *The impact of investments on e-grocery logistics operations*. Dissertation, Helsinki University of Technology, Industrial Engineering and Management, Espoo, Finland.
- Kämäräinen, V., J. Småros, T. Jaakola und J. Helmström (2001). Cost-effectiveness in the e-grocery business. *International Journal of Retail & Distribution Management* 29(1), S. 41–48.
- Kohl, N. (1995). *Exact methods for time constrained routing and related scheduling problems*. Dissertation, Institut for Matematisk Modellering, Danmarks Tekniske Universitet, Lyngby, Dänemark.
- Kohl, N. (1999). Application of OR and CP Techniques in a Real World Crew Scheduling System. Forschungsbericht, Carmen Systems AB, Göteborg, Schweden.

- Kohl, N. und O. B. G. Madsen (1997). An Optimization Algorithm for the Vehicle Routing Problem with Time Windows based on Lagrangian Relaxation. *Operations Research* 45(3), S. 395–406.
- Kummer, S. (2002). Strategisches Logistik-Controlling. In: D. Arnold, H. Isermann, A. Kuhn, und H. Tempelmeier (Hrsg.), *Handbuch Logistik*, Kapitel Logistikmanagement, S. D5–14 – D5–27. Berlin, Heidelberg: Springer-Verlag.
- Kunze, O. (2005). Ein praxistauglicher Ansatz zur Lösung eines spezifischen D-VRSP-TW-UC. In: *Operations Research Proceedings 2004*. Springer-Verlag.
- Laffont, J.-J. und D. Martimort (2001). *The Theory of Incentives: The Principal-Agent Model*. Princeton University Press.
- Land, A. und A. Doig (1960). An automatic method os solving discrete programming problems. *Econometrika* 28(3), S. 497–520.
- Laporte, G., M. Gendreau, J.-Y. Potvin und F. Semet (1999). Classical and Modern Heuristics for the Vehicle Routing Problem. Forschungsbericht Les Cahiers du GERAD G-99-21, GERAD, Montréal, Kanada.
- Larsen, J. (1999). *Parallelization of the Vehicle Routing Problem with Time Windows*. Dissertation, Department of Mathematical Modelling, Technical University of Denmark.
- Lasch, R. und A. Lemke (2002). Logistik im B2C-Bereich. Forschungsbericht, BWL-Lehrstuhl für Logistik, TU Dresden, Dresden. Marktstudie unter Onlineshops.
- Laseter, T., P. Houston, A. Chung, S. Byrne, M. Turner und A. Devendran (2000). The Last Mile to Nowhere: Flaws & Fallacies in Internet Home-Delivery Schemes.
- Lin, S. (1965). Computer Solutions of the Traveling Salesman Problem. *Bell System Technical Journal* 44, S. 2245–2269.
- Lin, S. und B. W. Kernighan (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research* 21, S. 498–516.
- Lippolt, C. (2003). *Spielzeitberechnung in doppeltiefen Lagern*. Wissenschaftliche Berichte des Instituts für Fördertechnik und Logistiksysteme, Institut für Fördertechnik und Logistiksysteme, Universität Karlsruhe (TH).
- Löbel, A. (1997). *Optimal Vehicle Scheduling in Public Transit*. Dissertation, Fachbereich 3 Mathematik, Technische Universität Berlin.
- Lodi, A. und M. Milano (2003). Discrepancy-based additive bounding. In: *Proceedings of the CP-AI-OR 2003*, Montréal, Canada, S. 17–23.
- Logistik heute (1998, September). Tourenplanungssysteme – Auswahl leicht gemacht. *Logistik Heute* (9), S. 76–83.

