Modelle und Lösungsverfahren für die integrierte Ressourceneinsatzplanung im öffentlichen Personennahverkehr

DISSERTATION

zur Erlangung des akademischen Grades
Dr. rer. pol.
im Fach Wirtschaftsinformatik

eingereicht an der
Fakultät für Wirtschaftswissenschaften
Universität Paderborn

von
Herr Dipl.-Wirt.-Inf. Vitali Gintner
geboren am 06.07.1976 in Syktywkar

Gutachter:

1. Prof. Dr. Leena Suhl
2. Prof. Dr.-Ing. habil. Wilhelm Dangelmaier

eingereicht am: 11. Januar 2008

Danksagung

Die vorliegende Arbeit entstand in der Zeit, die ich als Stipendiat der International Graduate School of Dynamic Intelligent Systems am Decision Support & Operations Research (DSOR) Lab der Universität Paderborn verbrachte. Ich möchte mich an dieser Stelle bei allen Menschen bedanken, die zum Gelingen dieser Arbeit beigetragen haben.

Zuerst gilt mein besonderer und herzlicher Dank Prof. Dr. Leena Suhl für die freundliche und engagierte Betreuung. Ich danke Leena ganz herzlich für die Möglichkeit, durch ihre Vorlesungen und Motivation die faszinierende Welt von Operations Research kennenlernen zu dürfen. Das bestimmte meine Vertiefung im Studium, sorgte für eine hochinteressante Promotionszeit und prägte schließlich mein späteres Berufsleben.

Ich möchte mich außerdem bei allen Kollegen am DSOR-Lehrstuhl für die gute Zusammenarbeit, hervorragende Forschungsatmosphäre und viele interessante Diskussionen, sowohl auf N4 als auch während zahlreicher Ausflüge bedanken. Mein besonderer Dank gilt Ingmar Steinzen für die perfekte Arbeitsatmosphäre in unserem Zimmer, viele interessante Diskussionen und seine freundliche Unterstützung bei der inhaltlichen Korrektur der Arbeit. Außerdem möchte ich mich bei Junior-Prof. Dr. Natalia Kliewer für ihre hilfreichen Ratschläge und die Durchsicht der Arbeit bedanken.

Ich danke auch dem International Graduate School of Dynamic Intelligent Systems, insbesondere dem Leiter PD. Eckhard Steffen und dem ganzen Organisationsteam für die engagierte Begleitung und finanzielle Unterstützung meiner Arbeit.

Des Weiteren danke ich Inna Dischke und Georg Kliewer für ihre freundliche Unterstützung bei der Korrektur des Manuskripts.

Ein besonders herzlicher Dank gilt meinen lieben Eltern und meiner ganzen Familie, die mir mit ihrer Liebe und bedingungslosem Rückhalt unter anderem das Studium und die Promotion erst ermöglichten und mir jederzeit zur Seite standen. Ich danke Viktoria und allen meinen Freunden, die meiner Arbeit viel Verständnis und Geduld entgegenbrachten und mich in allen Umbrüchen und Veränderungen stärkten und unterstützten.
Vielen herzlichen Dank!

Paderborn, im Januar 2008 Vitali Gintner

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung und Motivation

Der öffentliche Personennahverkehr (ÖPNV) ist ein unverzichtbarer Bestandteil unserer Mobilitäts- und Alltagskultur. Ein attraktiver und leistungsfähiger ÖPNV ist unmittelbar mit der Lebensqualität und Urbanität von Städten verbunden. Er entlastet die Ballungsräume vom Individualverkehr, trägt zur Reduzierung klimarelevanter Emissionen bei und gewährleistet gleichwertige Lebensverhältnisse in den Regionen. Für viele Städte ist ein erfolgreiches Stadtbussystem ein unverzichtbarer Wirtschafts- und Standortfaktor. Laut dem Bundesministerium für Verkehr, Bau und Stadtentwicklung nutzen 26 Millionen deutscher Bürgerinnen und Bürger täglich den ÖPNV. Mehr als 250.000 Beschäftigte in rund 6.000 privaten und kommunalen Verkehrsunternehmen erfüllen diese öffentliche Aufgabe (Stand 2000, vgl. [BMVBS, 2000]).

In den letzten Jahren erlebte der ÖPNV auf Deutschlands Straßen tiefgreifende Veränderungen. Mit dem Inkrafttreten des Gesetzes zur Regionalisierung des öffentlichen Personennahverkehrs (RegG) begann die Strukturreform des deutschen ÖPNV. Die Länder und Kommunen bekamen die Zuständigkeit für die Planung und Organisation des gesamten ÖPNV. Doch sie müssen nicht nur planen und organisieren, sondern auch die volle finanzielle Verantwortung übernehmen. Die allgemein leeren öffentlichen Kassen und fortlaufenden Kürzungen der vom Bund zur Verfügung gestellten Förderungen erfordern eine konsequente Ausnutzung von Einsparpotenzialen. Auch die weitgehende Liberalisierung und Deregulierung der europäischen Verkehrsmärkte sowie der Wegfall von Quersubventionierungen erhöhen den Druck auf kommunalen Aufgabenträger, durch öffentliche Ausschreibungen den kostengünstigsten Anbieter von ÖPNV-Leistungen zu finden. Dennoch soll die Qualität der Dienstleistung nicht leiden.

Die ersten Folgen der europaweiten Öffnung der ÖPNV-Märkte sind bereits in Deutschland zu sehen. Die ersten ausländischen Anbieter, die bereits Erfahrungen mit deregulierten Verkehrsmärkten gesammelt haben und im Vergleich zu kom-

munalen und privaten deutschen Betreibern Größenvorteile aufweisen, erschließen mit Erfolg den deutschen Markt. So gehört beispielsweise Veolia Transport (früher Connex) zu den führenden privaten Nahverkehrsanbietern in Deutschland. Laut Firmenwebseite ist die deutsche Tochter Veolia Verkehr-Gruppe an 40 Verkehrsunternehmen beteiligt und beschäftigt im Personenverkehr rund 4250 Mitarbeiter. Ein weiteres Beispiel ist der größte französische Nahverkehrdienstleiser Keolis als Teil der Rhenus-Keolis GmbH & Co KG.

Die privaten Verkehrsunternehmen müssen sich den neuen Rahmenbedingungen stellen. Eine Bestandsgarantie für einzelne Verkehrsunternehmen kann es nicht geben. Durch den offenen Wettbewerb um den Markzugang werden sie immer stärker unter Druck gesetzt, die Kosten zu senken, die Produktivität zu steigern und das Angebot zu verbessern. Auf der anderen Seite erzwingt der Wettbewerb eine hohe Flexibilität und Reaktionsgeschwindigkeit, sich auf verändernde Rahmenbedingungen schnell zu reagieren. Die angebotenen Leistungen müssen ständig überprüft und gegebenenfalls schnell angepasst werden. Die Möglichkeit, verschiedene Szenarien zu simulieren und Sensitivitätsanalysen durchzuführen, gewinnt immer mehr an Bedeutung.

Dabei spielen computergestützte Planungswerkzeuge auf Basis von Methoden aus dem Bereich des Operations Research eine entscheidende Rolle. In vielen modernen ÖPNV-Betrieben sind sie heutzutage nicht mehr wegzudenken. Mit der Weiterentwicklung von Computertechnik und mathematischer Optimierung werden solche Werkzeuge immer leistungsfähiger und bieten Verkehrsbetrieben immer mehr Unterstützung in unterschiedlichen Phasen im umfangreichen Planungsprozess. Dabei fällt ein ökonomischer und möglichst effizienter Ressourceneinsatz immer mehr in den Fokus, da die beiden Hauptressourcen eines Verkehrsbetriebs, nämlich Fahrzeuge und Personal, die größten Kostenfaktoren darstellen.

Der Einsatz von Optimierungsmethoden bei der Umlauf- und Dienstplanung wird schon seit längerer Zeit erforscht. Es existiert eine Reihe leistungsstarker Verfahren, die in kommerziellen Planungswerkzeugen integriert sind und eine hervorragende Hilfe bei der effizienten Planung von Fahrzeugen und Fahrern leisten. Allerdings verfolgen sie, ähnlich zu der manuellen Planung, eine streng sequenzielle Abarbeitung der beiden Planungsschritte: Zuerst werden die Umläufe für Fahrzeuge geplant und darauf basierend Dienste für die Fahrer. Auf der anderen Seite ist bekannt, dass eine simultane Betrachtung der beiden Planungsschritte einen weiteren Schritt in Richtung effizienter Ressourceneinsatzplanung geht und zusätzliches Einsparpotenzial ermöglicht.

Die Zielsetzung der vorliegenden Arbeit besteht in der Konzeption und Implementierung von effizienten Modellen und Lösungsverfahren für eine gekoppelte und vollständig integrierte Behandlung von Umlauf- und Dienstplanung mit mehreren

Depots. Die Arbeit gliedert sich in neun Kapitel, wobei die Kapitel fünf bis neun den inhaltlichen Kern der Arbeit bilden.

Kapitel 2 behandelt die Probleme der Umlauf- und Dienstplanung als Teile der Ressourceneinsatzplanung im ÖPNV. Um diese Aufgaben in den Gesamtzusammenhang des Planungsprozesses einzuordnen, wird zuerst der Gesamtprozess beschrieben. Danach werden die Probleme der Umlauf- und Dienstplanung sowohl einzeln als auch in einem integrierten Kontext ausführlich behandelt.

Einen Exkurs in die Welt des Operations Research bietet Kapitel 3. Es beginnt mit einer kurzen Vorstellung einiger Optimierungsprobleme, die zum Verständnis dieser Arbeit relevant sind. Danach werden ausgewählte Lösungstechniken diskutiert, die in den entwickelten Lösungsverfahren eingesetzt werden.

Kapitel 4 gibt einen Überblick über die bereits veröffentlichten Modellierungs- und Lösungsansätze aus der Literatur für beide Planungsprobleme sowohl bei sequenzieller als auch bei gleichzeitigen Betrachtung. Dabei werden die integrierten Modelle je nach Grad der Integration klassifiziert.

In Kapitel 5 wird einer der zentralen, im Rahmen der vorliegenden Arbeit entwickelten Ansätze zur Lösung integrierter Umlauf- und Dienstplanung vorgestellt. Zunächst wird ein neues Modell für das integrierte Umlauf- und Dienstplanungsproblem mit mehreren Depots beschrieben. Das zugrunde liegende Netzwerkmodell wird mit einer neuartigen Modellierungstechnik als Time-Space-Netzwerk formuliert, die dank ihrer Struktur zu einer erheblichen Reduktion der Netzwerkgröße und der daraus abgeleiteten mathematischen Formulierung führt. Der Lösungsansatz basiert auf einer Kombination von einem Column-Generation-Verfahren und Lagrange-Relaxation. Er wird in nachfolgenden Abschnitten des Kapitels detailliert beschreiben. Dabei werden zu jedem Schritt mehrere Lösungsalternativen bzw. unterschiedliche Formulierungen untersucht.

Einen weitereren Schwerpunkt der Arbeit bildet der in Kapitel 6 vorgestellte Ansatz zur adaptiven Teilintegration von Umlauf- und Dienstplanung. Trotz einer sequenziellen Vorgehensweise ermöglicht er eine gewisse Interaktion zwischen den beiden Planungsproblemen. Die derartige Kopplung hilft in etwa vergleichbarer Laufzeit deutlich bessere Gesamtlösungen als bei der rein sequenziellen Vorgehensweise bei der Verplanung von Umläufen und Diensten zu finden.

Die dritte tragende Säule der vorliegenden Arbeit ist die in Kapitel 7 vorgestellte Erweiterung des Lösungsansatzes aus Kapitel 5 für große Probleme. Es handelt sich um ein approximatives Verfahren, das vor der eigentlichen Behandlung des Umlauf- und Dienstplanungsproblems mit dem integrierten Ansatz die Problemgröße durch Fixierung einiger Fahrten heuristisch verkleinert.

In Kapitel 8 werden die Lösungszeiten und Lösungsqualitäten der drei entwi-

ckelten Hauptverfahren für unterschiedliche Testinstanzen gegenübergestellt. Außerdem wird die Kombination aller drei Ansätze in einem einzigen Lösungsverfahren untersucht.

Kapitel 9 enthält eine Zusammenfassung der Arbeit mit einem Ausblick und Vorschlägen für weitere Forschungsaktivitäten in diesem Bereich.

Kapitel 2

Umlauf- und Dienstbildung als Aufgaben der operativen ÖPNV-Planung

Dieses Kapitel behandelt die Probleme der Umlauf- und Dienstplanung als Teile der Ressourceneinsatzplanung im ÖPNV. Um diese Aufgaben in den Gesamtzusammenhang des Planungsprozesses einzuordnen, wird zuerst der Gesamtprozess beschrieben. Danach werden die Probleme der Umlauf- und Dienstplanung sowohl einzeln als auch in einem integrierten Kontext ausführlicher behandelt.

Unter einem ÖPNV-Unternehmen werden im Rahmen dieser Arbeit Verkehrsunternehmen verstanden, die Fahrgastbeförderung mit Bussen auf bestimmten Linien anbieten. Als eine *Linie* wird eine zwischen bestimmten Ausgangs- und Endpunkten eingerichtete regelmäßige Verkehrsverbindung bezeichnet, auf der die Fahrgäste an bestimmten Haltestellen ein- und aussteigen können. Weiterhin wird davon ausgegangen, dass die Linienfahrten in einem bestimmten Zeitraster angeboten werden. Eine Linie wird also nicht nur mit einer bestimmten Frequenz bedient, wie es in anderen Verkehrsformen[1] oder anderen Ländern[2] üblich ist, sondern auch zu den vorher festgelegten Zeitpunkten.

[1] Z.B. werden bei U-Bahnen in großen Städten oft keine genaue Angaben zur Ankunftszeit, sondern nur eine Taktfrequenz veröffentlicht.

[2] Z.B. wird in Russland auch bei Linienbussen im Nahverkehr nur eine Taktfrequenz vorgegeben.

2.1 Operativer Planungsprozess im ÖPNV

Das Ziel der Planung im ÖPNV ist eine möglichst kostengünstige und trotzdem hochqualitative Bedienung der Reisewünsche von Personen im Nahverkehr. Die Betriebsplanung öffentlicher Verkehrsunternehmen ist aufgrund zahlreicher Anforderungen, die es zu berücksichtigen gilt, ein komplexer Prozess. Er lässt sich grundlegend in *strategische* und *operative* Planung unterteilen.

Die *strategische* ÖPNV-Planung besteht aus zwei Teilaufgaben: der Netz- und Linienplanung. Den beiden Phasen liegt eine grobe Angebotsplanung, die aufgrund der geschätzten Passagierströme vorgenommen wurde. Basierend auf diesen Daten wird zunächst die Infrastruktur des Nahverkehrsnetzes, das Streckennetz, entworfen (***Netzplanung***), wovon anschließend die Verläufe einzelner Linien und ihre Fahrzeitprofile abgeleitet werden (***Linienplanung***). Dabei wird das Ziel verfolgt, mit einer oft begrenzten Anzahl von Linien ein Maximum an Direktverbindungen (d.h. Verbindungen ohne Umsteigenotwendigkeit) anzubieten. Die Fahrzeitprofile beschreiben den Fahrzeitbedarf für die einzelnen Fahrtabschnitte einer Linie (die erforderlichen Fahrzeiten können sich im Laufe des Tages unterscheiden).

Die strategische ÖPNV-Planung liegt typischerweise in der kommunalen Hand (Aufgabenträger), die sowohl für die Organisation des Wettbewerbs im straßengebundenen Nahverkehr als auch für die jeweilige Finanzierung des vor Ort für nötig befundenen gemeinwirtschaftlichen Verkehrsangebots verantwortlich ist. Der Aufgabenträger vergibt die Aufträge zur Bedienung einzelner Linien bzw. ganzer Linienbündel in einem Auschreibungsverfahren. Die Verkehrsunternehmen (Aufgabennehmer) können eine Konzession auf bestimmte Linien erwirken, indem sie sich auf die entsprechende Ausschreibung bewerben. Das günstigste Angebot erhält dabei den Zuschlag. So wird im Prozess der Deregulierung des Marktes für ÖPNV eine Vermischung von unternehmerischer und öffentlicher Verantwortung abgeschafft. Durch den offenen Wettbewerb bei den Ausschreibungen wird seitens der Verkehrsbetriebe eine höhere ökonomische Effizienz gefordert. Daher ist eine möglichst effiziente Verplanung der verfügbaren Ressourcen (Fahrzeuge und Fahrer) enorm wichtig.

Die *operative* ÖPNV-Planung ist dagegen eine Aufgabe von Verkehrsunternehmen. Die Planungsprobleme der operativen ÖPNV-Planung sind vielfältig, hier geht es sowohl um die Festlegung eines Produktionsprogramms (Fahrplan) als auch um die dazu benötigten Verbrauchsfaktoren (Fahrzeuge und Fahrer) und den Ablauf der Produktionsprozesse (Einsatzplan).

Eine gesamtoptimale, den wechselseitigen Abhängigkeiten gerechte ÖPNV-Planung unter Beachtung aller Nebenbedingungen würde eine simultane Betrachtung aller Teilaufgaben erfordern. Allerdings stößt die Aufgabe schon wegen der Komple-

xität der einzelnen Teilprobleme auf kaum zu überwindende Schwierigkeiten. Daher wird das Gesamtproblem in separate, aufeinander folgende, beherrschbare Teilaufgaben, nämlich *Fahr-, Umlauf-, Dienst-* und *Dienstreihenfolgeplanung*, zerlegt. Die meisten kommerziellen, auf mathematischer Optimierung basierenden Planungstools verfolgen eine streng sequenzielle Abarbeitung der einzelnen Planungsphasen, wobei das in einer Stufe entwickelte Ergebnis eine notwendige Voraussetzung für die nächste Stufe der Planung ist (siehe Abbildung 2.1). Nichtsdestotrotz geht die aktuelle Entwicklung in Richtung einer simultanen Betrachtung, wenn nicht des Gesamtproblems, dann zumindest einzelner Teilaufgaben. Dies ist dank der rasanten Entwicklung von Computertechnik und Methoden der mathematischen Optimierung in der letzten Zeit überhaupt erstmal möglich geworden. Als Hauptbeitrag der vorliegenden Arbeit wird eine simultane Betrachtung von Umlauf- und Dienstplanung behandelt.

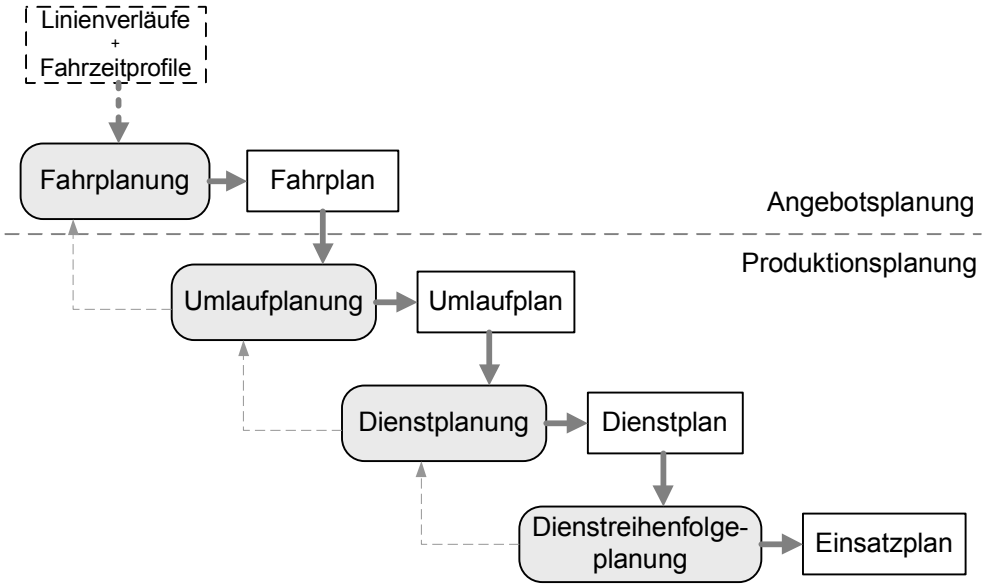

Abbildung 2.1: Teilaufgaben der operativen ÖPNV-Planung

Bei der ersten Stufe der operativen Planung, der ***Fahrplanung***, wird aus der räumlichen Struktur eines Liniennetzes und der zeitlichen Struktur des Fahrplans eine Menge von *Personenbeförderungsfahrten (Fahrgastfahrten)* zwischen zwei Endhaltestellen mit fest definierten Abfahrtszeiten abgeleitet. Das Ziel der Fahrplanerstellung ist eine optimale Gestaltung von Anschluss- und Umsteigebeziehungen. Außerdem wird die Vermeidung bzw. Minimierung der gegenseitigen Behinderung von Fahrplanfahrten auf gemeinsam benutzten Straßen und eine optimale Ergänzung der Fahrten im Gesamtverkehrssystem angestrebt. Bei Straßenbahnen ist es beispielsweise nicht möglich, ein Gleis zur gleichen Zeit doppelt zu benutzen; bei Bussen muss auf die Kapazitäten der Haltestellen geachtet werden. In dieser Phase

werden auch die zulässigen Fahrzeugtypen für die einzelnen Fahrten festgelegt, falls sie nicht für alle Fahrten einer Linie gleich sein sollen.

Die Planungsaufgaben Netz-, Linien- und Fahrplanung werden häufig unter dem Oberbegriff *Angebotsplanung* zusammengefasst, während die drei nachfolgenden Stufen, Umlauf-, Dienst- und Dienstreihenfolgeplanung üblicherweise als *Produktionsplanung* bezeichnet werden. Die Ergebnisse der Angebotsplanung werden als Netzplan, Linienplan und Fahrplan veröffentlicht. Im Gegensatz dazu dienen die Produktionspläne nur der innerbetrieblichen Leistungserstellung und brauchen dem potenziellen Kunden nicht bekannt gemacht zu werden.

Bei der ***Fahrzeugumlaufplanung*** (auch *Umlaufplanung* oder *Fahrzeugeinsatzplanung*) werden die vom Fahrplan vorgegebenen Fahrgastfahrten den vorhandenen Fahrzeugen zugeordnet. Dabei kann ein oder mehrere Ziele verfolgt werden. Wird bei der Umlaufplanung nur die Einhaltung der vordefinierten Regeln in den Vordergrund gestellt, um somit die Gültigkeit und die Qualität des resultierenden Umlaufplans sicherzustellen, spricht man von *regelbasierter* Planung der Umläufe. Wird stattdessen bzw. zusätzlich dazu die bestmögliche Erfüllung eines oder mehrerer Ziele verfolgt, heißt dies ein *zielbasiertes* Vorgehen. Die möglichen Ziele sind dabei eine Minimierung der Anzahl einzusetzender Fahrzeuge und/oder eine Minimierung der zu fahrenden Leerkilometer (Fahrten ohne Personenbeförderung). Das regelbasierte Vorgehen ist dank seiner Einfachheit noch heute sehr stark verbreitet, während die wirtschaftlichorientierte, zielbasierte Planung eine viel komplexere Herausforderung darstellt und ohne Computerunterstützung in der Regel nicht realisierbar ist.