- Lübbecke, M. (2001). *Engine Scheduling by Column Generation*. Dissertation, Fachbereich Mathematik und Informatik, Technische Universität Braunschweig.
- Lübbecke, M. (2004, April). persönliche Mitteilung.
- Marriott, K. und P. J. Stuckey (1998). *Programming with Constraints: An Introduction*. Cambridge, MA, USA: MIT Press.
- Martin, R. K. (1999). *Large scale linear and integer optimization: a unified approach*. Kluwer Academic Publishers.
- Milano, M. (2004). Constraint and Integer Programming. In: M. Milano (Hrsg.), *Constraint and Integer Programming - Toward a Unified Methodology*, Volume 27 of *Operations Research/Computer Science Interfaces Series*, S. 1–31. Kluwer Academic Publishers.
- Mucha, A., J. Spille und P. Wader (2002, Januar). Endkundenbelieferung – Neue Konzepte in der letzten Meile. Forschungsbericht, Forschungsinstitut für Rationalisierung, Aachen.
- mylogistics.net (2004, Mai). SAP entlastet interne Logistik durch Packstation. <http://www.mylogistics.net/de/news/themen.jsp?key=news128782>, zuletzt besucht am 21. Mail 2004.
- Nash, S. G. und A. Sofer (1996). *Linear and Nonlinear Programming*. Industrial Engineering Series. McGraw-Hill.
- Network, M. (1999). WWW-Seite des Metaheuristic Network Projekts. <http://www.metaheuristics.net>.
- Newcomb, S. (1881). Note on the Frequency of Use of the Different Digits in Natural Numbers. *American Journal of Mathematics* 4, S. 39–40.
- Nieschlag, R., E. Dichtl und H. Hörschgen (1994). *Marketing* (17., neu bearbeitete Auflage Aufl.). Berlin: Duncker & Humblot.
- Nuijten, W. (1994). *Time and Resource Constraint Scheduling, a Constraint Satisfaction Approach*. Dissertation, Universität Eindhoven, Eindhoven, NL.
- Oberhuber, N. (2003). Hallo? Niemand zu Hause. *Die Zeit* (27).
- Or, I. (1976). *Traveling Salesman-Type Combinatorial Problems and Their Relation to the Logistics of Regional Blood Banking*. Dissertation, Northwestern University, Evanston, Illinois, USA.
- Ottosson, G. (2000). *Integration of Constraint Programming and Integer Programming for Combinatorial Optimization*. Dissertation, Uppsala University.
- Ottosson, G. und E. S. Thorsteinsson (2000). Linear relaxations and reduced-cost based propagation of continuous variable subscripts. In: *CP-AI-OR'00 Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems*.

- Ottosson, G., E. S. Thorsteinsson und J. N. Hooker (1999). Mixed global constraints and inference in hybrid CLP-IP solvers. In: *CP99 Post-Conference Workshop on Large Scale Combinatorial Optimisation and Constraints*.
- Pesant, G. und M. Gendreau (1999). A Constraint Programming Framework for Local Search Methods. *Journal of Heuristics* 5, S. 255–279.
- Pesant, G., M. Gendreau, J.-Y. Potvin und J.-M. Rousseau (1998, 2). An Exact Constraint Logic Programming Algorithm for the Traveling Salesman Problem with Time Windows. *Transportation Science* 32(1), S. 12–29.
- Pesant, G., M. Gendreau, J.-Y. Potvin und J.-M. Rousseau (1999). On the flexibility of constraint programming models: From single to multiple time windows for the traveling salesman problem. *European Journal of Operational Research* (117), S. 253–263.
- Pesant, G., M. Gendreau und J.-M. Rousseau (1997). GENIUS-CP: A Generic Vehicle Routing Algorithm. In: *Principles and Practice of Constraint Programming CP97: Proceedings of the Third International Conference*, Volume 1330 of *Lecture Notes in Computer Science*, S. 420–434. Springer-Verlag.
- PickPoint AG (2005). Homepage der PickPoint AG. <http://www.pickpoint.de/>, zuletzt besucht am 7. Februar 2005.
- Plattform Versandhandel (2003). Ein kleiner Streifzug durch die Geschichte – Der Versandhandel ist ein Spiegel des technischen Fortschritts. Pressekonzferenz Versandhandel 24. September 2003. <http://www.versandhandel-online.at/Presse/presse28.htm>, zuletzt besucht am 19.12.2004.
- Prockl, G. (2001). Die letzte Meile - E-Commerce und seine Herausforderungen an die Logistik. *Schwäbischer E-Commerce Tag*.
- Punakivi, M. (2003). *Comparing alternative home delivery models for e-grocery business*. Dissertation, Helsinki University of Technology, Department of Industrial Engineering and Management, Helsinki.
- Rahn, K.-P. (2004). *Optimierung der Distributionslogistik im Bereich der Kurier-, Express- und Paketdienstleister*. Dissertation, Institut für Fördertechnik und Logistik, Fakultät für Maschinenbau der Universität Stuttgart, Stuttgart.
- Retail Logistics Task Force (2001). @Your Home – New Markets for Consumer Service and Delivery. Forschungsbericht, UK Government Foresight program, UK.
- Régin, J.-C. (1994). A filtering algorithm for constraints of difference in CSPs. In: *Proceedings of the National Conference on Artificial Intelligence*, S. 362–367.
- Rindfleisch, K.-H. (2005). Betriebliche Aspekte der Packstation. persönliche Mitteilung. KEBA AG, Linz (Österreich).