Bei der ***Dienstplanung*** geht es darum, die eingeplanten Fahrzeuge mit Fahrpersonal zu besetzen. Dafür werden die vorgegebenen Umläufe in kleinere Einheiten (*Dienststücke*) „geschnitten", die von einem Fahrer am Stück, ohne Pausenunterbrechung bedient werden können. Solche Dienststücke werden schließlich unter Einhaltung der Dienst- und Betriebsvorschriften aneinandergereiht und ergeben den Tagesdienst eines Fahrers. Die Vielfalt aller einzuhaltenden Dienstregel (wie z.B. minimale und maximale Dienstlänge, Arbeitszeit, ununterbrochene Lenkzeit, Pausendauer usw.) ist sehr groß und macht das Problem der Dienstplanung besonders komplex. Dabei werden oft außer Regeln für einzelne Dienste auch bestimmte Anforderungen an die Zusammensetzung unterschiedlicher Diensttypen im Dienstplan gestellt (*Dienstmix*). Bei der Dienstplanung wird als Ziel eine Minimierung der Kosten des Dienstplans (bestimmt durch Anzahl der Dienste, Anzahl der Überstunden, Verhältnis zwischen unproduktiver und produktiver Zeit je Fahrer, Diensttypen, Bezahlung usw.) angestrebt. Analog zu der Umlaufplanung unterscheidet man bei der Dienstplanung zwischen regelbasiertem und zielbasiertem Vorgehen.

Die Planungsphase ***Dienstreihenfolgeplanung*** (auch *Personaleinsatzplanung*

oder *Turnusplanung*) schließt den operativen Planungsprozess ab. In diesem Schritt werden mehrere Tagesdienste unter Berücksichtigung von Mindestruhezeiten, durchschnittlicher Wochenarbeitszeit und vorgegebener Turnusfolge von Arbeitstagen und arbeitsfreien Tagen zu Wochendienstplänen zusammengefasst. Danach erfolgt schließlich die Zuweisung der Dienstpläne zu den einzelnen Fahrern.

Verkehrsunternehmen müssen also im Rahmen des Planungsprozesses genaue Routenverläufe ihrer Fahrzeuge sowie deren Fahrpläne festlegen und die Arbeitspläne für die benötigten Fahrer erstellen. Der Planer steht einer schwierigen Aufgabe gegenüber: Die von ihm erstellten Pläne müssen in sich schlüssig, also fahrbar sein, und dabei einen möglichst effizienten Ressourceneinsatz ermöglichen. Dabei dürfen jedoch nicht nur die Kostenaspekte eine Rolle spielen, da es viele betriebliche, gesetzliche und tarifliche Rahmenbedingungen gibt. Das sind unter anderem Arbeitszeitvorschriften für Fahrer, technische Bedingungen (beispielsweise Wartungsintervalle oder Geschwindigkeit von Transportmitteln) oder Bedienung vorgeschriebener unrentabler Verbindungen.

Ein weiteres Problem entsteht bei der Wahl der Zielsetzung für die einzelnen Planungsschritte. Selbst wenn man sich auf die eindimensionale, monetäre Zielgröße beschränkt (z.B. Gesamtbetriebsergebnis in €), so ist zunächst unklar, wie dieses Gesamtziel auf den einzelnen Planungsstufen umgesetzt werden kann. Die Angebotsplanung setzt mit ihren monetär schwer fassbaren Zielen den Rahmen für die nachgeordnete Produktionsplanung in der Planungshierarchie. Darüber hinaus kann eine einzelne Zielsetzung auf einer hierarchisch höheren Planungsstufe den Zielen einer nachfolgenden Stufe entgegenstehen: So kann ein guter Fahrplan mit sehr niedrigen Umsteigewartezeiten einige Planungsphasen tiefer ein zusätzliches Fahrzeug samt Fahrer erfordern; oder die offensichtlich sinnvolle Minimierung der Fahrzeuganzahl bei der Umlaufplanung kann u. U. zu einer erhöhten Anzahl von Diensten führen. Besonders der letzte Punkt ist problematisch: Die mit dem Einsatz von Personal befassten Planungsstufen (Dienst- und Dienstreihenfolgeplanung) stehen in der Hierarchie ganz unten, obwohl die Personalkosten doch oft die ausschlaggebende Rolle spielen.

Die tatsächliche Arbeitsreihenfolge der vorgestellten Planungsaufgaben ist allerdings nicht immer streng sequenziell, sondern sie bewegt sich häufig in Schleifen: So wird u. U. der einmal bestimmte Fahrplan noch einmal geändert, wenn sich herausstellt, dass unwirtschaftliche Fahrzeugumläufe entstehen. Eine andere Möglichkeit ist vorausschauend zu arbeiten und z.B. schon bei der Linien- und Fahrplanung einige Aspekte der späteren Umlauf- und Dienstplanung zu berücksichtigen, etwa indem bereits Mindestwendezeiten und Auswirkungen von Arbeitszeitregelung (z.B. Pausenabwicklung) beachtet werden. Derartige Rückkopplungen und Vorausschau erfordern den Einsatz von erfahrenen Planern. Die Vorgehensweise ist nur

schwer formalisierbar und kann daher kaum in Planungsalgorithmen nachgebildet werden.

In weiteren Verlauf des Kapitels werden die beiden Probleme der Umlauf- und Dienstplanung sowohl einzeln als auch in einem integrierten Kontext ausführlich diskutiert.

2.2 Umlaufplanung

In diesem Abschnitt wird die Umlaufplanungsaufgabe mit ihren Zielen und Rahmenbedingungen näher betrachtet. Zunächst werden die zentralen Begriffe eingeführt, die im Rahmen dieser Arbeit in Bezug auf die Umlaufbildung von Bedeutung sind.

Unter einem *Betriebshof* wird ein Ort für das Abstellen der Fahrzeuge, die zu einem bestimmten Zeitpunkt nicht im Einsatz sind, verstanden. Das bedeutet, dass die Fahrzeuge nicht nur über Nacht, sondern auch vorübergehend dort abgestellt werden, da sämtliche Pausen im Betriebshof, im Gegensatz zu Pausen an Haltestellen, in der Regel keine oder geringere Personalkosten verursachen und deswegen nach Möglichkeit vorzuziehen sind. Steht in einem Betriebshof nur eine begrenzte Anzahl von Stellplätzen zur Verfügung, spricht man von einer *Betriebshofkapazität*.

Unter einer *Fahrgastfahrt* oder *Servicefahrt (engl.: trip)* wird eine durch den Fahrplan vorgegebene Fahrt zur Personenbeförderung verstanden. Dabei entspricht sie einer Fahrt zwischen zwei Endhaltestellen einer Linie. Jede Fahrgastfahrt ist durch eine Anfangs- und eine Endhaltestelle sowie eine Start- und eine Endzeit gekennzeichnet. Die Anfangs- und Endstation können dabei gleich sein, wenn die entsprechende Linie einen zyklischen Verlauf besitzt. Die Menge aller zu verplanenden Fahrgastfahrten aus dem Fahrplan wird als *Fahrplanmasse* bezeichnet.

Die eingesetzten Fahrzeuge können unterschiedliche Eigenschaften aufweisen und lassen sich nach *Fahrzeugtypen (engl.: vehicle type)* spezifizieren. Sie können sich in ihrer Größe, Geschwindigkeit oder Ausstattung unterscheiden. Diese Unterschiede müssen bei der Konstruktion des Umlaufplans berücksichtigt werden. So kann für jede Fahrgastfahrt angegeben werden, von Fahrzeugen welches Types sie bedient werden kann. Hierbei müssen unter anderem die Beförderungskapazitäten, spezielle Anforderungen (z.B. Niederflurbusse auf den Strecken in der Nähe von Krankenhäusern oder Altersheimen), aber auch technische Gegebenheiten der Fahrzeuge (z.B. keine Gelenkbusse bei Fahrten durch schmale Straßen, oder keine Doppeldecker bei Fahrten mit niedrigen Brücken) genau beachtet werden.

Unter dem Begriff *Depot* verstehen wir einen Betriebshof in Kombination mit einem Fahrzeugtyp. Zwei Fahrgastfahrten heißen *kompatibel*, wenn sie von einem

Fahrzeug nacheinander ausgeführt werden können. Falls die Endstation der ersten Fahrt ungleich der Startstation der darauffolgenden Fahrt ist, ist eine entsprechende *Verbindungsfahrt* (ohne Personenbeförderung) notwendig. Haben die beiden Fahrgastfahrten eine gemeinsame Anschlusshaltestelle, dann können sie direkt miteinander verknüpft werden. Ist die Wartezeit zwischen dem Ende der ersten und dem Anfang der darauffolgenden Fahrgastfahrt groß genug, ist es aus Kostengründen unter Umständen vorteilhafter, das Fahrzeug (ohne Fahrgäste) zwischenzeitlich ins Depot und später wieder zurück zu schicken, anstatt es an einer Haltestelle warten zu lassen. Eine solche indirekte Verbindung zweier kompatiblen Fahrgastfahrten bezeichnen wir als eine *Verbindungsfahrt über ein Depot*. Eine Fahrt, die vom Depot zur Starthaltestelle einer Fahrgastfahrt führt, heißt *Ausrückfahrt*. Analog verbindet eine *Einrückfahrt* den Endhaltepunkt einer Fahrgastfahrt mit dem Depot. Ein- und Ausrückfahrten werden auch als *Depotfahrten* bezeichnet. Neben den möglichen Depotfahrten während des Arbeitstages beginnt ein Fahrzeug seinen Arbeitstag typischerweise mit einer Ausrückfahrt und beendet ihn mit einer Einrückfahrt. Verbindungs- und Depotfahrten werden auch unter dem Begriff *Leerfahrten (engl.: deadheads)* zusammengefasst. Der Einsatz von Leerfahrten verursacht zwar zusätzliche operative Kosten, kann aber die Anzahl benötigter Fahrzeuge wesentlich reduzierten.

Unter einem *(Fahrzeug-)Umlauf (engl.: vehicle block)* wird die Fahrtroute eines Fahrzeuges für einen Betriebstag verstanden. Damit stellt ein Umlauf eine Folge von Fahrzeugaktivitäten wie Fahrgastfahrten, Leerfahrten und Wartezeiten dar, die im Laufe des Arbeitstages nacheinander von einem Fahrzeug ausgeführt werden. Die Umläufe beginnen und enden immer in einem Depot. Der Teil eines Umlaufs zwischen zwei Depotaufenthalten wird als ein *Umlaufstück* bezeichnet. Die Gesamtheit der Fahrzeugumläufe wird in einem *Umlaufplan (engl.: vehicle schedule)* festgehalten.

Bei dem *Umlaufplanungsproblem (engl.: Vehicle Scheduling Problem, VSP)* wird eine kostenminimale Zuordnung von Fahrgastfahrten zu Fahrzeugumläufen gesucht, sodass:

- jede Fahrgastfahrt in genau einem Fahrzeugumlauf enthalten ist,
- alle Fahrten eines Fahrzeugumlaufs von einem Fahrzeug nacheinander ausführbar und kompatibel sind,
- alle Fahrten eines Fahrzeugumlaufs von dem entsprechenden Fahrzeugtyp fahrbar sind und
- jedes Fahrzeug zu dem Depot zurückkehrt, von dem es zum Einsatz ausgerückt ist.

Dabei sind unterschiedliche Zielsetzungen möglich. In der Praxis wird oft eine Minimierung der Fahrzeuganzahl vorrangig angestrebt, da dabei der hohe Kapitalbedarf für die Beschaffung neuer bzw. die Instandhaltung vorhandener Fahrzeuge minimiert wird. Die (anteiligen) Anschaffungskosten für eingesetzte Fahrzeuge bilden den Block der *fixen* Kosten. Sie sind unabhängig von der eigentlichen Fahrleistung. An zweiter Stelle werden die operativen Kosten zur Ausführung des Umlaufplans minimiert. Diese von der Einsatzzeit und der Kilometerleistung abhängigen Kostenkomponenten werden als *variable* Kosten bezeichnet.

Umlaufplanungsprobleme werden grundsätzlich in zwei Klassen unterteilt. Handelt es sich nur um einen Betriebshof und einen homogenen Fuhrpark, spricht man von einem *Eindepot-Umlaufplanungsproblem* (engl.: *Single-Depot Vehicle Scheduling Problem, SDVSP*). Existieren mehrere Betriebshöfe bzw. werden unterschiedliche Fahrzeugtypen eingesetzt, dann gehört so ein Problem zur Klasse der *Mehrdepot-Umlaufplanungsprobleme* (engl.: *Multiple-Depot Vehicle Scheduling Problem, MDVSP*). Dabei wird jedes Fahrzeug genau einem Depot zugeordnet und darf nur die Fahrgastfahrten bedienen, für die das Depot und der entsprechende Fahrzeugtyp zulässig sind.

Das SDVSP ist in polynomieller Zeit lösbar, während die Mehrdepot-Variante mit mehr als zwei Depots ein $\mathcal{NP}$-schweres Problem darstellt (siehe [Bertossi et al., 1987]). Außerdem zeigte Löbel [Löbel, 1997], dass sogar eine ϵ-Approximation von MDVSP $\mathcal{NP}$-schwer ist. Die Komplexität des Problems hängt im Wesentlichen von der Anzahl der Depots, der Anzahl der Fahrplanfahrten und der Anzahl möglicher Verbindungsfahrten ab. Zusätzlich können weitere Aspekte wie z.B. Depotkapazitäten berücksichtigt werden, was zu einer weiteren Erhöhung der Problemkomplexität führt.

Bei der Planung der Umläufe kann außerdem unterschieden werden, ob ein Fahrzeug im Laufe des Tages nur Fahrten einer Linie bedienen darf (*linienreine* Planung) oder von einer Linie zur anderen wechseln darf (*liniengemischte* Planung). Die Anforderung bezüglich der Linienreinheit variiert von Betrieb zu Betrieb. Während die Fahrer eines Verkehrsunternehmens beliebig viele Linien im Laufe ihrer Arbeitstage fahren dürfen, wird bei einem anderen Betrieb eine Anzahl von vier Linien pro Tag schon als eine echte Herausforderung angesehen und darf auf keinen Fall überschritten werden. Allgemein lässt sich sagen, dass die Umläufe aus Wirtschaftlichkeitsgründen liniengemischt geplant werden sollen, falls die Variante zugelassen ist.

2.3 Dienstplanung

In diesem Abschnitt wird das Dienstplanungsproblem mit seinen Begriffen, Zielen und Rahmenbedingungen näher betrachtet. Dabei wird zwischen der umlaufbasierten und fahrplanbasierten Dienstplanung unterschieden. Außerdem werden die gängigen Dienstregeln diskutiert.

2.3.1 Traditionelle (umlaufbasierte) Dienstplanung

Wie bereits beschrieben geht ein zuvor generierter Umlaufplan beim traditionellen Vorgehen als Eingabe in die Dienstplanung ein (siehe Abbildung 2.1), wo die eingeplanten Fahrzeuge mit Fahrpersonal zu besetzen sind. Außerdem wird für jeden Umlauf eine Menge der so genannten Ablösepunkte definiert. Ein *Ablösepunkt (engl.: relief point)* ist eine durch die Uhrzeit (Ablösezeit) und den Ort (Ablösehaltestelle) festgelegte Möglichkeit, einen Fahrerwechsel auf einem Umlauf durchzuführen. Grundsätzlich sind insbesondere jeder Umlaufbeginn und jedes Umlaufende ein Ablösepunkt.

Die Umläufe werden an den zuvor definierten Ablösepunkten in elementare Arbeitseinheiten, die so genannten Dienstelemente, zerlegt. Ein *Dienstelement (engl.: task)* ist eine nicht mehr zu unterteilende Arbeitsaufgabe und besteht aus einer Folge von Fahrgast- und Leerfahrten zwischen zwei aufeinanderfolgenden Ablösepunkten in einem Umlauf. Die Menge aller Dienstelemente wird als *Dienstplanmasse* bezeichnet und gilt als Grundlage für die Dienstplanung. Zusätzlich zu den Dienstelementen, die sich aus dem Umlaufplan ergeben, können innerhalb der Dienste noch Zusatzarbeiten notwendig sein, wie z.B. Fahrzeugkontrolle vor dem Einsatzbegin, Abrechnung am Ende des Tages, Transfer zwischen Ablösepunkten usw. Solche a priori nicht festgelegten Aktivitäten, die sich erst aus der Planung ergeben, werden *Ergänzungselemente* genannt.

Eine Sequenz von aufeinanderfolgenden Dienstelementen eines Umlaufs, die von einem Fahrer ununterbrochen (ohne gesetzlich vorgeschriebe Pause) ausgeführt werden, wird als ein *Dienststück (engl. piece of work)* bezeichnet. Ein *(Tages-)Dienst (engl.: duty)* besteht aus einem oder mehreren solcher Dienststücke mit gesetzlich vorgeschriebenen Pausen dazwischen. Abbildung 2.2 veranschaulicht ausgewählte Begriffe anhand eines Beispiels. Die Gesamtheit aller Dienste wird *Dienstplan (engl.: crew schedule)* genannt.

Ziel des *Dienstplanungsproblems (engl.: Crew Scheduling Problem, CSP)* ist es, aus einer vordefinierten Dienstplanmasse eine Menge von Tagesdiensten zu bilden, sodass

- jedes Dienstelement in genau einem Dienst enthalten ist,

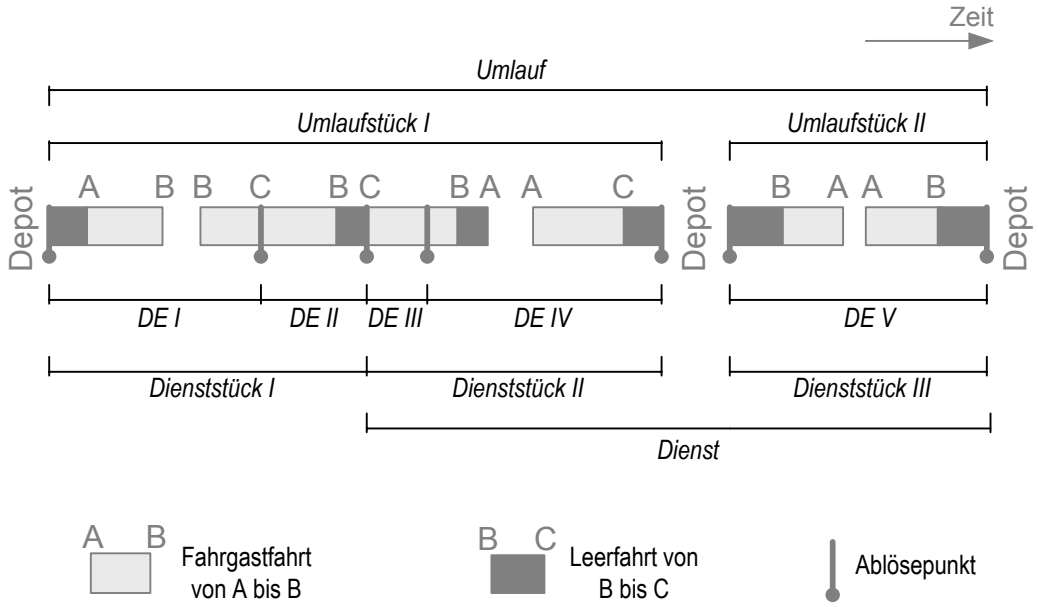

Abbildung 2.2: Übersicht der Begriffe bei der Umlauf- und Dienstplanung

- jeder Dienst bezüglich einer Reihe von gesetzlichen, tariflichen sowie technischen Vorschriften und Dienstregeln gültig ist und
- die durch den Dienstplan geplanten Personalkosten minimiert werden.

Die Personalkosten können je nach Verkehrsbetrieb unterschiedlich definiert sein. Im Allgemeinen sind sie die Summe der Kosten einzelner Dienste. Die Kosten eines einzelnen Dienstes setzen sich üblicherweise aus einer *variablen* Komponente, die von der Arbeitszeit abhängt, einem *fixen* Anteil, der von dem zugrunde liegenden Typ des Dienstes abhängt, sowie aus den tariflich geregelten Zuschlägen (z.B. Nacht, Wochenende oder auch Abweichung von Zielwerten) zusammen. Ziele müssen nicht notwendigerweise eine unmittelbare monetäre Interpretation haben. Man kann auch soziale und betriebliche Kriterien, wie Arbeitszeit, Wirkungsgrad, Attraktivität für Fahrer oder Abweichungen vom angestrebten Dienstschnitt miteinbeziehen, oder auch Kombinationen davon.

Jeder Dienst muss hinsichtlich einer Reihe von gesetzlichen, tariflichen, technischen, betrieblichen Bedingungen zulässig sein. Die Vielfalt aller einzuhaltenden *lokalen* Dienstregeln (wie z.B. minimale und maximale Dienstlänge, Arbeitszeit, ununterbrochene Lenkzeit, Pausendauer) ist sehr groß und macht das Problem der Dienstplanung besonders komplex. Dazu kommen oft noch *globale* Dienstregeln, die den gewünschten Dienstmix festlegen. Vorgaben dieser Art zielen einerseits auf die Sozialverträglichkeit der Dienstplanung, andererseits auf eine Integration der Dienstplanung und der nachfolgenden Dienstreihenfolgeplanung. Der Dienstmix ist ein typisches *„weiches"* Nebenkriterium, das eine grobe Abweichung vom an-

gestrebten Zielwert nicht zulässt, jedoch einen (vom Anwender zu bestimmenden) Spielraum zugunsten eines Hauptplanungsziels durchaus erlaubt.

Ein weiterer Aspekt der Dienstplanung ist die Anzahl der Depots. Existieren mehr als ein Depot, wird in der Praxis oft verlangt, dass jeder Dienst, ähnlich zu Umläufen, genau einem Depot zugeordnet wird. Dabei darf ein Dienst nur aus Dienstelementen der Umläufe bestehen, die demselben Depot zugeordnet sind. Praktisch heißt das, dass die Fahrer nur Fahrzeuge aus dem „eigenen" Depot bedienen dürfen. Ist das der Fall, zerfällt die gesamte umlaufbasierte Dienstplanung in separate Dienstplanungsprobleme für jedes Depot. Dabei werden nur die für das entsprechende Depot relevanten Umläufe als Eingabe betrachtet. Bei Problemstellungen, bei denen die Fahrer auch „depotfremde" Fahrzeuge bedienen dürfen oder ein depotübergreifender Dienstmix vorgegeben ist, wird die Dienstplanung über mehrere Depots in einem Gesamtproblem durchgeführt.

2.3.2 Fahrplanbasierte Dienstplanung

Die *fahrplanbasierte Dienstplanung* unterscheidet sich von der umlaufbasierten, indem sie unabhängig von der Umlaufplanung stattfindet, d. h. sie basiert nicht auf den a priori berechneten Umläufen, sondern direkt auf der Fahrplanmasse. Sämtliche Fahrzeugaktivitäten werden komplett ignoriert bzw. sind gar nicht bekannt. Somit ergeben sich für die Konstruktion der Dienststücke bzw. Dienste viel mehr Freiheitsgrade, da sie nicht mehr entlang der Umläufe gebildet werden. Die Möglichkeit, zwei Dienstelemente zu einem Dienststück aneinanderzureihen, ist nur durch zeitliche und räumliche Gegebenheiten begrenzt.

Bei einem *fahrplanbasierten Dienstplanungsproblem (engl.: Independent Crew Scheduling Problem, ICSP*[3]*)* wird eine kostenminimale Menge von Diensten gesucht, sodass jedes von Fahrgastfahrten direkt abgeleitete Dienstelement in genau einem Dienst enthalten ist und alle Dienste bezüglich der vorgegebenen Dienstregeln gültig sind.

Die Idee, Dienste direkt auf der Fahrplanmasse anstatt auf den Fahrzeugumläufen zu planen, wird in einigen partiell integrierten Lösungsansätzen zur Umlauf- und Dienstplanung verwendet. Bei diesen Ansätzen werden die beiden Planungsphasen zwar sequenziell, aber in umgekehrter Reihenfolge ausgeführt, d.h. zuerst Dienst-

[3]In englischsprachiger Literatur wird das Dienstplanungsproblem auf Basis von Umläufen oft als das (traditionelle) *sequenzielle* Dienstplanungsproblem bezeichnet, während die umlaufunabhängige Variante als das *unabhängige* Dienstplanungsproblem genannt wird. Wir finden dafür jedoch die Begriffe *umlaufbasiertes* bzw. *fahrplanbasiertes* Dienstplanungsproblem aussagekräftiger und benutzen sie als das deutsche Equivalent. Bei der Benutzung von Abkürzungen verwenden wir allerdings weiterhin die englischen CSP bzw. ICSP, um die entsprechenden Problemklassen mit denen aus der englischsprachigen Literatur in Relation zu setzen.

und danach Umlaufplanung. Diese Vorgehensweise, auch als *Crew-First–Vehicle-Second* bekannt, wird mit der Tatsache begründet, dass operative Personalkosten bei manchen Verkehrsbetrieben deutlich höher als operative Fahrzeugkosten sind (siehe [Bodin et al., 1983], [Leuthardt, 1998], [Freling et al., 2001]).

Ein Nachteil von Crew-First–Vehicle-Second-Methoden liegt darin, dass es oft sehr problematisch ist, für eine gegebene Menge an Diensten einen gültigen Umlaufplan zu bestimmen. Für bestimmte Problemstellungen lässt sich das Dienstplanungsproblem durch zusätzliche Anforderungen erweitern, sodass ein gültiger Umlaufplan im Nachhinein garantiert werden kann. Eine solche Möglichkeit ist zum Beispiel nur ein einziges Depot als Ablösepunkt zu akzeptieren. Dann startet und endet jedes Dienststück nur im Depot und entspricht gleichzeitig einem Umlaufstück. Aus den Dienststücken der resultierenden Dienste lassen sich im Nachhinein gültige Umläufe zusammenbauen. Waren allerdings ursprünglich auch andere Ablösepunkte als das Depot erlaubt, wird der Lösungsraum durch eine solche Vereinfachung u. U. stark eingeschränkt, sodass sowohl der Dienst- als auch der Umlaufplan oft schlechter als bei der traditionellen Vehicle-First–Crew-Second-Vorgehensweise sind.

Noch problematischer wird das Finden guter, zueinander kompatibler Umlauf- und Dienstpläne, sobald mehrere Depots berücksichtigt werden. Da jeder Dienst genauso wie jeder Umlauf zu einem einzigen Depot zugeordnet wird, kann der darauf folgende Umlaufplan von sehr schlechter Qualität oder oft gar unzulässig werden, besonders wenn die Anzahl verfügbarer Fahrzeuge pro Depot begrenzt ist.

Eine weitere Einsatzmöglichkeit von ICSP ist die Berechnung einer unteren Schranke für die Dienstplanung (ohne eine nachträgliche Bestimmung der Umläufe). Freling [Freling, 1997, Freling et al., 1999] vergleicht diese Schranke mit der Lösung der traditionellen (sequenziellen) Planung von Umläufen und Diensten und benutzt die Differenz als Maß für eine potenzielle Einsparung durch die Integration. Unterscheiden sich die beiden Werte für eine Probleminstanz nur gering, dann ist der Einsatz von einem viel aufwendigeren und zeitintensiveren integrierten Verfahren für sie nicht lohnenswert.

2.3.3 Dienstregeln

Im Folgenden werden weitere, zum Verständnis dieser Arbeit notwendige Begriffe eingeführt und die gängigen Dienstarten definiert. Sämtliche Definitionen und Kennzahlen basieren auf [Fachwort, 1992] und einer Spezifikation, die im Rahmen eines Projektes am DS&OR-Lab (Uni Paderborn) mit der PTV AG entstanden ist. Sie stellen nur einen Richtwert dar und können sich von Betrieb zu Betrieb unterscheiden.

In der Dienstplanung werden jedem Dienst verschiedene *Attribute* zugeordnet. Wir unterscheiden zwischen Zeitattributen, wie Arbeits-, Lenk- und Pausenzeit, Dienstbeginn und -Ende, Dienstlänge und anderen Attributen wie Anzahl der Dienststücke. Diesen Kennzahlen können minimale (ggf. früheste), maximale (ggf. späteste), durchschnittliche und angestrebte Werte zugeordnet sein.

Arbeitszeit (engl.: working time) ist die vergütete Zeit eines Dienstes. Die Arbeitszeit muss den vom Gesetzgeber und dem Unternehmen vorgegebenen Bedingungen genügen. Folgende Zeiten zählen üblicherweise zu Arbeitszeit:

- *Lenkzeit (engl.: driving time).* Die Lenkzeit umfasst die reine Fahr- bzw. Kurbelzeit eines Fahrers auf einem Fahrzeug während einer Arbeitsschicht. Die maximale ununterbrochene Lenkzeit darf typischerweise 4:30 Stunden nicht übersteigen. Als *Lenkzeitunterbrechung* zählen gesetzlich vorgeschriebene Pausen, anrechenbare Wendezeit sowie Anwesenheiten, die in einem fahrenden Fahrzeug verbracht werden, ohne dass gelenkt wird (ab einer bestimmten Länge).

- *Wendezeit (engl.: layover time).* Die Wendezeit umfasst die Standzeit zwischen zwei Fahrgastfahrten. Man unterscheidet zwischen *anrechenbarer Wendezeit (AWZ)* und *nicht anrechenbarer Wendezeit (NWZ).* Die anrechenbare Wendezeit wird für Pausen angerechnet und zählt somit (evtl. bis zu einer bestimmten Grenze) nicht zur Arbeitszeit. Sie muss eine je nach zugrundeliegender Pausenregelung vorgegebene minimale Dauer erfüllen. Außerdem kann nur der Teil als pausenvertretbar angerechnet werden, der regelmäßig nicht für dienstliche Verrichtungen, wie z.B. Umsetzen, Beschilderung, Fahrzeugkontrolle, Auskunftserteilung, Fahrgastbedienung, Verspätungsausgleich, benutzt wird.

- *Vorbereitungszeit (engl.: sign-on time).* Als Vorbereitungszeit gilt die Zeit, die einem Fahrer zu Beginn eines Dienstes (ggf. Dienststückes) gewährt wird, z. B. um sich über den Dienst zu informieren oder das Fahrzeug zu kontrollieren.

- *Abschlusszeit oder Nachbereitungszeit (engl.: sign-off time).* Die Abschlusszeit (Nachbereitungszeit) umfasst die Zeit, die einem Fahrer am Ende eines Dienstes (ggf. Dienststückes) gewährt wird, z. B. um abzurechnen oder das Fahrzeug abzurüsten.

- *Transferzeit oder Wegezeit (engl.: walking time).* Zur Transferzeit (Wegezeit) zählt die Zeit, die ein Fahrer benötigt, um von einem Ablöseort zum nächsten zu gelangen.

Als *Dienstbeginn (engl.: duty start time)* wird der Beginn des ersten Dienststückes eines Dienstes abzüglich einer ggf. anzurechnenden Vorbereitungszeit und

abzüglich einer ggf. anzurechenden Wegezeit bezeichnet. Als *Dienstende (engl.: duty end time)* wird das Ende des letzten Dienststückes eines Dienstes zuzüglich einer ggf. anzurechnenden Abschlusszeit und zuzüglich einer ggf. anzurechnenden Wegezeit bezeichnet. Die *Dienstlänge (engl.: duty length/spread time)* ergibt sich aus der Differenz zwischen dem Dienstende und Dienstbeginn.

Neben den oben beschriebenen Attributen wird jedem Dienst ein *Diensttyp* zugeordnet. Je nach Dienstbeginn und -Ende unterscheidet man zwischen *Früh-, Mittel-, Spät-* und *Nachtdiensten*. Dienste müssen nicht zusammenhängend sein. Zur Abdeckung von Hauptverkehrszeiten sind auch *geteilte Dienste (engl.: split duty)* möglich, die typischerweise aus zwei Arbeitseinheiten mit einer längeren Pause dazwischen bestehen. Eine *Teilungsgrenze* ist eine zeitliche Angabe zur minimalen Unterbrechungslänge, ab welcher ein geplanter Dienst als geteilt anzusehen ist. Außerdem sind Dienste möglich, die nur ein Dienststück und keine Pause beinhalten. Sie werden *Teildienste (engl.: tripper)* genannt.

Die gesetzliche Lenkzeitverordnung legt fest, wann und wie lange ein Fahrer eine Pause machen muss und wann eine Arbeitsunterbrechung als Pause anerkannt wird. In Deutschland unterscheidet man grundsätzlich zwischen fünf Pausenregelungen, die sich in zwei Klassen, *Blockpausen-* und *Verhältnispausenregelungen*, aufteilen lassen. Zu den Blockpausenregelungen gehören:

- *Block30-Regel* - der Dienst wird durch 1 Pause (oder ausreichend große AWZ) mit einer Mindestlänge von 30 Minuten unterbrochen,
- *Block20-Regel* - der Dienst wird durch 2 Pausen (oder ausreichend große AWZ) mit einer Mindestlänge von 20 Minuten unterbrochen,
- *Block15-Regel* - der Dienst wird durch 3 Pausen (oder ausreichend große AWZ) mit einer Mindestlänge von 15 Minuten unterbrochen.

Dabei beträgt die maximale ununterbrochene Lenkzeit zwischen zwei Pausen 4:30 Stunden.

Bei Verhältnispausen muss das Verhältnis von Lenkzeit zu Arbeitszeit in einem bestimmten Verhältnis stehen. Dabei können nur die Wendezeiten einer bestimmten Länge angerechnet werden. Zu den Verhältnispausenregelungen zählen:

- *1/5-Regel (Fünftelregel)* - die Summe aller AWZ, die mindestens 8 Minuten lang sind, ist mindestens 1/5 der Summe aller Lenkzeiten des Dienstes
- *1/6-Regel (Sechstelregel)* - die Summe aller AWZ, die mindestens 10 Minuten lang sind, ist mindestens 1/6 der Summe aller Lenkzeiten des Dienstes

Alle Dienste müssen den gesetzlich vorgegebenen Pausenregelungen entsprechen, d.h. sie müssen mindestens eine der fünf Pausenregelungen erfüllen.

Durch eine konkrete Festlegung der bisher genannten Parameter (also Diensttyp, Pausenregelung, Attribute) wird eine *Dienstart (engl.: duty type)* festgelegt, z.B. ein zusammenhängender Frühdienst mit Sechstelregel, Dienstbeginn zwischen 6:00 und 10:00 Uhr, Dienstende zwischen 13:00 und 22:00 Uhr, maximale Dienstlänge 9 Stunden, maximale Arbeitszeit 8 Stunden.

2.4 Integrierte Umlauf- und Dienstplanung

Wie zum Beginn des Kapitels bereits erwähnt, würde eine gesamtoptimale, den wechselseitigen Abhängigkeiten gerechte ÖPNV-Planung unter Beachtung aller Nebenbedingungen eine simultane Betrachtung aller Teilaufgaben des Planungsprozesses erfordern. Allerdings stößt die Aufgabe schon wegen der Komplexität der einzelnen Teilprobleme auf kaum zu überwindende Schwierigkeiten. Daher verfolgen die meisten kommerziellen, auf mathematischer Optimierung basierenden Planungstools eine streng sequenzielle Abarbeitung der einzelnen Teilaufgaben. Nichtsdestotrotz geht die aktuelle Entwicklung auf diesem Gebiet in Richtung einer simultanen Betrachtung mehrerer Teilprobleme. Eine Möglichkeit ist eine synchrone Verplanung der verfügbaren Ressourcen, d.h. der Fahrzeuge und Fahrer. In diesem Fall sprechen wir von integrierter Umlauf- und Dienstplanung.

Die wesentlichen Unterschiede zwischen integrierter und sequenzieller Umlauf- und Dienstplanung finden sich in den zusätzlichen Freiheitsgraden im Bereich der Dienstplanung. Während bei der sequenziellen Planung die Leerfahrten eines Umlaufplans der nachfolgenden Dienstplanung fest vorgegeben sind, stehen in der integrierten Planung alle möglichen Fahrtverknüpfungen als Freiheitsgrade bei der Konstruktion von Diensten zur Verfügung. In dieser planerischen Freiheit liegt das Potenzial, durch eine integrierte Planung im Vergleich zur sequenziellen Planung Kosten einzusparen. Hinzu kommt, dass die Personalkosten in der Regel die fahrzeugbezogenen Betriebskosten dominieren.

Zum besseren Verständnis der Vorteile einer simultanen Betrachtung beider Planungsaufgaben betrachten wir ein Beispiel:

Gegeben ist ein Fahrplan mit fünf Fahrgastfahrten F_1, F_2, F_3, F_4 und F_5 (siehe Abbildung 2.3 für Start- und Endzeiten sowie Start- und Endhaltestellen der Fahrten) und zwei Depots, wobei in jedem Depot jeweils ein Fahrzeug zur Verfügung steht. Bei der Umlaufplanung wird primär die Anzahl der Fahrzeuge und danach die Summe von Leerkilometer minimiert. Abbildung 2.3 veranschaulicht den optimalen Umlaufplan. Dabei bedient das Fahrzeug aus dem ersten Depot die Fahrten F_1, F_2, F_3 und F_4 und das Fahrzeug aus dem zweiten Depot die Fahrt F_5. Die die operativen Kosten verursachenden Leerfahrten sind im Bild dunkel schattiert.

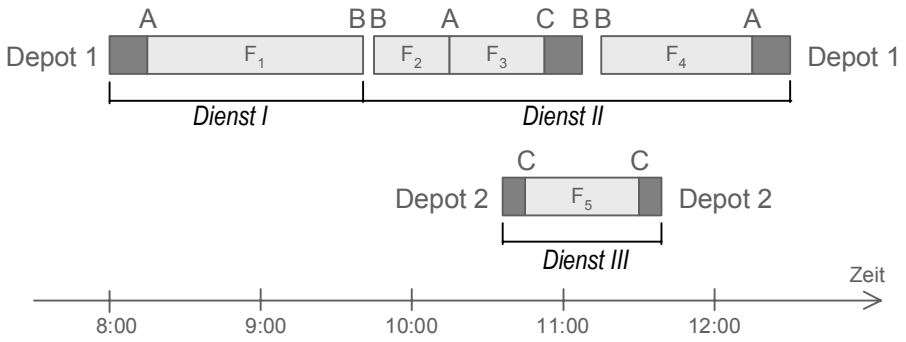

Abbildung 2.3: Optimale Lösung der sequenziellen Umlauf- und Dienstplanung (2 Umläufe und 3 Dienste)

Für die anschließende Dienstplanung ist gegeben: Als Ablösepunkte gelten die beiden Depots und die Haltestelle B; die maximale Dienststücklänge ist auf 4 Stunden begrenzt; jeder Dienst darf nur Fahrten aus dem eigenen Depot beinhalten; minimiert wird die Anzahl der Dienste. Offensichtlich ist der Umlauf im ersten Depot zu lang, um von einem einzigen Dienst komplett bedient zu werden. Somit sind zwei Dienste für den ersten Umlauf und ein Dienst für den zweiten Umlauf notwendig (siehe Abbildung 2.3). Die Lösung der sequenziellen Planung beinhaltet also 2 Fahrzeuge und 3 Fahrer.

Es ist allerdings eine bessere, gesamtoptimale Lösung möglich, die mit nur 2 Fahrzeugen und 2 Fahrern auskommt (siehe Abbildung 2.4). Dafür werden aber mehr Leerkilometer verplant und man verzichtet somit auf die Optimalität des Umlaufplans. Diese Lösung ist nur durch eine gleichzeitige Betrachtung der Dienst- und Umlaufplanung möglich.

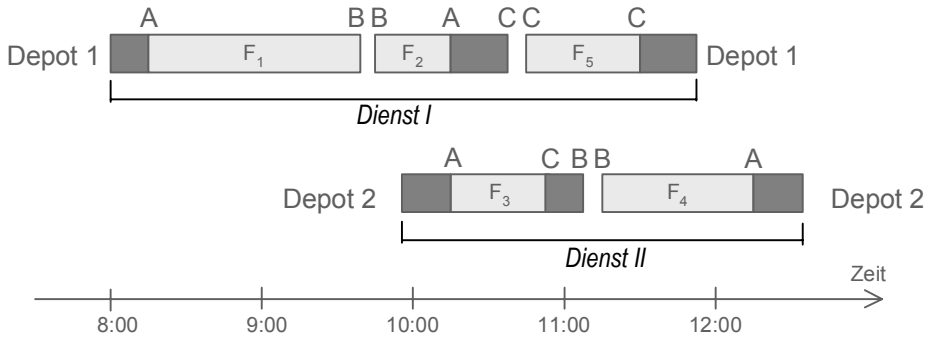

Abbildung 2.4: Gesamtoptimale Lösung der Umlauf- und Dienstplanung (2 Umläufe und 2 Dienste)

Das Beispiel zeigt, dass ein Umlaufplan, der für sich betrachtet zwar suboptimal ist, zu einer Verbesserung der Gesamtlösung führen kann. Dieser Sachverhalt ist

das erste wichtige Argument für die Verwendung von Ansätzen zur integrierten Umlauf- und Dienstplanung.

Ein weiteres Argument betrifft die besondere Planungssituation im *Nachbarsort-* und *Regionalverkehr*. Dort gibt es im Vergleich zum städtischen Nahverkehr nur wenige oder gar keine Ablösepunkte außerhalb der Depots. Außerdem sind die Ablöseorte oft weit voneinander entfernt, sodass kein Transfer (zu Fuß oder mit anderen Transportmitteln) zwischen ihnen möglich ist. Wird ein optimaler Umlaufplan zuerst konstruiert, kann er Umläufe beinhalten, die eine lange Zeit keinen Ablösepunkt passieren, sodass es zu unlösbaren Problemen bei der Gewährung von Pausen in der darauffolgenden Dienstplanung kommen kann. Ein zulässiger Dienstplan kann somit von schlechter Qualität sein oder gar nicht existieren. Eine gleichzeitige Betrachtung beider Planungsaufgaben würde das geschilderte Problem lösen.

Das *integrierte Umlauf- und Dienstplanungsproblem (engl.: Integrated Vehicle and Crew Scheduling Problem, VCSP)* kann wie folgt definiert werden: Gegeben sei eine Menge zu verplanenden Fahrgastfahrten innerhalb eines fest definierten Planungszeitraums. Gesucht wird ein zulässiger Umlauf- und ein dazu kompatibler, zulässiger Dienstplan mit minimalen Gesamtkosten. Die beiden Pläne sind dann *kompatibel*, wenn sie jede Fahrgastfahrt genau einmal und die Leerfahrten des Umlaufplans *konsistent* überdecken. Ein Fahrzeug muss also mit einem Fahrer besetzt sein, wenn es sich außerhalb eines Depots befindet. Für die Zulässigkeit der Umlauf- bzw. Dienstpläne gelten die üblichen Anforderungen (siehe Abschnitt 2.2 bzw. 2.3).

Man unterscheidet zwischen VCSP mit einem und mehreren Depots. Aus der Sichtweise der Problemkomplexität liegt dabei der Unterschied darin, wie effizient die zugrunde liegenden Unterprobleme VSP und CSP gelöst werden können. Während das Eindepot-Umlaufplanungsproblem noch in polynomieller Zeit lösbar ist, ist die Mehrdepot-Variante ein $\mathcal{NP}$-schweres Problem (siehe [Bertossi et al., 1987]). Der zweite Bestandteil von VCSP, das Dienstplanungsproblem, ist dagegen in beiden Fällen $\mathcal{NP}$-hart (siehe [Fischetti et al., 1987], [Fischetti et al., 1989]).

Kapitel 3

Mathematische Optimierung

Der Begriff **Optimierung** wird in [Grünert and Irnich, 2005, S.6] wie folgt beschrieben:

> *„Optimierung ist eine auf quantitativen mathematischen Modellen beruhende Technik zur Berechnung von Entscheidungsvorschlägen. Es handelt sich dabei um eine Methode, die von einem geschlossenen mathematischen Modell ausgeht und darauf aufbauend eine Lösung bestimmt. Entsprechende Lösungsverfahren, auch Algorithmen genannt, werden in der Regel nicht „von Hand gerechnet", sondern auf Computern in Programmen bzw. Softwaresystemen umgesetzt."*

Es hat sich historisch entwickelt, dass für diese Art von Optimierung oft der Begriff Mathematische Programmierung *(engl.: mathematical programming)* benutzt wird.

Für einen Abschnitt der Realität wird ein abstraktes Modell gebildet, mit dessen Hilfe Analysen durchgeführt werden können, um somit eine gute Basis für Entscheidungen zu schaffen. Die Entscheidungen werden durch *Variablen (Entscheidungsvariablen)* im Modell kodiert, wobei sie nur bestimmte Werte annehmen dürfen. Die zulässigen Wertebereiche für Variablen werden durch eine Menge von Gleichungen und Ungleichungen, die sogenannten *Restriktionen (Nebenbedingungen)*, beschrieben. Eine bestimmte Entscheidungssituation, auch *Lösung* genannt, wird dadurch definiert, dass den Variablen konkrete Werte zugewiesen werden. Eine Lösung ist zulässig, wenn alle Restriktionen erfüllt sind. Die Gesamtheit aller Lösungen (Systemzustände) wird als ein *Lösungsraum* bezeichnet. Die Qualität jeder Lösung wird mit Hilfe einer bzw. mehrerer Bewertungsfunktionen (*Zielfunktionen*) beurteilt. Das *Optimierungsproblem* ist somit die Aufgabe der Bestimmung einer zulässigen Lösung mit dem optimalen Zielfunktionswert. Wird dabei die Lösung mit maximalem (minimalem) Zielfunktionswert gesucht, spricht man von einem *Maximierungsproblem (Minimierungsproblem)*.