- Rodošek, R., M. Wallace und M. T. Haijan (1999). A new approach to integrating mixed integer programming and constraint logic programming. *Annals of Operations Research* 86, S. 63–87.
- Rousseau, L. M. (2004). Stabilization Issues in Constraint Programming Based Column Generation. In: J.-C. Régim und M. Rueher (Hrsg.), *CPAIOR 2004*, Volume 3011 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, S. 402–408. Springer-Verlag.
- Rousseau, L.-M. und M. Gendreau (2002). Using Constraint-Based Operators to Solve the Vehicle Routing Problem with Time Windows. *Journal of Heuristics* 8, S. 43–58.
- Rousseau, L.-M., M. Gendreau und G. Pesant (2002). Solving small VRPTWs with Constraint Programming Based Column Generation. In: *Proceedings CPAIOR'02*.
- Ryan, D. M. und B. A. Foster (1981). An Integer Programming Approach to Scheduling. In: A. Wren (Hrsg.), *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*, S. 269–280. Amsterdam: North-Holland.
- Savelsbergh, M. W. P. (1990). An efficient implementation of local search algorithms for constrained routing problems. *European Journal of Operational Research* 47, S. 75–85.
- Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In: *Principles and Practice of Constraint Programming*, Volume 1520 of *Lecture Notes in Computer Science*. Springer.
- Siebel, L. und M. ten Hompel (2001). *Logistik und E-Commerce – Konzepte für Ballungszentren*. Praxiswissen.
- Simonis, H. (2001). Building Industrial Applications with Constraint Programming. In: H. Comon, C. Marché, und R. Treinen (Hrsg.), *Constraints in Computational Logic 1999*, Volume 2002 of *Lecture Notes in Computer Science*, S. 271–309. Berlin, Heidelberg: Springer-Verlag.
- Sol, M. (1994). *Column Generation Techniques for Pickup and Delivery Problems*. Dissertation, Technische Universität Eindhoven, Eindhoven, NL.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* 35, S. 254 – 265.
- SSI Schäfer (2003, März). Pressemitteilung: Die Zukunft für das Online-Shopping.
- Ströbel, M. (2003). *Engineering Electronic Negotiations*. Series in Computer Science. New York: Kluwer Academic /Plenum Publishers.
- Thienel, S. (1995). *ABACUS – A Branch-and-CUt System*. Dissertation, Universität zu Köln.

- Thomas, H. (2002). Kurier-, Express- und Paketdienste. In: D. Arnold, H. Isermann, A. Kuhn, und H. Tempelmeier (Hrsg.), *Handbuch Logistik*, Kapitel Außerbetriebliche Logistik, S. C3–70 – C3–76. Berlin, Heidelberg: Springer-Verlag.
- TNS (2002). Global eCommerce Report 2002. Bericht des Marktforschungsunternehmens Taylor Nelson Sofres Interactive.
- Toth, P. und D. Vigo (2002). *The Vehicle Routing Problem*. SIAM monographs on discrete mathematics and applications. Philadelphia, USA: SIAM.
- van Hentenryck, P. (1989). *Constraint Satisfaction in Logic Programming*. Logic Programming Series. Cambridge, MA, USA: MIT Press.
- van Hentenryck, P. (1999). *The OPL Optimization Programming Language*. Cambridge, MA, USA: MIT Press.
- Vanderbeck, F. (1994). *Decomposition and Column Generation for Integer Programs*. Dissertation, Université Catholique de Louvain.
- VDI 3633 (1993, Dezember). Simulation von Logistik-, Materialfluß- und Produktionssystemen.
- Verkehrsforum, D. (2004, April). Last-Mile-Logistics: Best Practices. Studie im Auftrag des Deutschen Verkehrsforums, erstellt durch TransCare AG, Berlin.
- Verra (2004). B2C-Handel: Qualitative und quantitative Analyse des Konsumenten- und Anbieterverhaltens und deren verkehrliche Auswirkungen in Ballungsräumen. Abschlußbericht des Projekts B2C-Verra, gefördert durch das BMBF in Zusammenarbeit mit der Stadt Köln, erstellt durch KE-Consult.
- Vogt, M. (1998). *Tourenplanung in Ballungsgebieten*. Kasseler Wirtschafts- und Verwaltungswissenschaften, Band 6. Wiesbaden: Deutscher Universitätsverlag. Zugleich Dissertation Universität Kassel, 1997.
- Voß, S. (1999). *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Boston: Kluwer.
- Wagener & Herbst Management Consultants GmbH (2002, Juni). Der Transportmarkt im Wandel – Ergebnisse der Delphi-Studie. Ergebnisbericht. Delphi-Studie durchgeführt von Wagener & Herbst Management Consultants und der TU Dresden im Auftrag des Deutschen Verkehrsforums.
- Williams, H. P. (1999). *Model building in Mathematical Programming* (4. Aufl.). John Wiley & Sons.
- Wolsey, L. A. (1998). *Integer Programming*. Wiley-Interscience.
- Zillus, A. (2003). *Untersuchung der Wartezeit von Kundenaufträgen in der Supply Chain*. Wissenschaftliche Berichte des Instituts für Fördertechnik und Logistiksysteme, Institut für Fördertechnik und Logistiksysteme, Universität Karlsruhe (TH).