Laut der Komplexitätstheorie ist ein Problem $\mathcal{NP}$-schwer, wenn bis jetzt noch kein Algorithmus gefunden wurde, der dieses Problem in polynomieller Zeit lösen kann; aber es wurde auch nicht bewiesen, dass so ein Algorithmus nicht existiert. Die Komplexitätstheorie besagt, dass wenn für ein $\mathcal{NP}$-schweres Problem ein polynomieller Algorithmus existiert, dann existiert so ein Algorithmus für jedes $\mathcal{NP}$-schwere Problem. Viele Optimierungsprobleme gehören zur Klasse der $\mathcal{NP}$-schweren Probleme.

Im Folgenden stellen wir kurz ausgewählte Modellierungs- und Lösungstechniken von Optimierungsproblemen vor.

Lineare Optimierung
Bei Modellen der *linearen Optimierung* oder *linearen Programmierung (LP)* bestehen die Zielfunktion und alle Restriktionen aus Linearkombinationen der Entscheidungsvariablen. Die Variablen selbst können dabei reelle Werte annehmen. Eine Vielzahl verschiedener Probleme kann als lineare Programme modelliert werden. Die lineare Programmierung spielt eine wichtige Rolle in der mathematischen Optimierung. Zum Lösen auch riesiger linearer Programme existiert eine Reihe schneller, effizienter Algorithmen.

Gemischt-ganzzahlige Optimierung
Viele praktische Probleme sind nur mit diskreten Variablen sinnvoll, da eine Teilbarkeit von Ressourcen oft nicht gegeben ist oder die Variablen unteilbare 0/1-Entscheidungen abbilden. Dürfen alle bzw. einige der Variablen nur ganze Zahlen annehmen, dann spricht man von *ganzzahliger* bzw. *gemischt-ganzzahliger Optimierung (engl.: integer programming, IP; mixed integer programming, MIP)*. Probleme, die im Bereich der Transportplanung auftreten, werden meistens als gemischt-ganzzahlige Optimierungsprobleme formuliert. Im Gegensatz zur linearen Optimierung sind ganzzahlige und gemischt-ganzzahlige Modelle in der Regel sehr schwer zu lösen. Ein gängiges Verfahren zur Lösung von MIP-Problemen ist das *Branch-and-Bound*-Verfahren.

Heuristiken und Metaheuristiken
Viele praktisch relevante Optimierungsprobleme sind sehr groß bzw. sehr schwer zu lösen. Daher ist man oft gezwungen auf Verfahren zurückzugreifen, die mit einem vertretbaren Rechenaufwand eine möglichst gute, aber nicht zwingend optimale Lösung liefern. Solche Verfahren werden *Heuristiken* genannt. Heuristiken sind in der Regel auf ein spezielles Problem zugeschnitten und nutzen die Problemstruktur aus. *Metaheuristiken* sind dagegen allgemeine, übergeordnete Schemata zur Steuerung eines oder mehrerer abhängiger heuristischer Verfahren. Viele Metaheuristiken gehören zu den sogenannten naturanalogen Verfahren, weil sie die Natur nachbil-

den. Typische Metaheuristiken sind Simulated Annealing, genetische Algorithmen, neuronale Netze und Ameisenalgorithmen. Die meisten Metaheuristiken lassen es zu, dass man sich während der Suche in Bezug auf die Zielfunktion kurzfristig verschlechtern kann. Damit ist die Hoffnung verbunden, dass man aus einem lokalen Optimum wieder herauskommt. Bei Heuristiken und Metaheuristiken gibt es im Allgemeinen keine Abschätzung, wie weit eine gefundene Lösung vom Optimum entfernt ist.

Dynamische Programmierung
Bei der dynamischen Programmierung werden Probleme betrachtet, die in mehrere Stufen zerlegt werden können. Hängt dabei der Zustand einer Stufe von dem Zustand und der Entscheidung der vorhergehenden Stufe ab, spricht man von *dynamischer Optimierung*. Die Gesamtentscheidung wird dabei durch stufenweise, rekursive Einzelentscheidungen ersetzt.

Der Rest dieses Kapitels ist wie folgt aufgebaut: Nach einer kurzen Vorstellung einiger für das Verständnis dieser Arbeit relevanter Optimierungsprobleme werden ausgewählte Lösungstechniken diskutiert, die in entwickelten Lösungsverfahren eingesetzt werden.

3.1 Ausgewählte Probleme der mathematischen Optimierung

In diesem Abschnitt werden ausgewählte Standardprobleme der mathematischen Optimierung vorgestellt, die im Rahmen dieser Arbeit verwendet werden.

3.1.1 Netzwerkflussprobleme

Netzwerkflussprobleme vereinigen Modelle und Methoden der Optimierung mit der Graphentheorie. Ein *Netzwerk* ist ein zyklenfreier Digraph mit genau einer Quelle und genau einer Senke.

Das Minimalkostenfluss-Problem
Als eines der Grundmodelle der Netzwerktheorie kann das *Minimalkostenfluss-Problem (engl.: min-cost flow problem)* aufgefasst werden. Viele andere Netzwerkflussprobleme sind Spezialfälle dieses Problems. Das Minimalkostenfluss-Problem wird häufig auch als *Umladeproblem (engl.: transshipment problem)* bezeichnen und lässt sich wie folgt beschreiben: Bestimme einen kostenminimalen Fluss eines Gutes in einem Netzwerk, sodass der Bedarf nach diesem Gut in jedem Knoten befriedigt

wird.

Hierbei liegt ein gerichteter Graph $G = (N, A)$ zugrunde, bei dem N die Menge der Knoten und A die Menge der gerichteten Kanten ist. Für jede Kante $(i, j) \in A$ sind Kosten c_{ij} definiert, die für jede Einheit des Gutes anfallen, wenn es über (i, j) gesendet wird. Die minimale bzw. maximale Kapazität einer Kante wird als l_{ij} bzw. u_{ij} definiert. Jeder Knoten $i \in N$ hat einen Bedarf b_i. Ein negativer Bedarf entspricht einer Nachfrage, ein positiver einem Angebot. Ist der Bedarf Null, so handelt es sich um einen Umladeknoten. Für jede Kante $(i, j) \in A$ gibt die Flussvariable $x_{ij} \in \mathbb{R}$ die Flussgröße auf dieser Kante an. Das mathematische Modell kann wie folgt formuliert werden:

$$\min \quad \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{3.1}$$

$$\text{s.t.} \quad \sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = b_i \qquad \forall i \in N \tag{3.2}$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \qquad \forall (i, j) \in A \tag{3.3}$$

In der Zielfunktion (3.1) werden die Kosten des Gesamtflusses minimiert. Die *Flusserhaltungsbedingungen (engl.: flow balance constraints)* (3.2) stellen für jeden Knoten sicher, dass die Differenz zwischen dem ankommenden und dem ausgehenden Fluss immer gleich der Nachfrage bzw. dem Bedarf dieses Knoten ist. Schließlich fordern die *Kapazitätsbedingungen (engl.: flow bound constraints)* (3.3), dass der Fluss jeweils zwischen der unteren und oberen Schranke liegen muss.

Sind alle Bedarfe und Schranken ganzzahlig, dann ist auch jede Lösung von (3.1)-(3.3) ganzzahlig, obwohl $x_{ij} \in \mathbb{R}$ für alle $(i, j) \in A$ ist (siehe z.B. [Wolsey, 1998]). Das liegt an der *unimodularen* Struktur der Inzidenzmatrix, sie ist total . Somit kann das Minimalkostenfluss-Problem mit effizienten Algorithmen der linearen Programmierung, wie z.B. der Simplex- oder Netzwerk-Simplex-Methode in polynomieller Zeit gelöst werden.

Durch Spezialisierung des Minimalkostenfluss-Problems erhält man eine Reihe anderer bekannter Probleme. Zwei davon, die für die vorliegende Arbeit relevant sind, sind das Zuordnungsproblem und das Kürzeste-Wege-Problem.

Das Zuordnungsproblem

Bei einem *(linearen) Zuordnungsproblem (engl.: linear assignment problem)* wird eine kostenminimale 1-zu-1–Zuordnung von Objekten aus einer Menge N_1 zu Objekten aus einer gleichgroßen Menge N_2 gesucht, wobei jede mögliche Zuordnung mit Kosten bewertet wird.

Das lineare Zuordnungsproblem kann als ein Minimalkostenfluss-Problem auf einem bipartiten Graphen $G = (N_1 \cup N_2, A)$ formuliert werden. Dabei bildet A die

Menge aller möglichen Zuordnungen ab. Man setzt $b_i = 1$ für alle $i \in N_1$, $b_i = -1$ für alle $i \in N_2$ und $l_{ij} = 0$ und $u_{ij} = 1$ für alle $(i, j) \in A$.

Das Kürzeste-Wege-Problem
Das *Kürzeste-Wege-Problem (engl.: shortest path problem)* kann wie folgt formuliert werden: Gesucht wird ein kürzester Weg von einem Knoten s (Quelle) zu einem anderen Knoten t (Senke). Der Ausdruck „kurz" bezieht sich dabei auf die Kostenbewertung der enthaltenen Kanten (setzt man sie für alle Kanten gleich, bekommt man einen Weg mit minimaler Anzahl der Kanten).

Offensichtlich handelt es sich bei dem Kürzeste-Wege-Problem um einen Spezialfall des Minimalkostenfluss-Problems, bei dem $b_s = 1$, $b_t = -1$, $b_i = 0$ für alle $i \in N \setminus \{s, t\}$ und $l_{ij} = 0$ und $u_{ij} = 1$ für alle $(i, j) \in A$ gesetzt wird. Dann wird nämlich eine Flusseinheit durch das Netzwerk entlang des kürzesten Weges von s nach t transportiert.

Das Kürzeste-Wege-Problem kann mit relativ einfachen Algorithmen, wie z.B. dem *Dijkstra-* oder *Ford/Moore-Algorithmus*, gelöst werden. Es taucht häufig als Unterproblem in kombinatorischen und netzwerkorientierten Optimierungsproblemen auf. Eine Verallgemeinerung dieses Problems, die im Rahmen dieser Arbeit verwendet wird, ist das Problem zur Bestimmung kürzester Wege zwischen allen Paaren von Knoten *(engl.: all-pair shortest path problem)* in einem Graphen. Es kann beispielsweise mit dem *Floyd/Warshall-Verfahren* gelöst werden.

Das ressourcenbeschränkte Kürzeste-Wege-Problem
Eine der Erweiterungen des klassischen Kürzeste-Wege-Problems ist das *(ressourcen)beschränkte Kürzeste-Wege-Problem (engl.: (resource) constrained shortest path problem, RCSP)*. Dabei ist jeder Weg nicht nur durch Kosten, sondern auch durch einen Ressourcenverbrauch spezifiziert.

Sei wie oben $G = (N, A)$ ein gerichteter Graph und R die Menge unterschiedlicher Ressourcen. Mit jeder Kante $(i, j) \in A$ wird außer Kosten c_{ij} auch einen Ressourcenverbrauch d_{ij}^r für jede Ressource $r \in R$ assoziiert. Ein Pfad $\mathcal{P}$ von s nach t verbraucht $\sum_{(i,j)\in\mathcal{P}} d_{ij}^r$ Ressourcen vom Typ $r \in R$. Der Zulässigkeitsbereich jeder Ressource $r \in R$ ist durch ein Intervall $[l^r, u^r]$ definiert. Um die mathematische Formulierung des (kantenbasierten) ressourcenbeschränkten Kürzeste-Wege-Problems zu bekommen, wird die Formulierung des Kürzeste-Wege-Problems um folgende Restriktionen erweitert:

$$l^r \leq \sum_{(i,j)\in A} d_{ij}^r x_{ij} \leq u^r \qquad \forall r \in R \tag{3.4}$$

Das Mehrgüterfluss-Problem
Alle oben diskutierten Netzwerkprobleme setzen einen homogenen Fluss (Eingüter-Fluss) über die Kanten des Netzwerks voraus. In vielen Fällen müssen jedoch unterschiedliche Güter modelliert werden, die jeweils unterschiedliche Senken und Quellen haben, aber ein gemeinsames Netzwerk nutzen (z.B. verschiedene Fahrzeugtypen). Jedes Gut hat dabei in jedem Knoten eine eigene Flusserhaltungsbedingung, aber sie nutzen alle zusammen eine gemeinsame Ressource, die durch die Kapazität der Kanten gegeben ist. Dieses Problem heißt das *Mehrgüterfluss-Problem (engl.: multicommodity network flow problem)* und ist eine direkte Verallgemeinerung des Minimalkostenfluss-Problems. Sei $G = (N, A)$ ein gerichteter Graph und K die Menge der Güter, die durch Kanten aus A fließen. Die Kosten für den Transport einer Flusseinheit des Gutes $k \in K$ durch die Kante $(i, j) \in A$ seien durch c_{ij}^k gegeben. Jede Kante $(i, j) \in A$ hat güterspezifische Kapazitäten l_{ij}^k bzw. u_{ij}^k, die den Transport des Gutes k auf ihr einschränken, und zusätzlich eine gemeinsame minimale bzw. maximale Kapazität L_{ij} bzw. U_{ij} für den Gesamtfluss auf (i, j). Das Angebot bzw. die Nachfrage nach einem Gut $k \in K$ im Knoten $i \in N$ wird durch b_{ij}^k definiert. Die Flussvariable x_{ij}^k gibt den Fluss des Gutes k auf der Kante (i, j) an. Das mathematische Modell kann wie folgt definiert werden:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k \tag{3.5}$$

$$\text{s.t.} \sum_{\{j:(i,j) \in A\}} x_{ij}^k - \sum_{\{j:(j,i) \in A\}} x_{ji}^k = b_i^k \qquad \forall i \in N, \forall k \in K \tag{3.6}$$

$$l_{ij}^k \leq x_{ij}^k \leq u_{ij}^k \qquad \forall (i, j) \in A, \forall k \in K \tag{3.7}$$

$$L_{ij} \leq \sum_{k \in K} x_{ij}^k \leq U_{ij} \qquad \forall (i, j) \in A \tag{3.8}$$

Ein wichtiger Unterschied zwischen Ein- und Mehrgüterfluss-Problemen ist die Eigenschaft der Ganzzahligkeit einer Lösung. Während das Minimalkostenfluss-Problem bei ganzzahligen Kapazitäten immer eine ganzzahlige Lösung liefert, ist das bei dem Mehrgüterfluss-Problem nicht unbedingt der Fall. Will man hier nur ganzzahlige Lösungen akzeptieren, muss dies explizit von Flussvariablen gefordert werden. Die ganzzahlige Version des Mehrgüterfluss-Problems ist $\mathcal{NP}$-vollständig, sobald mindestens zwei Güter vorhanden sind (siehe [Garey and Johnson, 1979]).

3.1.2 Set-Partitioning- and Covering-Probleme

Viele Zuordnungsprobleme (z.B. Dienstplanung) lassen sich als ein *Set-Partitioning-Problem (SPP)* formulieren. Ein Set-Partitioning-Problem ist ein Auswahlproblem, bei dem eine Menge von disjunkten Teillösungen gesucht wird, sodass die Gesamtheit der ausgewählten Teillösungen bestimmte Eigenschaften erfüllt.

Sei N die Menge aller Teillösungen und M die Menge aller Elemente. Die Kosten einer Teillösung $j \in N$ seien durch c_j definiert. Sei a_{ij} genau dann 1, wenn die Teillösung j das Element i enthält und sonst 0. Die Teillösungen werden also als Spalten und die Elemente als Zeilen dargestellt. Die binäre Entscheidungsvariable x_j gibt an, ob eine Teillösung j zu der optimalen Lösung gehört oder nicht. Das Ziel ist nun eine kostenminimale Menge von Teillösungen zu finden, sodass jedes Element in genau einer Teillösung enthalten ist. Das zugehörige mathematische Modell lässt sich wie folgt formulieren:

$$\min \sum_{j \in N} c_j x_j \tag{3.9}$$

$$\text{s.t.} \sum_{j \in N} a_{ij} x_j = 1 \qquad \forall i \in M \tag{3.10}$$

$$x_j \in \{0, 1\} \qquad \forall j \in N \tag{3.11}$$

Es wurde bewiesen, dass das Set-Partitioning-Problem ein $\mathcal{NP}$-schweres Problem ist (siehe [Garey and Johnson, 1979]).

Das *Set-Covering-Problem (SCP)* unterscheidet sich nur wenig vom SPP. Der Unterschied besteht darin, dass beim SCP jedes Element in mindestens einer (anstatt in genau einer) Teillösung enthalten sein muss. Im mathematischen Modell wird lediglich das Gleichheitszeichen in den Restriktionen (3.10) durch ein Größer-Gleich-Zeichen ersetzt.

Zwischen den beiden Problemen besteht folgender Zusammenhang: Jede zulässige Lösung eines SPP ist auch eine zulässige - aber nicht notwendigerweise optimale - Lösung für das zugehörige SCP. Die optimale Lösung für ein SCP ist eine untere Schranke für das zugehörige SPP.

Ein Sonderfall des Set-Partitioning/Covering-Problems stellt das so genannte *generalisierte Set-Partitioning/Covering-Problem (engl.: generalized set partitioning/covering problem)* dar. Diese Variante erhält man, wenn man die Eins auf der rechten Seite der Nebenbedingungen (3.10) durch eine natürliche Zahl ersetzt.

3.2 Lagrange-Relaxation

Lagrange-Relaxation gehört zu einer der Standardtechniken der kombinatorischen Optimierung zur Berechnung von unteren Schranken und wird in diesem Bereich seit über dreißig Jahren eingesetzt. Die folgende Darstellung der grundlegenden Funktionsweise orientiert sich an [Beasley, 1993].

3.2.1 Grundidee

Die Grundidee des Verfahrens besteht darin, eine Menge von „schwierigen" Nebenbedingungen aus dem Modell zu streichen und ihre Verletzung in der Zielfunktion zu bestrafen. Als Grundlage der weiteren Ausführung betrachten wir ein gemischt-ganzzahliges Optimierungsproblem P:

$$\text{P:} \quad \min \quad \sum_{j \in J} c_j x_j \tag{3.12}$$

$$\text{s.t.} \quad \sum_{j \in J} a_{ij} x_j \geq b_i \qquad \forall i \in N_1 \tag{3.13}$$

$$\sum_{j \in J} d_{kj} x_j \geq e_k \qquad \forall k \in N_2 \tag{3.14}$$

$$x_j \geq 0, \quad \textit{ganzzahlig} \quad \forall j \in J \tag{3.15}$$

Das Problem P besitzt neben den Ganzzahligkeitsbedingungen (3.15) zwei Mengen von Restriktionen (3.13) und (3.14), von denen wir annehmen, dass es sich bei der ersten Menge um eher „schwierige" und bei der zweiten Menge um eher „einfache" Nebenbedingungen handelt. Das bedeutet, dass P ohne Restriktionen (3.13) relativ einfach gelöst werden kann.

Für jede schwierige Restriktion $i \in N_1$ definieren wir einen so genannten *Lagrange-Multiplikator* $\pi_i \geq 0$. Die Formulierung

$$\text{LR}(\pi): \quad \min \quad \sum_{j \in J} c_j x_j + \sum_{i \in N_1} \pi_i (b_i - \sum_{j \in J} a_{ij} x_j) \tag{3.16}$$

$$\text{s.t.} \quad \sum_{j \in J} d_{kj} x_j \geq e_k \qquad \forall k \in N_2 \tag{3.17}$$

$$x_j \geq 0 \quad \textit{ganzzahlig} \quad \forall j \in J \tag{3.18}$$

heißt die *Lagrange-Relaxation* $LR(\pi)$ von P bezüglich der Nebenbedingungen (3.13) und des Multiplikatoren-Vektors $\pi = (\pi_i)_{i \in N_1}$.

Wird eine relaxierte Restriktion i verletzt, d.h. $(b_i - \sum_{j \in J} a_{ij} x_j) > 0$, dann wird diese Verletzung die Zielfunktion (3.16) mit Strafkosten versehen, d.h. der Zielfunktionswert wird größer, da $\pi_i \geq 0$ für alle $i \in N_1$ ist. Eine zulässige Lösung x von $LR(\pi)$ ist also nicht notwendigerweise zulässig für P. Eine optimale Lösung von $LR(\pi)$ liefert jedoch für jeden Vektor $\pi > 0$ eine untere Schranke für die optimale Lösung von P. Denn für jede zulässige Lösung von P gilt $\sum_{i \in N_1} \pi_i (b_i - \sum_{j \in J} a_{ij} x_j) \leq 0$, und somit ist der Zielfunktionswert der Lagrange-Relaxation größer oder gleich dem Wert der Zielfunktion ohne Strafkosten.

Da wir jedoch nicht nur irgendeine untere Schranke für P, sondern eine möglichst gute berechnen wollen, suchen wir nach einem Vektor π, der die Lagrange-Funktion

$LR(\pi)$ maximiert. Dieses Problem wird als das *Lagrange-Multiplikator-Problem (LMP)* oder *Lagrange-Dual-Problem (LDP)* bezeichnet und wie folgt formuliert:

$$\text{LMP:} \qquad \max_{\pi \geq 0} \; LR(\pi) \tag{3.19}$$

Im besten Fall stimmt das Optimum von LMP mit dem Optimum des Ausgangsproblems P überein. Ist dies nicht der Fall, so spricht man von einer *Dualitätslücke (engl.: duality gap)*, die als relative Differenz zwischen den beiden Optima gemessen wird.

Wenn es sich bei den „schwierigen" Nebenbedingungen von P nicht um Ungleichungen, sondern um Gleichungen handelt, entfällt die Beschränkung $\pi \geq 0$. Die Lagrange-Multiplikatoren π_i können dann sowohl negative als auch positive Werte annehmen.

Eine wichtige Frage ist das Verhältnis von Lagrange-Relaxation zur LP-Relaxation[1]. Sei c_{LP} der optimale Wert der LP-Relaxation, c_{opt} der optimale Wert des Ausgangsproblems P und π^* der optimale Vektor, der $LR(\pi)$ maximiert. Es lässt sich zeigen, dass $c_{LP} \leq LR(\pi^*) \leq c_{opt}$ gilt (siehe z.B. [Wolsey, 1998]). Das heißt also, dass die Lagrange-Schranke mindestens so gut wie die LP-Schranke ist. Außerdem gilt $c_{LP} = LR(\pi^*)$, wenn die Lagrange-Relaxation die so genannte *Ganzzahligkeitseigenschaft (engl.: integrality property)* besitzt. Diese Eigenschaft ist gegeben, falls die Ganzzahligkeitsbedingung in (3.18) redundant ist, d.h. sie lässt sich durch die Bedingung $x_j \geq 0,\ x_j \in \mathbb{R}$ für alle $j \in J$ ersetzen, ohne dass sich der Lösungsvektor x (bei allen möglichen Multiplikatoren π) verändert.

Die Funktion $LR(\pi)$ ist stückweise linear, konkav und nichtdifferenzierbar, daher ist das LMP ein nichtlineares Optimierungsproblem. In den nächsten zwei Unterabschnitten werden zwei Methoden vorgestellt, die zur approximativen Lösung des Lagrange-Multiplikator-Problems im Rahmen dieser Arbeit eingesetzt werden.

3.2.2 Subgradienten-Verfahren

Das *Subgradienten-Verfahren* ist ein iteratives Suchverfahren, um nichtlineare und nichtdifferenzierbare Funktionen zu optimieren. Es wurde zuerst von [Held and Karp, 1971] eingesetzt, um das Lagrange-Multiplikator-Problem approximativ zu lösen, d.h. gute Multiplikatoren π zu bestimmen, die die Lagrange-Funktion $LR(\pi)$ maximieren. Gestartet wird mit einer Initiallösung π^1, die dann in jeder Iteration t mit Hilfe einer *Suchrichtung* d^t und einer *Schrittweite* w^t aktualisiert wird, bis ein Abbruchkriterium erfüllt ist. Das Grundverfahren läuft wie folgt ab (vgl. [Beasley, 1993, S. 267ff]):

[1] *LP-Relaxation* ist eine sehr verbreitete Relaxationstechnik, bei der in einem gemischtganzzahligen Programm auf die Forderung der Ganzzahligkeit von Variablen verzichtet wird

Schritt 1: Initialisierung
Initialisiere π^1, setze $t = 1$,
berechne obere Schranke UB (z.B. durch eine Heuristik);

Schritt 2: Löse Lagrange-Relaxation
Löse $LR(\pi^t)$ mit aktuellen Multiplikatoren π^t und
bekomme die zugehörige optimale Lösung x^t.

Schritt 3: Berechne Suchrichtung d^t
Berechne Subgradient $s_i^t = b_i - \sum_{j \in J} a_{ij} x_j^t$ für alle $i \in N_1$,
setze[2] $d^t = s^t$.

Schritt 4: Berechne Schrittweite w^t
$w^t = \lambda \frac{UB - LR(\pi^t)}{\sum_{i \in N_1} (d_i^t)^2}$

Schritt 5: Aktualisiere Lagrange-Multiplikatoren
$\pi_i^{t+1} = \pi_i^t + w^t d_i^t$ für alle $i \in N_1$

Schritt 6: Überprüfe Abbruchkriterien
Abbruch wenn mindestens eins der folgenden Kriterien erfüllt ist:

$s^t = 0$,
$\sum_{i \in N_1} (d_i^t)^2 \leq \epsilon$,
$\lambda \leq \epsilon$,
$t \geq t_{max}$,
$UB - LR(\pi^t) \leq \epsilon$,

sonst setzte $t = t + 1$ und gehe zum Schritt 2.

Der Parameter λ im Schritt 4 heißt *Skalierungsparameter*, der in der Regel verkleinert wird (meistens halbiert), wenn das Verfahren in einer Reihe von aufeinanderfolgenden Iterationen keine Verbesserung des Zielfunktionswertes erreicht *(engl.: stalling)*. Um Konvergenz zu gewährleisten, sollte $0 \leq \lambda \leq 2$ sein.

Wenn der Subgradienten-Vektor gleich Null ist (erste Abbruchbedingung), dann ist die dual optimale Lösung auch für das Ausgangsproblem P primal zulässig und somit optimal. Allerdings tritt dieser Fall in der Praxis so gut wie nie ein. Häufiger bricht die Suche ab, wenn die Norm der Suchrichtung (zweite Abbruchbedingung) oder die Schrittweite (dritte Abbruchbedingung) einen Wert nahe Null erreichen. Durch weitere Iterationen ist dann keine große Verbesserung von $LR(\pi)$ zu erwarten. Die Wahl von ϵ legt die Genauigkeit der Lösung fest und bestimmt auch maßgeblich die Anzahl der Iterationen. Die Gesamtanzahl der Iterationen wird durch t_{max} begrenzt.

[2] In der Grundversion des Verfahrens, die in der Praxis meist eingesetzt wird, wird die Suchrichtung d^t gleich dem Subgradienten-Vektor s^t gesetzt.

3.2.3 Volume-Algorithmus

Das oben beschriebe Subgradienten-Verfahren findet dank seiner Einfachheit und Schnelligkeit eine sehr weite Verbreitung als Methode zur Lösung des Lagrange-Multiplikator-Problems. Allerdings besitzt das Subgradienten-Verfahren einige entscheidende Nachteile. So liefert es keine Lösung für die primalen Variablen und hat keine gut definierten Abbruchkriterien. Vor kurzem präsentierten Barahona und Anbil [Barahona and Anbil, 2000] eine Erweiterung des klassischen Subgradienten-Verfahrens, in der zusätzlich zu der dualen Lösung eine Approximation der primalen Lösung berechnet wird. Außerdem besitzt die neue Methode, die von den Autoren *Volume Algorithm* genannt wird, neben einer vergleichbaren Laufzeit auch bessere Abbruchbedingungen. Der Name stammt von der Art und Weise ab, wie die primale Lösung berechnet wird. Das geschieht durch eine Abschätzung des Volumens unterhalb der aktiven Flächen der dualen Lösung. Außerdem bestimmt das Volumen auch die Suchrichtung im Laufe des Verfahrens.

Sei P, $LP(\pi)$ und π wie oben definiert und zusätzlich $\bar{x}$ ein Vektor mit primalen Variablen und $\bar{\pi}$ die besten gefundenen Lagrange-Multiplikatoren. Der Volume-Algorithmus kann wie folgt beschrieben werden (vgl. [Barahona and Anbil, 2000]):

Schritt 1: Initialisierung
Initialisiere $\bar{\pi}$, $\bar{x}$,
setze $t = 1$, $\pi^1 = \bar{\pi}$.

Schritt 2: Löse Lagrange-Relaxation
Löse $LR(\pi^t)$ mit aktuellen Multiplikatoren π^t und
bekomme die zugehörige optimale Lösung x^t.
Wenn $LR(\pi^t) > LR(\bar{\pi})$, dann aktualisiere $\bar{\pi} = \pi^t$.

Schritt 3: Berechne Suchrichtung d^t
Berechne Subgradient $s_i^t = b_i - \sum_{j \in J} a_{ij} x_j^t$ für alle $i \in N_1$,
berechne Suchrichtung $d^t = \alpha s^t + (1 - \alpha) d^{t-1}$,
berechne primale Lösung $\bar{x} = \alpha x^t + (1 - \alpha)\bar{x}$.

Schritt 4: Berechne Schrittweite w^t
$w^t = \lambda \frac{T - LR(\bar{\pi})}{\sum_{i \in N_1} (d_i^t)^2}$

Schritt 5: Aktualisiere Lagrange-Multiplikatoren
$\pi_i^{t+1} = \bar{\pi}_i + w^t d_i^t$ für alle $i \in N_1$

Schritt 6: Überprüfe Abbruchkriterien
Abbruch wenn mindestens eins der folgenden Kriterien erfüllt ist:
$$\left| \frac{\sum_{j \in J} c_j \bar{x}_j - LR(\bar{\pi})}{LR(\bar{\pi})} \right| \leq \epsilon' \text{ und } \frac{\sum_{i \in N_1} |d_i^t|}{|N_1|} \leq \epsilon'',$$

$t \geq t_{max}$,
$UB - LR(\bar{\pi}) \leq \epsilon$,
sonst setzte $t = t + 1$ und gehe zum Schritt 2.

Wie man aus dem Ablauf sieht, weist der Volume-Algorithmus einige Unterschiede zum klassischen Subgradienten-Verfahren auf (vgl. Unterabschnitt 3.2.2):

- Die Multiplikatoren werden im Schritt 5 ausgehend von den bis dahin besten Multiplikatoren π^* aktualisiert. Die Aktualisierung der besten Multiplikatoren findet nur statt, wenn im Schritt 2 eine bessere Lagrange-Schranke gefunden wurde. Die Autoren imitieren damit die Bundle-Methode [Lemaréchal, 1989], ohne dabei ein quadratisches Optimierungsproblem lösen zu müssen.

- Im Schritt 3 wird eine Approximation der primalen Lösung als eine konvexe Kombination der lokalen Lösungen aus jeder Iteration berechnet. Diese Überlegung basiert auf der Dualitätstheorie (siehe [Barahona and Anbil, 2000]). Der Gewichtungsfaktor α wird so gewählt, dass er das Problem $\|\alpha s^t + (1-\alpha)d^t - 1\|$ minimiert, wobei $0 < \alpha \leq 1$ gelten muss.

- Bei der Aktualisierung der Suchrichtung w^t im Schritt 3 werden die früheren Suchrichtungen mitberücksichtigt. Dadurch wird das Zickzack-Verhalten des Subgradienten-Verfahrens vermieden.

- Der Skalierungsfaktor λ im Schritt 4 wird nach folgendem Schema berechnet: Man unterscheidet drei Typen von Iterationen:
  - *Rot:* Wird in einer Iteration keine Verbesserung der Lagrange-Funktion gefunden, so wird sie *rot* genannt. Nach einer Sequenz von roten Iterationen wird λ verringert.
  - *Gelb:* Ist $LR(\pi^t) > LR(\bar{\pi})$, dann wird $d = s^t d^{t-1}$ berechnet. Ist $d < 0$, dann wird die Iteration *gelb* genannt. Bei einer gelben Iteration wird der alte Wert für λ beibehalten.
  - *Grün:* Ist $LR(\pi^t) > LR(\bar{\pi})$ und $d \geq 0$, wird die Iteration *grün* genannt und der Skalierungsfaktor λ wird vergrößert.

- Im Schritt 4 wird anstatt von $UB - LR(\pi^t)$ die Differenz $T - LR(\bar{\pi})$ benutzt, wobei T der sogenannte Zielwert ist *(engl.: target value)*. Gestartet wird mit einem kleinen Ziel, und jedes Mal, wenn $LR(\bar{\pi}) \geq 0,95T$ gilt, wird T auf $1,05LR(\bar{\pi})$ erhöht.

- Der Algorithmus wird abgebrochen, wenn die Differenz zwischen der primalen und dualen Lösung weniger als $\epsilon'(\%)$ ist, dabei muss die durchschnittliche Verletzung der relaxierten Restriktionen unter ϵ'' liegen.

Der Volume-Algorithmus wurde erfolgreich auf unterschiedlichen Klassen von Optimierungsproblemen getestet. Für weitere Details wird auf [Barahona and Anbil, 2000], [Barahona and Anbil, 2002] und [Bahiense et al., 2002] verwiesen.

3.3 Column-Generation-Verfahren

Column-Generation ist das Verfahren, das zur Lösung linearer Optimierungsprobleme mit einer großen Anzahl an Variablen eingesetzt wird. Die Grundidee stammt von Dantzig und Wolfe [Dantzig and Wolfe, 1960] im Kontext der *Dantzig-Wolfe-Dekomposition* für lineare Optimierungsprobleme und lässt sich wie folgt beschreiben (siehe Abbildung 3.1):

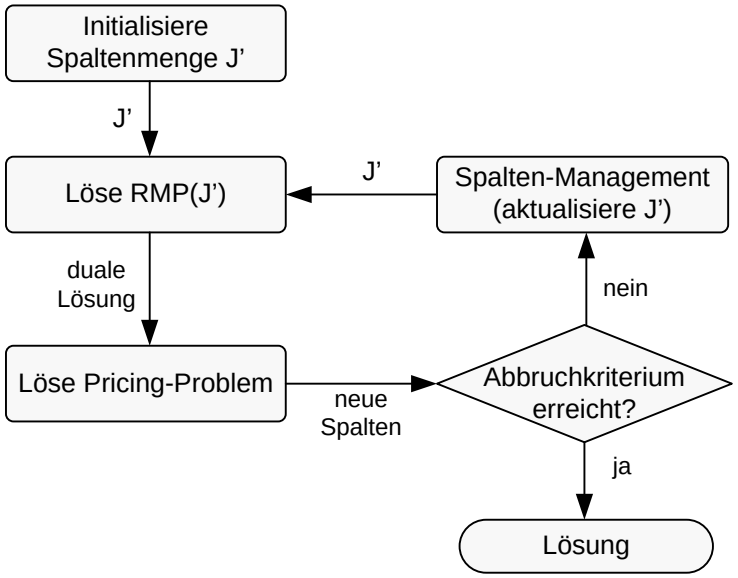

Abbildung 3.1: Column-Generation-Schema

Anstatt ein großes Optimierungsproblem mit allen Variablen (Spalten) zu lösen, wird in einer Reihe von Iterationen je eine kleine Untermenge der Variablen explizit betrachtet. So ein partielles Problem mit einer Auswahl an Spalten wird *das eingeschränkte Master-Problem* (*engl: Restricted-Master-Problem, RMP*) genannt. Nachdem das RMP gelöst wurde, werden die dualen Informationen der Lösung, die Schattenpreise, dazu genutzt, in einem Unterproblem neue Spalten zu finden, die die Lösung verbessern können. Die Dualitätstheorie besagt, dass das nur solche Spalten sein können, die negative reduzierte Kosten aufweisen (im Falle einer Minimierung). Das RMP wird um diese Spalten erweitert und erneut gelöst. Das Problem des Findens neuer Spalten wird als *Pricing-Problem* bezeichnet. Das Column-Generation-Verfahren terminiert, wenn keine neuen Spalten mit negativen reduzier-

ten Kosten gefunden werden können. Das bedeutet, dass die optimale Lösung des aktuellen RMP nicht mehr verbessert werden kann und somit auch die optimale Lösung des Ausgangsproblems ist. Diese Vorgehensweise entspricht in etwa der Aufnahme einer Nicht-Basisvariablen in die Basis beim Simplex-Algorithmus. Wird das RMP durch Aufnahme neuer Spalten zu groß, kann es im Spalten-Management z.B. durch Löschen von Spalten mit hohen positiven reduzierten Kosten wieder verkleinert werden.

Findet die Spaltenauswahl aus einer Menge explizit verfügbarer Spalten statt (d.h. sie sind alle erzeugt und befinden sich im Computerspeicher), wird das gesamte Verfahren als *explizites* Column-Generation-Verfahren bezeichnet. Sind dagegen alle Spalten nur implizit verfügbar (d.h. während des Pricing-Problems werden nur die notwendigen generiert), heißt diese Variante das *implizite* Column-Generation-Verfahren. Eine Kombination aus beiden Varianten ist auch möglich.

Grundsätzlich reicht nur eine Variable mit negativen reduzierten Kosten, um das Column-Generation-Verfahren fortzusetzen. Diese Strategie wird *Single Pricing* genannt. Durch die Aufnahme einer Menge von solchen Variablen auf einmal (als *Multiple Pricing* bezeichnet) kann aber tendenziell die Anzahl der Iterationen gesenkt werden. Allerdings wächst dabei die Größe des RMP von Iteration zu Iteration schneller und es ist somit immer schwieriger zu lösen. Grundsätzlich ist bei jedem Problem ein Kompromiss zwischen der Anzahl der Iterationen und der Dauer einzelner Iterationen zu finden.

Das Pricing-Problem kann je nach Formulierung oft als ein eigenständiges Optimierungsproblem aufgefasst werden, aus dessen Lösung neue Spalten für Column-Generation implizit bestimmt werden. Ist dabei das Pricing-Problem $\mathcal{NP}$-schwer, gehört auch das gesamte Column-Generation-Verfahren zur Klasse $\mathcal{NP}$-schwerer Probleme (siehe [Desrosiers and Lübbecke, 2005]).

Column-Generation-Verfahren und gemischt-ganzzahlige Optimierung
Column-Generation-Verfahren wurde ursprünglich nur zur Lösung linearer Programme eingesetzt. Ist das Ausgangsproblem ein gemischt-ganzzahliges Programm, wird als Master-Problem seine LP-Relaxation benutzt. Um die Ganzzahligkeit der Lösung herzustellen, wird oft das *Branch-and-Bound*-Verfahren eingesetzt (siehe auch 3.5). Dabei sind zwei Strategien möglich:

- Die einfachste Variante ist das Branch-and-Bound-Verfahren (B&B) nach dem Column-Generation-Verfahren auszuführen und dabei nur Spalten aus dem letzten RMP zu betrachten. Das Column-Generation-Verfahren wird also nur im *Wurzelknoten (engl.: root node)* des B&B-Baumes ausgeführt, somit können keine neuen Spalten während der B&B-Suche generiert werden. Allerdings garantiert diese Vorgehensweise keine optimale ganzzahlige Lösung, da

sich die dualen Informationen während der B&B-Suche durch das Hinzufügen neuer Restriktionen, die die Ganzzahligkeit für bestimmte Variablen erzwingen, ändern können. Somit können Variablen, deren Aufnahme bisher nicht vorteilhaft war, nun negative reduzierte Kosten aufweisen. Die Qualität dieser heuristischen Vorgehensweise hängt von der jeweiligen Problemstruktur ab.

- Will man eine optimale ganzzahlige Lösung bestimmen, muss das Column-Generation-Verfahren in das Branch-and-Bound-Verfahren eingebunden werden. Somit wird die Möglichkeit gegeben, neue Spalten auch während der B&B-Suche zu generieren. Dieses Verfahren wird *Branch-and-Price* genannt (siehe [Barnhart et al., 1998]).

Column-Generation und Lagrange-Relaxation

LP-Relaxation ist nur eine Methode zur Berechnung unterer Schranken. Eine Alternative dazu ist die Lagrange-Relaxation. [Carraresi et al., 1995] und [Freling, 1997] schlagen vor, Lagrange-Relaxation mit Column-Generation zu kombinieren, indem die dualen Variablen des RMP nicht optimal mit z.B. Simplex-Methode, sondern approximativ mit Lagrange-Relaxation bestimmt werden. Die optimalen Lagrange-Multiplikatoren werden mit dem Subgradienten-Verfahren in jeder Iteration von Column-Generation bestimmt. Sie stellen eine Approximation der optimalen dualen Variablen des RMP dar und können zur Generierung neuer Spalten im Pricing-Problem benutzt werden.

Da die duale Lösung nur eine Approximation ist, müssen die Lagrange-Multiplikatoren vor jedem Pricing-Schritt angepasst werden, um zu verhindern, dass Spalten, die schon im RMP enthalten sind, nochmal generiert werden. [Freling, 1997] und [Carraresi et al., 1995] beschreiben dazu eine Greedy-Heuristik, die die Lagrange-Multiplikatoren so modifiziert, dass alle Spalten aus RMP nichtnegative reduzierte Kosten bekommen und der Wert der Lagrange-Funktion sich dabei nicht verringert.

Die Benutzung von Lagrange-Relaxation anstatt der LP-Relaxation im Column-Generation weist einige Vorteile auf:

- Das Subgradienten-Verfahren ist sehr schnell, leicht zu implementieren und macht die Notwendigkeit des Einsatzes von kommerziellen Optimierungsbibliotheken überflüssig.
- Während der Subgradienten-Phase werden mögliche zulässige Lösungen generiert.
- Lagrange-Relaxation liefert bessere Schranken für einige Problemtypen (siehe z.B. [Beasley, 1993]).

- Es wurde gezeigt, dass die Benutzung von Lagrange-Relaxation die Degeneration des Problems verringert und das Column-Generation-Verfahren beschleunigt (siehe z.B. [Jans and Degraeve, 2004]).

- Master-Probleme mit einer großen Zahl von Restriktionen werden mit einem Lagrange-Relaxation-basierten Verfahren oft schneller als mit einem LP-Solver gelöst.

Die Kombination aus Column-Generation und Lagrange-Relaxation ist die Hauptkomponente der Algorithmen, die im Rahmen dieser Arbeit zur Lösung integrierter Umlauf- und Dienstplanungsprobleme entwickelt wurden.

Für einen guten Überblick über das Column-Generation-Verfahren und dessen Einsatz zur Lösung von Optimierungsproblemen wird auf [Lübbecke and Desrosiers, 2005] und [Desaulniers et al., 2005] verwiesen. Techniken zur Beschleunigung des Column-Generation-Verfahrens werden u.a. in [Desaulniers et al., 2001] diskutiert. Das Problem der starken Degeneration primaler und dualer Lösung, das zur schlechten Konvergenz des Prozesses und zu den Instabilitäten der dualen Variablen führt, wird u.a. von [du Merle et al., 1999] und [Ben Amor et al., 2004] behandelt.

3.4 Simulated Annealing

Es ist ein grundlegendes Problem jedes lokalen Suchverfahrens, in einem schlechten lokalen Optimum zu enden. Darum wurde eine Reihe von Strategien entwickelt, die diesen Nachteil beheben. Die Grundidee von *Simulated Annealing* besteht darin, mit einer gewissen Wahrscheinlichkeit auch eine schlechtere Lösung als die bisher gefundene als Ausgangspunkt für die weitere Nachbarschaftssuche zu akzeptieren. Diese Wahrscheinlichkeit ist von dem Ausmaß der Verschlechterung abhängig. Weiterhin wird die Akzeptanzwahrscheinlichkeit durch den so genannten Temperaturparameter so kontrolliert, dass sie mit dem fortschreitendem Lösungsprozess immer seltener eine Verschlechterung akzeptiert.

Wichtige Fragestellungen bei der Konstruktion von Simulated Annealing für ein konkretes Problem sind z.B. wie die *Nachbarschaft*, d.h. der Übergang von einer Lösung zu einer anderen definiert ist, die Wahl der *Starttemperatur* oder wie sie im Laufe des Prozesses verringert wird (*Abkühlungsschemata*) usw. Für eine ausführliche Beschreibung von Simulated Annealing wird auf [Dowsland, 1993] verwiesen.

3.5 Branch-and-Bound

Branch-and-Bound (B&B) ist ein Suchverfahren, um gemischt-ganzzahlige Optimierungsprobleme zu lösen. Die Grundidee besteht in einer hierarchischen Zerlegung eines Optimierungsproblems in Teilprobleme, sodass die optimale Lösung eines übergeordneten Problems in einem der Teilprobleme enthalten ist oder aus mehreren Teilproblemen rekonstruiert werden kann. Die Zerlegung lässt sich durch einen Entscheidungsbaum (B&B-Baum) repräsentieren, der zuerst nur aus einem Wurzelknoten besteht und im Laufe des Verfahrens sukzessive aufgebaut und abgearbeitet wird. Die Knoten des Baumes entsprechen den Teilproblemen, die durch die Zerlegung des Elternknotens gebildet wurden. Die zwei Hauptkomponenten des Branch-and-Bound-Verfahrens sind *Branching* und *Bounding*.

Branching
Branching oder *Verzweigung* beschreibt zunächst in allgemeiner Form wie ein Problem in Teilprobleme zerlegt wird. Die erste Frage ist dabei, wie man ausgehend vom aktuellen Knoten neue Knoten erzeugen kann. Eine mögliche Strategie ist, eine Variable auszuwählen, die in der LP-Lösung einen fraktionalen Wert besitzt, und ihren zulässigen Wertebereich so zu modifizieren, dass der aktuelle fraktionale Wert ausgeschlossen wird. Die zweite Frage ist, auf welchen Variablen verzweigt werden soll. Beispiele für die Variablenauswahl-Strategien sind Verzweigen auf Variablen mit geringsten bzw. höchsten (reduzierten) Kosten oder auf Variablen, die am nächsten an einem ganzzahligen Wert liegen usw. Die letzte Frage ist, in welcher Reihenfolge die bereits erzeugten Knoten abgearbeitet werden. In der Literatur unterscheidet man drei grundlegende Suchstrategien, nämlich Tiefen-, Breiten- und Bestensuche.

Bounding
Für die neu generierten Knoten werden die assoziierten Unterprobleme dahingehend untersucht, ob der B&B-Baum weiter verzweigt werden muss (*Verzweigungsfall*), oder der Teilbaum unter dem Knoten von der weiteren Betrachtung ausgenommen werden kann (*Auslotungsfall*). Dazu wird eine Relaxation des Problems gelöst und mit Hilfe von oberen und unteren Schranken - je nachdem, ob es sich um ein Maximierungs- oder Minimierungsproblem handelt - entschieden, welcher der beiden Fälle vorliegt.

Eine Relaxation eines Problems bekommt man, indem man einige oder alle Nebenbedingungen des Problems weglässt oder „schwächt". Die gängigsten Relaxationen für MIP-Probleme sind die *LP-Relaxation* oder die *Lagrange-Relaxation*. Der Zielfunktionswert der Lösung der LP-Relaxation liefert eine untere Schranke (im Falle eines Minimierungsproblems) für das Ausgangsproblem. Ist die Lösung

der Relaxation besser als die bisher beste gefundene Lösung, aber nicht zulässig für das Ausgangsproblem, werden ausgehend von diesem Knoten weitere Teilprobleme gebildet (Verzweigungsfall). Ist diese Lösung aber auch für das Ausgangsproblem zulässig, wird sie als neue beste Lösung gespeichert und für diesen Knoten keine weitere Verzweigung vorgenommen. In diesem Fall wird das eigentliche *Bounding* oder Abschneiden durchgeführt: Es werden alle Knoten und ihre Unterbäume im B&B-Baum gestrichen, deren untere Schranken schlechter als der Zielfunktionswert der eben gefundenen MIP-Lösung ist. Besitzt die Relaxation keine zulässige Lösung oder ist sie schlechter als die bisher beste gefundene Lösung, dann wird der Knoten als „bounded" erkannt und nicht weiter bearbeitet, da seine Teilbäume keine bessere ganzzahlige Lösung mehr enthalten können.

Das Branch-and-Bound-Verfahren kann mit anderen Optimierungsverfahren kombiniert werden. Ein Beispiel dafür ist der bereits erwähnte ***Branch-and-Price-Algorithmus***, der eine Kombination von Branch-and-Bound und Column-Generation darstellt. Branch-and-Price wird erfolgreich bei Problemen mit sehr vielen Variablen eingesetzt. Eine weitere Erweiterung von Branch-and-Bound ist der ***Branch-and-Cut-Algorithmus***, bei dem in jedem Knoten des Entscheidungsbaumes ein *Schnittebeneverfahren (engl.: cutting plane algorithm)* angewandt wird. Die speziell definierten und hinzugefügten Schnittebenen grenzen einen Teil des Lösungsraums in einem Knoten ein, um dadurch eine bessere untere Schranke zu berechnen. Branch-and-Cut wird erfolgreich bei Problemen mit sehr vielen Restriktionen eingesetzt, die zunächst ignoriert werden. Verletzt die Lösung eines Unterproblems in einem B&B-Knoten eine bzw. mehrere Restriktionen, werden nur die Verletzten dem Problem hinzugefügt, um die somit unzulässige Lösung auszuschließen. Wird in ein Branch-and-Bound-Schema sowohl Column-Generation als auch ein Schnittebeneverfahren eingebettet, dann wird eine solche Technik als ***Branch-and-Cut-and-Price*** bezeichnet.

Kapitel 4

Methoden der Umlauf- und Dienstplanung: Stand der Forschung

In diesem Kapitel wird ein Überblick über die wichtigen Arbeiten im Bereich sequenzieller und integrierter Umlauf- und Dienstplanung gegeben. Insbesondere werden die unterschiedlichen Modellierungs- und Lösungsansätze für die beiden Planungsprobleme im einzelnen und bei einer gleichzeitigen Betrachtung diskutiert.

4.1 Sequenzielle Umlauf- und Dienstplanung

Probleme der (sequenziellen) Umlauf- und Dienstplanung liefern eine gute Einführung in das integrierte Problem. Zum einen basieren Modelle und Algorithmen für das integrierte Umlauf- und Dienstplanungsproblem auf Modellen und Algorithmen der zugrunde liegenden Unterprobleme. Zum anderen benutzen wir die Lösung der traditionellen (sequenziellen) Umlauf- und Dienstplanung als Referenz zur Beurteilung der Lösungsqualität integrierter Verfahren.

4.1.1 Umlaufplanung

Das Problem der Umlaufbildung im Busverkehr wird seit über 20 Jahren von zahlreichen Forschern behandelt. Dabei lässt sich oft eine sehr ähnliche Vorgehensweise bei der Formulierung des Umlaufplanungsproblems feststellen: Erst wird das Problem als ein Netzwerkmodell modelliert, wovon anschließend eine geeignete mathematische Formulierung abgeleitet wird. Die Fahrgastfahrten werden typischerweise durch Knoten oder Kanten im Netzwerk dargestellt. Alle zulässigen Verbindungsfahrten sowie Ausrück- und Einrückfahrten werden im Netzwerkmodell durch gerichtete Kanten oder Wege abgebildet, die die entsprechenden Knoten verbinden.

Die durch das Netzwerk fließenden Flusseinheiten entsprechen den eingesetzten Fahrzeugen. Knoten, die eine Flusseinheit „besucht", bilden einen Umlauf.

Die bekannten Lösungsansätze unterscheiden sich in der Struktur der Netzwerkmodelle, mathematischer Formulierung und in den angewandten Algorithmen bzw. Lösungsverfahren. Im Folgenden werden die wichtigsten Arbeiten in diesem Bereich zusammengefasst, wobei zunächst die einfache Variante mit nur einem Depot und erst dann das Mehrdepot-Umlaufplanungsproblem diskutiert wird.

Eindepot-Umlaufplanungsproblem (SDVSP)
Wie schon erwähnt, kann das SDVSP in polynomieller Zeit gelöst werden. In den letzten zwei Jahrzehnten wurde der Entwicklung schneller Lösungsmethoden für dieses Problem viel Aufmerksamkeit gewidmet. Nicht nur weil das SDVSP ein interessantes Optimierungsproblem an sich darstellt, sondern weil es von vielen Autoren außerdem als Unterproblem in komplexen Problemstellungen wie das MDVSP oder das VCSP verwendet wird. In der Literatur werden mehrere netzwerkflussbasierte Modelle und Algorithmen für das SDVSP beschrieben. [Carraresi and Gallo, 1984] formulieren das SDVSP als *Zuordnungsproblem*, das mit einem Minimalkostenfluss-Algorithmus gelöst werden kann. [Paixão and Branco, 1987] schlagen einen speziell für das SDVSP entwickelten Algorithmus vor, der auf einer *Quasi-Assignment*-Formulierung basiert. Sie vergleichen ihren Ansatz mit Formulierungen als *Transport-* und *Zuordnungsproblem* und zeigen, dass er signifikant überlegend ist. [Song and Zhou, 1990] und [Dell'Amico et al., 1993] stellten einen *Successive-Shortest-Path-Algorithmus* für SDVSP vor.

[Löbel, 1996] beschreibt eine effiziente Implementierung eines *Netzwerk-Simplex-Algorithmus* für das Minimalkostenfluss-Problem, das für die Formulierung des SDVSP verwendet wurde. [Freling, 1997] formuliert SDVSP als ein *Quasi-Assignment-Problem* und löst es mit Hilfe eines effizienten *kombinierten Vorwärts- und Rückwärts-Auktions-Algorithmus* aus [Bertsekas and Castañon, 1992]. [Silva et al., 1999] nutzen eine flussbasierte Formulierung für SDVSP und schlagen ein *Arc-Generation*-Verfahren vor. Das Prinzip dieser Methode ist dem des Column-Generation-Verfahrens ähnlich: Gestartet mit einem Netzwerk mit nur direkten Verbindungskanten wird das Problem mehrmals iterativ gelöst. In jeder Iteration werden anhand von dualen Informationen der nicht berücksichtigten Kanten, die eine Über-Depot-Verbindung darstellen, diejenige von ihnen ermittelt und dem Netzwerk hinzugefügt, die den Zielfunktionswert verbessern können.

[Daduna and Paixão, 1995] und [Desrosiers et al., 1995] präsentieren gute Überblicke über Algorithmen und Applikationen für SDVSP, die bis dahin veröffentlicht worden waren. Außerdem diskutieren sie einige Erweiterungen des Problems.

Mehrdepot-Umlaufplanungsproblem (MDVSP)
Die existierenden Lösungsansätze für das MDVSP lassen sich in zwei grundlegende Gruppen aufteilen, nämlich Lösungsansätze, die das Problem optimal oder annähernd optimal (heuristisch) lösen.

Der Grund für den Einsatz *heuristischer* Verfahren beim MDVSP ist, dass damit oft Probleme mit mehr Fahrgastfahrten und Depots in kürzerer Zeit gelöst werden können und die Abweichung von der optimalen Lösung oft nur sehr klein und/oder aus praktischer Sicht akzeptabel ist. Die meisten heuristischen Lösungsansätze aus der Literatur lassen sich in eine der folgenden zwei Gruppen einordnen:

- *Cluster-First–Schedule-Second (CF-SS)* Heuristik: Hier wird jede Fahrt zuerst einem Depot nach einem festgelegten Zuordnungsprinzip, wie z.B. das am nächsten gelegene Depot, zugeordnet (*clustering*). So wird das Mehrdepot-Problem in mehrere unabhängige Eindepot-Probleme zerlegt, die schnell gelöst werden können (*scheduling*). Über Lösungsansätze, die nach diesem Prinzip das MDVSP angehen, wird unter anderem bei [Carraresi and Gallo, 1984], [Lamatsch, 1988], [Mesquita and Paixão, 1992], [Dell'Amico et al., 1993], [Branco et al., 1995], [Löbel, 1997] und [Larsen and Madsen, 1997] berichtet.
- *Schedule-First-Cluster-Second (SF-CS)* Heuristik: Hier werden die unterschiedlichen Depots zuerst ignoriert und das gesamte Problem als ein großes Eindepot-Problem gelöst (*scheduling*). Danach wird jedem resultierenden Umlauf ein Depot zugeordnet (*clustering*). Diese Vorgehensweise wird unter anderem von [Bodin and Golden, 1981], [Assad et al., 1983], [Carraresi and Gallo, 1984], [Daduna and Mojsilovic, 1988], [Dell'Amico et al., 1993], [Daduna and Paixão, 1995] und [Grötschel et al., 1997] angewandt.

Einige Autoren schlagen außerdem vor, die resultierende heuristische Lösung mit einer *Reschedule-Heuristik* zu verbessern, indem SF-CS und CF-SS abwechselnd (mehrfach) nacheinander ausgeführt werden (siehe z.B. [Dell'Amico et al., 1993], [Löbel, 1997] und [Grötschel et al., 1997]).

In [Gintner et al., 2005a] und [Kliewer, 2005] wird ein weiterer heuristischer Ansatz für große praxisbezogene Probleminstanzen präsentiert. Zuerst werden aus dem Originalproblem mehrere vereinfachte Probleme nach unterschiedlichen Kriterien gebildet. Das können zum Beispiel Probleme mit Berücksichtigung nur eines bzw. weniger Depots, nur eines Fahrzeugtyps oder durch Weglassen bestimmter Nebenbedingungen sein. Das Hauptmerkmal vereinfachter Probleme ist ihre einfachere Lösbarkeit. Alle neu gebildeten Probleme werden optimal gelöst und ihre Lösungen auf das Auftreten gleicher Fahrtmuster untersucht. Tritt eine Sequenz von Fahrgastfahrten als gleiches Muster in allen Lösungen auf, so werden die entsprechenden Verbindungen zwischen den betroffenen Fahrten im Originalproblem

fixiert. Anschließend wird das ursprüngliche Problem mit den fixierten Fahrtverbindungen als MDVSP gelöst. Die Autoren berichten, dass mit Hilfe dieser Methode große realistische Probleminstanzen aus der Praxis viel schneller als mit dem exakten Ansatz gelöst werden konnten. Dabei weicht die gefundene Lösung von dem bekannten Optimum nur sehr geringfügig ab.

Die meisten *exakten* Methoden zu MDVSP werden von [Fischetti et al., 2001] nach Art der genutzten Formulierung in drei Basisklassen unterteilt:

1. Eingüterfluss-Formulierung
2. Mehrgüterfluss-Formulierung
3. Set-Partitioning-Formulierung

Der erste exakte Algorithmus basiert auf dem Modell der ersten Klasse. [Carpaneto et al., 1989] formulieren MDVSP als ein Zuordnungsproblem mit zusätzlichen pfadorientierten Flusserhaltungsbedingungen. Die untere Schranke der MIP-Formulierung wird mit Hilfe der sogenannten *Additive-Lower-Bounding*-Methode (siehe [Fischetti and Toth, 1988]) berechnet. Anschließend wird ein *Branch-and-Bound*-Verfahren mit einer speziellen Verzweigungsstrategie eingesetzt, um eine ganzzahlige Lösung des MDVSP zu erhalten. Der Nachteil dieser Formulierung ist die Schwäche der LP-Relaxation. Ein Jahrzehnt später „verschärfen" [Fischetti et al., 2001] diese Formulierung durch das Hinzufügen von speziellen *Pfad-Eliminierung-Bedingungen* und binden diese in einen *Branch-and-Cut*-Algorithmus ein. Ihr Verfahren zeigt vielversprechende Ergebnisse, besonders bei großen realistischen Instanzen mit einer großen Anzahl von Fahrgastfahrten pro Fahrzeug.

Formulierungen des MDVSP als *Mehrgüterfluss-Problem* zeichnen sich durch eine schärfere LP-Relaxation als Formulierungen der ersten Klasse aus (siehe [Mesquita and Paixão, 1999]). [Forbes et al., 1994] schlägt ein exaktes Verfahren vor, das das MDVSP in drei Schritten löst. Zuerst wird das Originalproblem in ein SDVSP relaxiert und mit dem *Netzwerk-Simplex-Algorithmus* gelöst. Im zweiten Schritt wird die optimale Lösung des relaxierten Problems dazu genutzt, um eine dual zulässige Basis für die LP-Relaxation zu schaffen, die dann mit dem *Dual-Simplex-Algorithmus* optimal gelöst wird. Im dritten Schritt des Verfahrens wird eine optimale ganzzahlige Lösung mit Hilfe des *Branch-and-Bound*-Ansatzes errechnet.

[Mesquita and Paixão, 1999] untersuchten die beiden verbreiteten Arten der MDVSP-Formulierung als Mehrgüterfluss-Problem, mit Zuordnungs- und Flussvariablen und nur mit Flussvariablen. Sie bewiesen, dass ihre LP-Relaxationen äquivalent sind und schärfer als LP-Relaxation der *Eingüterfluss-Formulierung* sind.

In [Löbel and Strubbe, 1996] wird das als *Mehrgüterfluss-Problem* formulierte MDVSP durch ein *Schnittebenenverfahren* in Verbindung mit einem *Branch-and-Bound*-Algorithmus optimal gelöst. Dabei wird die *Column-Generation*-Methode mit speziellen *Pricing-Strategien* benutzt. [Grötschel et al., 1997] verbessern diese Lösungsmethode und bezeichnen die neue Variante als *Branch-and-Cut*-Verfahren.

[Löbel, 1997, Löbel, 1999] kombiniert mehrere Lösungsmethoden miteinander. Zuerst wird mit Hilfe der *Langrange-Relaxation* eine untere Schranke für die minimale Flottengröße und die minimalen betrieblichen Kosten ermittelt. Außerdem wird eine zulässige Lösung (obere Schranke) von MDVSP durch eine der beiden Heuristiken *CF-SS* oder *Schedule-Cluster-Reschedule* berechnet. Danach wird die LP-Relaxation durch *Column-Generation* und *Column-Elimination* gelöst, wobei ein neues *Lagrange-Pricing* vorgeschlagen wird. Optional kann während des Prozesses versucht werden, die aktuelle ganzzahlige Lösung mit Hilfe einer Rundungsheuristik *(LP-Plunging)* zu verbessern. Dabei werden die nicht ganzzahligen Werte des LP-Lösungsvektors iterativ gerundet und fixiert und die neue LP-Relaxation erneut gelöst, bis das Problem unzulässig wird oder eine ganzzahlige Lösung gefunden werden kann. Ist die neue ganzzahlige Lösung aus praktischer Sicht gut genug, wird der Algorithmus abgebrochen, ansonsten wird mit *Column-Generation* fortgeführt. Anschließend wird die resultierende LP-Relaxation mit *Branch-and-Cut*-Verfahren bis zur bewiesenen Optimalität gelöst (vorausgesetzt, dass alle Spalten mit negativen reduzierten Kosten generiert werden konnten). Der Autor berichtet über das Lösen von besonders großen Probleminstanzen aus der Praxis.

[Haghani and Banihashemi, 2002] erweitern das Mehrgüterfluss-Modell, indem sie alle Fahrgastfahrten in drei Mengen, nämlich Morgen-, Mittags- und Nachmittagsfahrten, unterteilen. Dabei können die Nachmittagsfahrten nicht direkt nach Morgenfahrten, sondern nur über eine Zwischenfahrt zum Depot und zurück bedient werden. Somit existieren auch keine direkten Verbindungskanten zwischen den Fahrten dieser zwei Mengen, was zu einer enormen Reduzierung der Anzahl der Kanten bzw. Variablen führt. Außerdem erweitern die Autoren das MDVSP um die Beschränkung der maximalen Umlaufdauer. Zur Lösung des Problems schlagen sie einen *Constraints-Generation*-Ansatz vor, in dem die Umlaufdauer-Bedingungen iterativ zu dem Problem hinzugefügt werden.

[Huisman, 2004] präsentiert eine alternative Mehrgüterfluss-Formulierung für MDVSP. Die Hauptidee stammt ursprünglich von [Haase et al., 2001] im Kontext der integrierten Umlauf- und Dienstplanung. Der Hauptunterschied zu den klassischen Mehrgüterfluss-Modellen besteht darin, dass im Netzwerk nur direkte Verbindungskanten und keine Über-Depot-Verbindungen existieren. Damit kann die Anzahl der Verbindungskanten und somit die Anzahl der Variablen drastisch reduziert werden. Allerdings führt das dazu, dass ein Pfad von der Quelle zur Senke

nicht mehr den kompletten Umlauf, sondern ein Umlaufstück präsentiert und die Betrachtung der fixen Fahrzeugkosten durch die Kosten auf den Depotkanten nicht mehr möglich ist. Um aber sowohl Fixkosten als auch Depotkapazitäten weiterhin berücksichtigen zu können, fügt der Autor zu der abgeleiteten mathematischen Formulierung zusätzliche Nebenbedingungen hinzu, die die Anzahl der Fahrzeuge zählen, die zu einem gegebenen Zeitpunkt unterwegs sind. Diese Anzahl, multipliziert mit den fixen Fahrzeugkosten, wird in die Zielfunktion aufgenommen. Der Autor behauptet, dass trotz der großen Anzahl an zusätzlichen Nebenbedingungen diese Formulierung schneller gelöst werden kann, gibt aber keine genaueren Angaben dazu.

Modelle der dritten Klasse basieren auf einer *Set-Partitioning-Formulierung* und zeichnen sich durch eine sehr große, üblicherweise exponentiell wachsende Anzahl an binären Entscheidungsvariablen aus, die alle möglichen Umläufe repräsentieren. Das Ziel ist aus der Menge aller möglichen Umläufe eine kostenminimale Teilmenge auszuwählen, die alle Fahrgastfahrten genau einmal überdeckt. Wegen der sehr großen Anzahl der möglichen Umläufe wird in Verbindung mit der *Set-Partitioning-Formulierung* oft die *Column-Generation*-Methode verwendet, die die nötigen Spalten (Umläufe) dynamisch erzeugt und dem Modell hinzufügt. Allerdings ist die Wahl neuer Spalten (*Pricing-Problem*) nicht eindeutig, sodass die Lösung der LP-Relaxation (*Master-Problem*) üblicherweise stark degeneriert ist. Das kann zu einer langen Sequenz von Iterationen führen, in denen neue Spalten dem Modell zwar hinzugefügt werden, beeinflussen aber nicht effektiv die Qualität der Lösung der LP-Relaxation. Das eingeschränkte Master-Problem wird durch die große Anzahl der Spalten immer schwieriger zu lösen sein und der Gesamtprozess wird immer langsamer.

[Lamatsch, 1988] diskutierte die Verwendung des *Dantzig-Wolfe-Dekompositionsprinzips* für seine MIP-Formulierung und gab eine Set-Partitioning-Formulierung für MDVSP ohne Betrachtung der Depotkapazitäten an. Allerdings verwirft der Autor diese Methode, da sie wegen der mangelnden Rechenleistung von Computer damaliger Zeit für einigermaßen realistische Probleminstanzen nicht anwendbar war.

Der erste Bericht über den Einsatz und Tests dieser Formulierung stammt von [Ribeiro and Soumis, 1994]. Das MDVSP wird durch die Anwendung der *Dantzig-Wolfe-Dekomposition* als *Set-Partitioning-Problem* mit zusätzlichen Nebenbedingungen für Depotkapazitäten formuliert. Sie lösen die LP-Relaxation mit einem *Column-Generation*-Verfahren, wobei zur Generierung neuer Spalten mit minimalen negativen reduzierten Kosten ein *Kürzeste-Wege-Problem* gelöst wird. Anschließend wird mit einem *Branch-and-Bound*-Algorithmus eine ganzzahlige Lö-

sung gesucht. Außerdem beweisen die Autoren, dass die LP-Relaxation dieser Modelle gleich der LP-Relaxation der *Mehrgüterfluss-Formulierung* für MDVSP ist.

Die *Set-Partitioning-Formulierung* für MDVSP wird auch in Publikationen von [Bianco et al., 1994], [Branco et al., 1995], [Desrosiers et al., 1995] und [Hadjar et al., 2006] diskutiert.

Fast alle vorgestellten Netzwerkmodelle besitzen eine gemeinsame Eigenschaft: Alle möglichen Fahrtenverknüpfungen werden explizit durch Kanten im Netzwerk repräsentiert. Sie werden auch *connection-basierte Modelle* genannt (siehe [Kliewer, 2005]). Da die Anzahl solcher Kanten quadratisch mit der Anzahl der Fahrten ansteigt, erreicht die Problemgröße bei großen praxisrelevanten Instanzen sehr schnell die kritische Grenze und lässt sich nicht mehr effizient exakt lösen. In der Literatur wurden zwar einige Ansätze vorgestellt, die die Kantenmenge durch unterschiedliche Heuristiken reduzieren, sie schränken aber dadurch den Lösungsraum des Problems ein, so dass die Optimalität der gefundenen Umlaufpläne nicht mehr garantiert werden kann.

[Mellouli and Kliewer, 2002] und [Kliewer et al., 2006] präsentieren eine Mehrgüterfluss-Formulierung für MDVSP, die von einem alternativen Netzwerkfluss-Modell abgeleitet wird. Die vorgeschlagene Modellierung als *Time-Space-Netzwerk* zeichnet sich dadurch aus, dass kompatible Fahrten nicht unmittelbar durch eine Verbindungskante, sondern implizit durch einen Netzwerkpfad verbunden sind. Die auf einer ähnlichen Idee basierende Modellierung ist zuvor bereits im Bereich der *Aircraft Rotation* im Flugverkehr eingesetzt worden (z.B. [Hane et al., 1995] oder auch später [Mercier et al., 2005]). Diese Technik wurde in [Mellouli and Kliewer, 2002] für das Busumlaufplanungsproblem angewandt. Der Ansatz ermöglicht eine Modellierung und Lösung von Umlaufplanungsaufgaben mit mehreren Depots, mehreren Fahrzeugtypen, mit Vorgaben von Fahrzeugtypgruppen für jede Fahrgastfahrt und von Kapazitäten verschiedener Art. Minimiert werden die Gesamtkosten des Umlaufplans, die aus den fixen Fahrzeugkosten (oder der Fahrzeuganzahl) und den operativen Einsatzzeit- und Kilometerkosten bestehen.

Dank der alternativen Modellierung und vorgeschlagenen Aggregationstechniken kann die Anzahl der Kanten drastisch reduziert werden, ohne den Lösungsraum zu beschränken. Somit wächst die Anzahl der Variablen annähernd linear[1] mit der Anzahl der Fahrgastfahrten. Dies führt zu einer Reduktion der Größe des mathematischen Modells, sodass es mit Hilfe mathematischer Optimierungssoftware gelöst werden kann. Die durchgeführten Tests zeigen sehr gute Ergebnisse für die-

[1] Die Anzahl der Variablen beläuft sich in Größenordnung $\mathcal{O}(nm)$, wobei n die Anzahl der Fahrgastfahrten und m die Anzahl der Endhaltestellen ist. Allerdings ist m typischerweise viel kleiner als n.

sen Lösungsansatz sowohl für kleine als auch für sehr große Probleminstanzen aus der Praxis (siehe [Kliewer, 2005]).

Diese Modellierung als Time-Space-Netzwerk liegt auch dem Modell zugrunde, das für die integrierte Umlauf- und Dienstplanung im Rahmen der vorliegenden Arbeit entwickelt wird.

4.1.2 Umlaufbasierte Dienstplanung

Das Problem der Dienstplanung wird in der Literatur sehr umfangreich behandelt. Die Veröffentlichungen aus diesem Bereich unterscheiden sich in der Komplexität der betrachteten Dienstregeln, in der Modellierung des Problems und der eingesetzten Algorithmen. Die weiter unten folgende Übersicht umfasst nicht nur Arbeiten aus dem ÖPNV-Bereich, sondern bezieht auch einige Veröffentlichungen aus dem verwandten Bereich der Flugplanung ein, da sie sowohl eine ähnliche Problemformulierung als auch ähnliche Lösungsansätze verwenden. Generell ist anzumerken, dass das Dienstplanungsproblem schon für den einfachen Fall, bei dem die Zulässigkeit der Dienste nur durch ihre Länge und Arbeitszeit begrenzt ist, ein $\mathcal{NP}$-schweres Problem darstellt (für Beweise siehe [Fischetti et al., 1987] und [Fischetti et al., 1989]).

Die meist verbreitete Formulierung des Dienstplanungsproblems ist die *Set-Partitioning-* bzw. *Set-Covering-Formulierung*. Dabei werden die zulässigen Dienste durch Spalten im mathematischen Modell dargestellt. Die Überdeckungsbedingungen stellen sicher, dass jedes Dienstelement in genau einem (Set-Partitioning-Problem) bzw. mindestens einem (Set-Covering-Problem) Dienst enthalten ist. Außerdem kann das mathematische Modell je nach Anforderungen globale Bedingungen beinhalten, die z.B. die Zusammensetzung der Diensttypen in der Lösung regeln. Für eine Übersicht über die wichtigsten Algorithmen für das Set-Covering-Problem wird auf [Caprara et al., 2000] verwiesen.

Da die Anzahl aller zulässigen Dienste schon bei kleinen Probleminstanzen mehrere Millionen erreichen kann, ist eine direkte Behandlung solcher mathematischen Probleme nicht möglich. Dafür wird oft das *Column-Generation*-Verfahren eingesetzt, bei dem immer eine relativ kleine Auswahl von Spalten (Diensten) explizit betrachtet wird und alle anderen implizit betrachtet werden (siehe Abschnitt 3.3 für eine detailliere Erklärung des Verfahrens). So müssen sämtliche Dienstregeln nur im *Pricing-Problem* berücksichtigt werden. Das *Master-Problem* kann mit Hilfe der *LP-Relaxation* (z.B. [Desrochers and Soumis, 1989], [Falkner and Ryan, 1992], [Desrochers et al., 1992]) oder *Lagrange-Relaxation* (z.B. [Carraresi et al., 1995], [Freling, 1997], [Huisman, 2004], [Borndörfer et al., 2003, Borndörfer et al., 2005]) gelöst werden. Das Pricing-Problem wird oft als ein *ressourcenbeschränk-*

tes Kürzeste-Wege-Problem formuliert und mit Hilfe der *Dynamischen Programmierung* gelöst. [Freling, 1997] und [Huisman, 2004] schlagen außerdem vor, die zulässigen Dienste durch die Aufzählung zulässiger Kombinationen aus den zuvor erzeugten Dienststücken zu bestimmen. Allerdings empfehlen sie diese Vorgehensweise aus kombinatorischen Gründen nur bei Dienstarten, die aus maximal drei Dienststücken bestehen.

Zur Bestimmung einer ganzzahligen Lösung wird in den meisten Verfahren der *Branch-and-Bound*-Ansatz verwendet. Dabei wird Column-Generation nur im ersten Knoten des B&B-Baumes eingesetzt, d.h. in der weiter folgenden Verzweigung bzw. Formulierung der Subprobleme werden keine neuen Spalten erzeugt. Um aber eine optimale Lösung zu bekommen, muss die Generierung neuer Dienste in das B&B-Verfahren integriert werden. Diese Vorgehensweise wird *Branch-and-Price* genannt (siehe z.B. [Barnhart et al., 1998]). Werden dabei auch zuvor ignorierte und jetzt verletzte Nebenbedingungen dem Problem hinzugefügt (allein als *Branch-and-Cut*-Ansatz bekannt), dann wird solch ein Verfahren *Branch-and-Cut-and-Price* genannt (siehe z.B. [Dallaire et al., 2004]). [Elhallaoui et al., 2005] schlagen einen alternativen Ansatz vor. Sie aggregieren die Nebenbedingungen nach bestimmten Kriterien dynamisch, anstatt sie erst zu ignorieren und dann bei der Verletzung hinzuzufügen. Während des Lösungsprozesses wird die Aggregation iterativ angepasst, sodass die Optimalität der Lösung garantiert werden kann.

[Borndörfer et al., 2003] verwenden einen *Branch-and-Generate*-Ansatz (heuristisches Branch-and-Price). Sie lösen das Master-Problem mit Hilfe einer *Coordinate-Ascent*-Heuristik, während die Pricing-Phase als ein ressourcenbeschränktes Kürzeste-Wege-Problem formuliert und ebenfalls heuristisch gelöst wird. Um eine ganzzahlige Lösung zu bekommen, wird eine *LP-Plunging*-Heuristik eingesetzt. Dabei wird zunächst für jede Variable eine Bewertungszahl (*score*) berechnet und mit deren Hilfe eine Kandidatenliste bestimmt. Danach werden Variablen aus dieser Liste mit Hilfe einer *Lagrange-Probing*-Methode iterativ fixiert.

[Desaulniers et al., 2001] präsentieren verschiedene Techniken zur Beschleunigung des Column-Generation-Verfahrens. [du Merle et al., 1999] und [Ben Amor et al., 2004] behandeln die Probleme starker Degeneration von primalen und dualen Lösungen, die zu einer schlechten Konvergenz des Prozesses und Instabilitäten der dualen Variablen führen.

Einige Lösungsansätze gehen von einer a priori gegebenen Menge aller möglichen Dienste und bestimmen daraus direkt, ohne Column-Generation, eine minimale (exakte oder heuristische) Überdeckung. [Hoffman and Padberg, 1993] beschreiben einen *Branch-and-Cut*-Algorithmus, der durch das Hinzufügen spezieller Schnittebenen (*cutting planes*), die von der zugrundeliegenden Struktur des Polytops abgeleitet werden, in der Lage ist, mittlere Set-Partitioning-Probleme bis zu

bewiesener Optimalität zu lösen.

[Wedelin, 1995] betrachtet das Set-Partitioning-Problem in direkter Weise. Beim Lösen der Lagrange-Relaxation des Problems versucht er durch eine leichte, approximative Manipulation des Kostenvektors solche Multiplikatoren zu finden, bei denen keine Variable die reduzierten Kosten gleich Null hat. Die somit eindeutige ganzzahlige duale Lösung liefert eine zulässige ganzzahlige primale Lösung des Problems, die iterativ verbessert wird.

[Beasley and Cao, 1998] formulieren das Dienstplanungsproblem als ein Graphenproblem und lösen es mit Hilfe der *Dynamischen Programmierung* zusammen mit einer Lagrange-basierten Bestrafungsprozedur, die in die Baumsuche eingebunden ist. Allerdings berücksichtigen sie nur einfache Dienstregeln (nur die maximale Arbeitszeit).

Sowohl [Ceria et al., 1998] als auch [Caprara et al., 1999] beschreiben eine Lagrange-basierte Heuristik zur Lösung des Set-Covering-Problems. Beide Ansätze verwenden die duale Information der Spalten, um mit Hilfe einer Greedy-Heuristik eine ganzzahlige Lösung zu bestimmen, die durch ein systematisches Fixieren der Variablen iterativ verbessert wird.

[Mingozzi et al., 1999] kombinieren unterschiedliche *Bounding*-Prozeduren, um eine duale Lösung der LP-Relaxation des Set-Partitioning-Problems zu berechnen. Mit den dualen Informationen reduzieren sie die Anzahl von primalen Variablen, sodass das resultierende Problem mit Hilfe eines Branch-and-Bound-Algorithmus gelöst werden kann.

Eine komplett andere Formulierung des Dienstplanungsproblems wird von [Fischetti et al., 2001] benutzt. Sie formulieren es als *Eingüterfluss-Problem*, was allerdings nur dank der Betrachtung von ganz einfachen Dienstregeln (nur Dienstlänge und Arbeitszeit) möglich ist. Im vorgeschlagenen *Branch-and-Cut*-Ansatz verschärfen sie die relativ schwache LP-Relaxation durch das Hinzufügen verschiedener *Cuts*.

[Banihashemi and Haghani, 2001] formulieren das Problem der Dienstplanung als *Mehrgüterfluss-Problem*. Die Güter im Netzwerk repräsentieren unterschiedliche Diensttypen, wobei alle Dienstregeln zuerst außer Acht gelassen werden. In der ersten Phase wird das Mehrgüterfluss-Problem optimal gelöst, was eine Menge von nicht unbedingt zulässigen Diensten liefert. In der zweiten Phase werden die unzulässigen Dienste durch zusätzliche Restriktionen im mathematischen Modell verboten bzw. die schlechten Dienste mit Strafkosten versehen. Danach wird das MIP-Modell nochmal gelöst. Der Prozess wird solange wiederholt, bis der Dienstplan nur zulässige Dienste enthält.

Auch Metaheuristiken, die in der letzten Zeit immer mehr Verbreitung finden,

werden zum Lösen des Dienstplanungsproblems eingesetzt. Beispiele dafür sollen nun kurz genannt werden:

- *Genetische Algorithmen* werden u.a. von [Wren and Wren, 1995], [Kwan and Wren, 1996], [Clement and Wren, 1995], [Beasley and Chu, 1996], [Kwan et al., 1999], [Marchiori and Steenbeek, 2000], [Kwan et al., 2001] verwendet.
- *Tabu Search* wird beispielsweise bei [Cavique et al., 1999] und [Shen and Kwan, 2001] eingesetzt.
- Der Ansatz von [Forsyth and Wren, 1997] basiert auf *Ant Colony Optimization.*
- [Lourenco et al., 2001] erweitern das Dienstplanungsproblem um weitere Zielsetzungen, wie z.B. Qualität des Services, und lösen es mit einer Variante von *Multiobjective Tabu Search* und *Multiobjective Genetic Algorithm.*
- [Li and Kwan, 2003, Li and Kwan, 2005] betrachten ebenfalls mehrere Zielsetzungen und greifen zum ersten Mal zur *Fuzzy-Theorie* in Kombination mit genetischen Algorithmen, um das Dienstplanungsproblem zu lösen.

[Yunes et al., 2005] setzen Methoden des *Constraint-Programming (CP)* ein und kombinieren sie mit einem Column-Generation-Verfahren. Sie formulieren und lösen das Pricing-Problem mit Hilfe von CP, während das Master-Problem mit mathematischer Optimierung gelöst wird. Die Sprache des CP erlaubt es, viele komplizierten Dienstregeln relativ einfach zu formulieren. Somit werden alle zulässigen Dienste implizit berücksichtigt. Der zweite Vorteil bei der Benutzung der CP-Techniken im Pricing-Problem besteht darin, dass sie sehr schnell eine zulässige Lösung liefern, d.h. neue Dienste mit negativen reduzierten Kosten (aber nicht unbedingt kleinsten reduzierten Kosten). Eine ähnliche Vorgehensweise wird von [Fahle et al., 2002] im Bereich von Crew Rostering angewandt.

Für eine Übersicht über den aktuellen Forschungsstand zur Dienstplanung im Flugverkehr wird auf [Barnhart et al., 2003] und [Gopalakrishnan and Johnson, 2005] verwiesen.

4.2 Fahrplanbasierte Dienstplanung

Bei der (traditionellen) umlaufbasierten Dienstplanung können Dienstelemente nur dann zu einem Dienststück aneinandergereiht werden, wenn sie dem gleichen Umlauf zugeordnet sind. Bei der fahrplanbasierten Dienstplanung sind die Fahrzeugumläufe dagegen nicht bekannt, somit ergeben sich viel mehr Freiheitsgrade bei

der Bildung von Dienststücken und Diensten. Die Möglichkeit zwei Dienstelemente zu einem Dienststück aneinanderzureihen ist nur durch zeitliche und räumliche Gegebenheiten begrenzt.

[Freling, 1997,Freling et al., 1999] und [Huisman, 2004] behandeln das fahrplanbasierte Dienstplanungsproblem für sich, d.h. ohne eine nachträgliche Bestimmung der Umläufe. Die Lösung dieses Problems ist von keiner praktischen Bedeutung, allerdings kann sie als untere Schranke für die Anzahl der Dienste bzw. Personalkosten verwendet werden. Zusammen mit einer optimalen Lösung der Umlaufplanung kann sie auch als eine untere Schranke für die integrierte Umlauf- und Dienstplanung angesehen werden. Die Differenz zwischen dieser Schranke und einer Lösung der sequenziellen Umlauf- und Dienstplanung wird bei [Freling, 1997] als Maß für eine potenzielle Einsparung durch die Integration verwendet. Unterscheiden sich die beiden Werte für eine Probleminstanz nur gering, ist der Einsatz eines viel aufwendigeren und zeitintensiveren, integrierten Verfahrens für sie nicht lohnenswert.

Beide formulieren das fahrplanbasierte Dienstplanungsproblem ähnlich zu der umlaufbasierten Variante, nämlich als Set-Covering-Problem. Der Unterschied besteht darin, dass die Überdeckungsbedingungen nur für die Dienstelemente formuliert werden, die direkt von Fahrgastfahrten der Fahrplanmasse abgeleitet wurden. Im Gegensatz dazu sind sie beim umlaufbasierten Dienstplanungsproblem für alle Dienstelemente inklusive der Leer-, Depotfahrten sowie Wartezeit definiert. Eine weitere Eigenschaft der fahrplanbasierten Dienstplanung besteht darin, dass die Menge zulässiger Dienste viel größer ist, da bei ihrer Konstruktion viel mehr Freiheitsgrade zur Verfügung stehen.

Zur Lösung des Set-Covering-Problems setzen die beiden Wissenschaftler ein *Column-Generation*-Verfahren in Kombination mit *Lagrange-Relaxation* ein. Es startet mit einer Initialmenge der Dienste, die sich aus einer umlaufbasierten Lösung ergeben. Das *Dual-Lagrange-Problem* wird mit einem Subgradienten-Algorithmus approximativ gelöst. Die suboptimalen Lagrange-Multiplikatoren aus dem Master-Problem werden dazu genutzt, um in dem Pricing-Schritt neue Dienste mit negativen reduzierten Kosten zu generieren. Dafür stellt Freling eine 2-Phasen-Prozedur vor. In der ersten Phase wird mit Hilfe eines speziellen *Dienststückerzeugungs-Netzwerkes* eine Menge von Dienststücken erzeugt, aus denen in der zweiten Phase potenzielle Dienste erstellt werden können. Eine zulässige Lösung wird mit der bereits erwähnten Heuristik von [Caprara et al., 1999] bestimmt. Dabei gehen sie davon aus, dass aus der Set-Covering-Lösung wie bei der traditionellen Dienstplanung nahezu problemlos eine Set-Partitioning-Lösung erzeugt werden kann.

Das fahrplanbasierte Dienstplanungsproblem ist ein Bestandteil der im Rahmen dieser Arbeit entwickelten Fix-and-Optimize Heuristik (siehe Kapitel 7).

4.3 Integrierte Umlauf- und Dienstplanung

In diesem Abschnitt werden die wichtigsten Veröffentlichungen zusammengefasst, die die Umlauf- und Dienstplanung partiell bzw. komplett integriert in einem Modell behandeln. Besonders ausführlich wird auf die Dissertation von Huisman [Huisman, 2004] eingegangen, da er als Erster den allgemeinen Fall mit mehreren Depots behandelte. Seine Arbeit dient als Referenz und als Messlatte für Modelle und Lösungsverfahren, die im Rahmen dieser Arbeit entwickelt wurden. Außerdem wird die Behandlung großer Probleme diskutiert. Die Literaturübersicht beschränkt sich nicht nur auf ÖPNV-Bereich, sondern beinhaltet auch einige Arbeiten zur Integration der äquivalenten Planungsphasen im Flugverkehr.

4.3.1 Teilintegration der Umlauf- und Dienstplanung

Optimierungsansätze zur integrierten Umlauf- und Dienstplanung werden seit über 20 Jahren untersucht. Die ersten integrierten Modelle wurden von [Ball et al., 1983] und [Patrikalakis and Xerocostas, 1992] vorgeschlagen. Allerdings war ihre direkte Lösung mit dem damaligen Stand der Computertechnik und Optimierungsmethoden noch nicht möglich. Deshalb basieren die ersten Ansätze auf mehrstufigen heuristischen Methoden, die die beiden Planungsprobleme zwar immer noch sequenziell behandeln, allerdings eine bessere Qualität und Kompatibilität von Umlauf- und Dienstplänen anstreben. Wir nennen solche Vorgehensweise *partiell integrierte Umlauf- und Dienstplanung* (engl.: *partially integrated vehicle und crew scheduling*). Die meisten partiell integrierten Lösungsansätze lassen sich in eine der zwei folgenden Kategorien einordnen (vgl. [Freling, 1997], S. 104ff):

- Dienste werden direkt auf der Fahrplanmasse konstruiert. Dabei werden die Anforderungen an den Dienstplan so modifiziert, dass ein gültiger Umlaufplan im Nachhinein gefunden werden kann.
- Umläufe werden unter Beachtung bestimmter Dienstregeln verplant, sodass sie „dienstplantauglicher" sind und der im Nachhinein berechnete Dienstplan von besserer Qualität bzw. gültig ist.

Die meisten Lösungsansätze sind von der ersten Kategorie und basieren auf der heuristischen Methode von [Ball et al., 1983]. Das integrierte Modell basiert auf einem Planungsgraphen, in dem Dienstelemente (von den Autoren *d-trips* genannt) als Knoten repräsentiert werden. Zusätzlich werden zwei weitere Knoten s und t erzeugt, die das Depot abbilden[2]. Kanten im Planungsgraphen verbinden kompatible Knoten miteinander und repräsentieren entsprechende Aktivitäten, wie z.B.

[2] anzumerken ist, dass sie von einem Planungsproblem mit nur einem Depot ausgehen

Warten, direkte Leerfahrt, Leerfahrt über Depot oder aber auch einfache Nacheinanderausführung von zugehörigen Dienstelementen. Weiterhin existieren für jeden Knoten zusätzliche Depot-Kanten, die ihn mit s und t verbinden. Da alle Dienstelemente direkt von den Fahrgastfahrten der Fahrplanmasse abgeleitet werden, muss jeder Knoten sowohl mit einem Fahrzeug als auch einem Fahrer bedient werden. Die Kanten werden dagegen entsprechend ihrer zu repräsentierenden Aktivität in zwei Kategorien aufgeteilt, und zwar in Kanten, die sowohl ein Fahrzeug als auch einen Fahrer benötigen (*Umlauf-Dienst-Kanten*) und Kanten, die reine Fahreraktivitäten ohne Fahrzeug (z.B. Fußweg, Warten, Mitfahrt als Fahrgast) darstellen (*Dienst-Kanten*). Dienste und Umläufe sind im Graphen durch Pfade von s nach t abgebildet (je nachdem, welche Kanten im Pfad enthalten sind), wobei nicht jeder Pfad gültig ist. Die Menge aller Umlauf-Pfade (Dienst-Pfade), die jeden Knoten genau einmal überdecken, bildet einen Umlaufplan (einen Dienstplan). Die Autoren formulieren das integrierte Problem als ein Problem des Findens je einer Menge von Umlauf- und Dienstpfaden, die jeweils alle Knoten überdecken und zueinander kompatibel sind. Die Pläne sind kompatibel, wenn eine Umlauf-Dienst-Kante genau dann in einem Dienst enthalten ist, wenn sie durch einen Umlauf überdeckt ist (*konsistente Überdeckung*). Dieses formal definierte Problem konnten die Autoren allerdings nicht direkt lösen und schlagen stattdessen eine heuristische Methode vor, die einen gültigen Dienstplan konstruiert und davon einen Umlaufplan ableitet. Der Lösungsansatz besteht aus drei Phasen. Zuerst wird mit Hilfe einer iterativen Zuordnungsheuristik eine Menge von Dienststücken erzeugt, die alle Knoten abdeckt und die maximal zugelassene Dauer jeweils nicht überschreitet. In der zweiten Phase wird versucht, diese Dienststücke zu verbessern, indem man sie umformt bzw. zusammenfasst. Aus den resultierenden Dienststücken wird im letzten, dritten Schritt mit Hilfe einer Zuordnungsheuristik eine Menge von Diensten konstruiert. Als Zielfunktion wird dabei die Minimierung der Gesamtkosten aller Dienste angestrebt. Die Art und Weise, wie die Dienststücke in der ersten Phase aus dem Planungsgraphen erzeugt werden, stellt sicher, dass aus einem Dienstplan im Nachhinein ein gültiger Umlaufplan erzeugt werden kann. Das geschieht durch das Löschen reiner Dienst-Kanten und das Fixieren sonstiger Kanten aus den resultierenden Dienststücken.

[Tosini and Vercellis, 1988] behandeln das Umlauf- und Dienstplanungsproblem im Regionalverkehr ebenfalls partiell integriert. Als Ablösepunkte werden nur Depots akzeptiert. Sie schlagen eine heuristische Methode vor, die erst ein modifiziertes Dienstplanungsproblem löst und anschließend aus dem resultierenden Dienstplan einen Umlaufplan ableitet. Die Menge gültiger Dienste wird mit einer ähnlich Idee wie in [Ball et al., 1983] durch das Lösen einer Sequenz von Zuordnungsproblemen bestimmt. Allerdings erweitern sie diese Methode um die Berücksichtigung mehrerer Depots und Einführung einer stochastischen Komponente, die

mehrere besten Zuordnungen in jeder Stufe zulässt. Aus der resultierenden Menge gültiger Dienste wird mit Hilfe einer Greedy-Heuristik unter Beachtung zusätzlicher Nebenbedingungen eine kostenminimale Überdeckung aller Dienstelemente bestimmt. Ein gültiger Umlaufplan wird im Nachhinein aus den Dienststücken ausgewählter Dienste mit einem Algorithmus zur Berechnung kostenminimaler Flüsse bestimmt.

[Falkner and Ryan, 1992] und [Patrikalakis and Xerocostas, 1992] behandeln die Umlauf- und Dienstplanung ebenfalls mit einer Methode aus der ersten Kategorie. Dabei nutzen sie einen Dienstplanungsgraphen, der dem in [Ball et al., 1983] ähnlich ist. Patrikalakis und Xerocostas präsentieren eine 3-Phasen-Prozedur zum Finden guter Umlauf- und Dienstpläne. In der ersten Phase wird das Dienstplanungsproblem heuristisch gelöst und eine kostenminimale Menge von Diensten bestimmt, die alle Dienstelemente genau einmal überdecken. Dabei werden bestimmte Diensteigenschaften, wie z.B. Start- und Endzeit, nur approximativ berechnet, da zu diesem Zeitpunkt noch keine Fahrzeugaktivitäten bekannt sind. Außerdem kann die Problemgröße für große Instanzen durch die Eliminierung einiger Ablösepunkte sowie Aufteilung des Gesamtproblems in unabhängige Teilprobleme heuristisch verkleinert werden. Für die resultierenden Dienste werden in der zweiten Phase kompatible Umläufe durch das Lösen eines kostenminimalen Flussproblems berechnet. Die nötige Kompatibilität wird dadurch sichergestellt, dass alle Fahreraktivitäten immer mit einem Fahrzeug abgedeckt werden. In der dritten Phase wird das Dienstplanungsproblem erneut auf Basis der Umläufe gelöst. Dafür wird ein modifizierter Dienstplanungsgraph aufgebaut, der für jeden Dienst aus der ursprünglichen Lösung einige alternativen Dienstkonstruktionen implizit berücksichtigt.

Ein Lösungsansatz aus der zweiten Kategorie wird von [Darby-Dowman et al., 1988] als interaktiver Teil eines Entscheidungsunterstützungssystems eingesetzt. [Scott, 1985] erweitert ebenfalls das Umlaufplanungsproblem um die Berücksichtigung der Personalkosten. Der ursprüngliche Umlaufplan wird dann heuristisch, entsprechend der geschätzten marginalen Kosten, die mit kleinen Änderungen im aktuellen Umlaufplan assoziiert sind, modifiziert. Die geschätzten marginalen Kosten werden durch das Lösen des dualen Problems von dem *HASTUS*-Dienstplanungsmodell (siehe [Rousseau and Blais, 1985]) bestimmt.

[Borndörfer et al., 2002] diskutieren weitere Ideen zur partiellen Integration. Sie bezeichnen sie als *Hilfsregeln zur internen Integration in der sequenziellen Planung*. Als Beispiel nennen sie Kostenmanipulationen in der Umlaufplanung. Dabei wird die Kostenstruktur bei der Umlaufplanung so geändert, dass die Konstruktion *„dienstplantauglicher"* Umläufe bevorzugt wird. Typische Manipulation der Umlaufkosten ist eine künstliche Verbilligung von Aus- bzw. Einrückfahrten von

bzw. zu einem Depot oder Bestrafung der Verknüpfungen zwischen zwei langen oder abgelegenen Fahrten etc. Eine weitere Möglichkeit partieller Integration ist die Planung von Umläufen mit Längenrestriktionen. Die Länge der Umläufe wird künstlich auf die maximale Dienstdauer beschränkt. Somit vergrößert sich die Wahrscheinlichkeit, dass jeder Umlauf mit einem Dienst besetzt werden kann.

[Kliewer, 2005] berichtet in ihrem Ansatz über die Möglichkeit einige Aspekte der Dienstplanung für die Umlaufplanung zu berücksichtigen. Dank der speziellen Netzwerkstruktur liefert der Algorithmus anstatt nur einer optimalen Lösung des Umlaufplanungsproblems ein Bündel davon. Diese Eigenschaft kann ausgenutzt werden, um aus den vorliegenden Umlaufplänen denjenigen auszuwählen, der zusätzliche Anforderungen aus der Dienstplanung erfüllt. Als eine solche mögliche Anforderung wird in [Kliewer, 2005] die Anzahl der Linienwechsel pro Umlauf diskutiert.

4.3.2 Vollständige Integration mit einem Depot

Die ersten Lösungsansätze zur vollständigen Integration von Umlauf- und Dienstplanung wurden erst Mitte der 90-ger Jahre veröffentlicht. [Patrikalakis and Xerocostas, 1992] beschreiben zwar eine mathematische Formulierung für das vollständig integrierte Modell, weisen aber darauf hin, dass es direkt nicht gelöst werden kann. Stattdessen benutzt sie zur Lösung eine partiell integrierte Vorgehensweise.

Das integrierte Umlauf- und Dienstplanungsproblem für ein Depot wurde intensiv von Freling untersucht. In seiner Dissertationsarbeit ([Freling, 1997]) bietet er eine ausführliche Beschreibung entwickelter Modelle und Lösungsansätze. Das mathematische Modell besteht aus drei Komponenten: ein Quasi-Assignment-Teil für die Umlaufplanung, Set-Partitioning-Bedingungen für die Dienstplanung und eine Menge von Kopplungsbedingungen, die eine korrekte Kompatibilität zwischen Umlauf- und Dienstplänen sicherstellen. Sowohl Umläufe als auch Dienste sind im Modell explizit abgebildet. Zur Lösung dieser Formulierung entwickelt Freling einen Algorithmus, der auf einer Kombination von Column-Generation und Lagrange-Relaxation basiert. Im eingeschränkten Master-Problem werden die Kopplungsbedingungen mit Hilfe der Lagrange-Relaxation relaxiert. Das resultierende Problem zerfällt in ein Eindepot-Umlaufplanungsproblem, das mit einer modifizierten Version des *Vorwärts-Rückwärts-Auktionsalgorithmus* gelöst wird, und ein triviales Auswahlproblem, in dem nur die Dienste gewählt werden, die negative reduzierte Kosten aufweisen. Neue Spalten (Dienste), die dem eingeschränkten Master-Problem hinzugefügt werden, konstruiert Freling in einer 2-Phasen-Prozedur. Im ersten Schritt wird eine Menge von gültigen Dienststücken durch das Finden kürzester Wegen zwischen allen kompatiblen Knotenpaaren auf einem

speziellen Dienststückerzeugung-Graphen erzeugt. Die Kosten auf den Kanten in diesem Graphen sind so definiert, dass die Kosten eines Pfades zwischen zwei Knoten den reduzierten Kosten des durch diesen Pfad repräsentierten Dienststückes entsprechen. In der zweiten Phase wird aus den so konstruierten Dienststücken eine Menge gültiger Dienste mit negativen reduzierten Kosten erstellt. Dafür schlägt Freling unterschiedliche Algorithmen vor, wie z.B. eine einfache Aufzählung von gültigen Dienststückkombinationen oder die Modellierung als ein ressourcenbeschränktes Kürzeste-Wege-Problem, das mit Hilfe Dynamischer Programmierung gelöst wird. Der Autor empfiehlt die erste Methode für Dienstarten, die aus zwei bis drei Dienststücken bestehen können. Zur Berechnung einer zulässigen Lösung werden ebenfalls mehrere Heuristiken vorgeschlagen, die entweder erst einen gültigen Umlaufplan und den darauf gültigen Dienstplan berechnen oder umgekehrt. Freling testete seinen Ansatz auf realen und künstlich generierten Probleminstanzen mit bis zu 148 Fahrten und einem Depot. Die Laufzeit für die größte Instanz betrug knapp unter einer Stunde, wovon ca. 98% im Pricing-Problem der Column-Generation verbracht wurden.

In [Freling et al., 2001] wird der oben beschriebene Algorithmus in einer Fallstudie zur Lösung realer Probleminstanzen von den Rotterdamer Verkehrsbetrieben (RET) eingesetzt. Dabei betrachten die Autoren einen Sonderfall, bei dem für Dienste, die eine Rundreise bilden, eine extra Anforderung an die minimale Wendezeit existiert. Die Besonderheit besteht darin, dass die minimal geforderte Wendezeit von der jeweiligen Konstruktion des Dienstes abhängt. Um dies berücksichtigen zu können, wird die Pricing-Phase entsprechend modifiziert. Getestet wird mit realen Probleminstanzen zwischen 131 und 259 Fahrten, einem Depot und nur einem Ablösepunkt. Dabei betrug die Laufzeit für das große Problem etwas weniger als 5 Stunden.

In [Freling et al., 2003] werden die Modelle und Methoden aus früheren Arbeiten nochmal zusammengefasst und einige neue Aspekte präsentiert. Die Menge der getesteten Instanzen wurde gegenüber der in [Freling, 1997] um eine zusätzliche, etwas größere Instanz mit 238 Fahrten ergänzt. Die Laufzeit für diese Instanz betrug über 67 Stunden, wovon ca. 90% im Pricing-Teil des Column-Generation-Verfahrens verbracht wurde. Die Differenz zwischen der besten zulässigen Lösung und der gefundenen unteren Schranke betrug dabei 3,6%.

[Haase and Friberg, 1999] beschreiben einen exakten Algorithmus für das integrierte Umlauf- und Dienstplanungsproblem. Dabei werden die Lösungsansätze von [Desrochers and Soumis, 1989] für die Dienstplanung und von [Ribeiro and Soumis, 1994] für die Umlaufplanung kombiniert. Das integrierte Modell ist als Set-Partitioning-Problem mit Diensten und Umläufen als Spalten und zusätzlichen Nebenbedingungen für eine korrekte Kopplung formuliert. Sie entwi-

ckeln einen *Branch-and-Cut-and-Price*-Algorithmus, in dem *Column-Generation* und *Cut Generation* in eine Branch-and-Bound-Methode eingebunden sind. Im Master-Problem des Column-Generation-Verfahrens wird die LP-Relaxation gelöst, während in der Pricing-Phase neue Umläufe bzw. Dienste durch das Lösen eines einfachen bzw. ressourcenbeschränkten Kürzeste-Wege-Problems erzeugt werden. Es konnten nur sehr kleine Instanzen mit dieser Methode gelöst werden. Die Autoren berichten, dass die Lösungszeit für Probleminstanzen mit 10 Fahrten innerhalb 1 Minute und mit 20 Fahrten über 1 Stunde lag. Instanzen mit 30 Fahrten konnten gar nicht gelöst werden.

[Haase et al., 2001] stellen einen weiteren exakten Ansatz zum Lösen integrierte Umlauf- und Dienstplanungsprobleme vor. Sie formulieren das Dienstplanungsproblem als ein Mehrgüterfluss-Problem mit zusätzlichen Nebenbedingungen für Fahrzeuge, die das Finden eines optimalen Umlaufplans im Nachhinein garantieren. Die Umläufe sind als Flüsse im Modell implizit abgebildet und können durch eine einfache Flussdekomposition extrahiert werden. Außerdem sind die Kosten auf den Kanten des zugrundeliegenden Netzwerks so definiert, dass sie sowohl personenbezogene als auch fahrzeugbezogene Kosten widerspiegeln. So wird beim Lösen des Dienstplanungsproblems eine gesamtoptimale Lösung sichergestellt. Der Lösungsalgorithmus basiert auf einem Branch-and-Price-Ansatz, in dem Column-Generation in eine Branch-and-Bound-Methode eingebunden ist. Das *Branching* erfolgt auf den Variablen für Verbindungskanten (vgl. [Desrochers and Soumis, 1989]). Zusätzlich werden leicht zu identifizierende Schnittebenen (*cutting planes*) hinzugefügt. Im Master-Problem wird die LP-Relaxation mit einer *Primalen-Simplex-Methode* gelöst. Das Pricing-Problem wird als ressourcenbeschränktes Kürzeste-Wege-Problem formuliert und exakt gelöst, wobei die Autoren an dieser Stelle auf die Möglichkeit der Dynamischer Programmierung hinweisen. Für große Instanzen schlagen sie eine heuristische Variante des Lösungsverfahrens vor. Sie unterscheidet sich von der exakten dadurch, dass alternative, heuristische Branching-Strategien benutzt werden und die rechte Handseite der mathematischen Formulierung mit Hilfe der *Perturbation*-Techniken (siehe [du Merle et al., 1999]) modifiziert wird. Außerdem kann der Lösungsprozess bei der heuristischen Variante durch ein Zusammenfassen von Knoten in der Pricing-Phase und einen früheren Abbruch von Column-Generation beschleunigt werden. Der vorgestellte exakte Ansatz wurde auf einer Reihe künstlich generierter Instanzen mit bis zu 150 Fahrten und 300 Dienstelementen (es sind auch Ablösepunkte innerhalb der Fahrten definiert) getestet. Die Lösungszeit war auf 3 Stunden pro Instanz begrenzt. Dabei konnten 6 Instanzen von 10 zum Optimum gelöst werden. Die durchschnittliche Differenz zwischen einer ganzzahliger und einer fraktionalen Lösung (*Integrality Gap*) lag bei ca. 0,3%. Mit der heuristischen Variante konnten Probleminstanzen mit bis zu 350 Fahrten und 700 Dienstelementen innerhalb der gegebenen 3 Stunden mit einem *Integrality Gap* von

bis zu 5,7% (aber durchschnittlich 0,3%) gelöst werden.

Ein weiterer Ansatz für die integrierte Planung wird von [Borndörfer et al., 2002] entwickelt. Das graphentheoretische Modell kombiniert früher entwickelte Modelle zur Umlaufplanung (siehe [Löbel and Strubbe, 1996], [Grötschel et al., 1997], [Löbel, 1999]) und Dienstplanung (siehe [Borndörfer et al., 2001]). Die mathematische Formulierung besteht aus einer Mehrgüterfluss-Komponente für die Umlaufplanung, Set-Partitioning-Komponente für die Dienstplanung und einer Menge von Kopplungsbedingungen, die eine konsistente Überdeckung der Leerfahrten sicherstellen. In dem vorgestellten Ansatz wird Column-Generation-Verfahren mit Lagrange-Relaxation der Kopplungsbedingungen kombiniert. Zur Lösung der Umlauf- und Dienstplanungsprobleme verwenden sie modifizierte Versionen der früher entwickelten Verfahren zur sequenziellen Umlauf- und Dienstplanung. Das *Lagrange-Dual-Problem* wird mit einem Subgradienten-Verfahren approximativ gelöst. Zur Berechnung einer zulässigen Lösung werden zwei primale Heuristiken beschrieben. Die erste löst das umlaufbasierte Dienstplanungsproblem für den aktuellen Umlaufplan in jeder Iteration der Column-Generation-Methode. Die zweite Heuristik benutzt das so genannte *Leerfahrt-Plunging*. Das Verfahren fixiert in jeder Iteration der Hauptschleife anhand der reduzierten Kosten einige Leerfahrten als durchzuführende Fahrten. Die Fixierungen sind zunächst vorläufig. Sie werden abhängig von der Entwicklung des Zielfunktionswertes in den Folgeiterationen als im Nachhinein gerechtfertigt angesehen oder rückgängig gemacht. Der Lösungsalgorithmus wird mit realen Praxisinstanzen mit bis zu 1457 Fahrgastfahrten und einem Depot getestet. Die Laufzeit betrug für größte Instanz ca. 6,5 Stunden, wovon über 50% im Pricing verbracht wurden.

4.3.3 Vollständige Integration mit mehreren Depots

Der wichtige Unterschied bei der Betrachtung mehrerer Depots im Vergleich zum Eindepot-Fall ist, dass das zugrundeliegende Mehrdepot-Umlaufplanungsproblem ein $\mathcal{NP}$-schweres Problem ist (siehe [Bertossi et al., 1987]), während das Eindepot-Umlaufplanungsproblem in polynomieller Zeit gelöst werden kann.

Nach unserem Wissen waren [Gaffi and Nonato, 1999] die Ersten, die das integrierte Umlauf- und Dienstplanungsproblem mit mehreren Depots behandelten. Allerdings betrachten sie kein allgemein gültiges Problem sondern einen Sonderfall für den Regionalverkehr, bei dem die Fahrer sich nur in Depots ablösen dürfen. Somit entsprechen die Dienststücke gleich den Umlaufstücken und das Problem reduziert auf das Finden einer Zuordnung von Fahrgastfahrten zu Fahrtsequenzen (Umlauf- oder Dienststück), die in einem Depot starten und enden und sich zu einer kostenminimalen Menge von Diensten und Umläufen zusammenfügen lassen. Die

mathematische Formulierung besteht aus einem Quasi-Assignment-Problem, das für jede Fahrt eine Mindestüberdeckung sicherstellt, und einer Menge von Kopplungsbedingungen, die eine mit dem Dienstplan konsistente Überdeckung der Leerfahrten garantieren. Außerdem kann die Anzahl der Umläufe pro Depot und die Zahl der Dienste pro Dienstart beschränkt werden. Der heuristische Lösungsansatz basiert auf einer Kombination von Column-Generation und Lagrange Relaxation. Wenn die Kopplungsbedingungen mit Hilfe der Lagrange-Relaxation relaxiert werden, zerfällt das Gesamtproblem in ein Quasi-Assignment-Problem, das mit einem *Bundle-Algorithmus* gelöst wird, und ein triviales Auswahlproblem. Die Lösung des Quasi-Assignment-Problems liefert eine Menge potenzieller Umlauf- bzw. Dienststücke, aus denen in der Pricing-Phase neue gültige Umläufe bzw. Dienste erzeugt werden (als kürzeste bzw. ressourcenbeschränkte kürzeste Wege). Der gültige Dienstplan wird mit Hilfe einer speziellen Greedy-Prozedur und einer iterativen Variablenfixierung berechnet. Aus Dienststücken der resultierenden Dienste wird durch das Lösen eines Mehrgüterfluss-Problems ein kompatibler Umlaufplan konstruiert. Der vorgestellte Ansatz wurde auf zwei Problemsätzen aus Regional- bzw. Stadtteilverkehr mit bis zu 257 Fahrten und 28 Depots bzw. 135 Fahrten und 4 Depots getestet. Für das Scenario Regionalverkehr konnte das Verfahren immer eine zulässige Lösung finden, während das aktuell eingesetzte Planungssystem teilweise nicht verplanbare Dienste erzeugte. Die Stadtteilverkehr-Instanzen zeichnen sich wegen kürzerer Entfernungen durch eine größere Anzahl an kompatiblen Fahrtverknüpfungen aus. Hier war das Ergebnis bei manchen Instanzen sogar schlechter als vom aktuell eingesetzten Planungssystem, was auf eine heuristische Funktionsweise des vorgestellten Lösungsverfahrens zurückzuführen ist.

[Huisman, 2004, Huisman et al., 2005] betrachtet das allgemeine integrierte Umlauf- und Dienstplanungsproblem mit mehreren Depots, das sowohl für den Regional- als auch den Stadtverkehr gültig ist. Er erweitert die Modelle und Algorithmen von Freling auf den Mehrdepot-Fall (siehe z.B. [Freling, 1997], [Freling et al., 1999] und [Freling et al., 2003]). Im Folgenden beschreiben wir den von Huisman vorgeschlagenen Ansatz etwas ausführlicher, da wir ihn als Referenz für die im Rahmen dieser Arbeit erarbeiteten Verfahren benutzten (vgl. [Huisman, 2004], S. 80ff).

Sei $N = \{1, 2, \ldots, n\}$ die Menge aller Fahrgastfahrten, $E = \{(i,j) | i, j \; kompatibel, i, j \in N\}$ die Menge aller Fahrtverbindungen und D die Menge aller Depots. Für jedes Depot $d \in D$ wird ein azyklischer, gerichteter Graph $G^d = (V^d, A^d)$ mit Knoten $V^d = N^d \cup \{r^d, t^d\}$ und Kanten $A^d = E^d \cup (r^d \times N^d) \cup (N^d \times t^d)$ definiert, wobei r^d und t^d jeweils Depot d repräsentieren. Die Kosten c_{ij}^d für Kante $(i,j) \in A^d$ spiegeln die Fahrzeugkosten wider, die bei der Ausführung der durch diese Kante repräsentierten Aktivität entstehen. Dabei werden zu allen (r^d, i)- oder

(j, t^d)-Kanten (für alle $i, j \in N^d$) die fixen Fahrzeugkosten für die Benutzung eines Fahrzeugs hinzuaddiert. Es wird angenommen, dass ein Fahrzeug zwischen zwei Fahrten immer in ein Depot zurückkehrt anstatt an der Anschlusshaltestelle zu warten, wenn das aus Kostengründen günstiger ist. Solche Verbindungskanten, die zwei Fahrten mit einer Fahrt über Depot verbinden, werden *lange Kanten (long arcs)* genannt, während die übrigen Verbindungskanten *kurze Kanten (short arcs)* genannt. Die Menge der kurzen bzw. langen Kanten wird als $A^{sd} \subset A^d$ bzw. $A^{ld} \subset A^d$ bezeichnet. Weiterhin sei K^d die Menge aller für das Depot d gültigen Dienste und f_k^d die Kosten des Dienstes $k \in K^d$. Die Menge der Dienste, die eine durch den Knoten i repräsentierte Fahrgastfahrt bzw. eine durch die Kante (i, j) repräsentierte Verbindung beinhalten, wird mit $K^d(i)$ bzw. $K^d(i, j)$ bezeichnet. Entscheidungsvariable y_{ij}^d besagt, ob Kante (i, j) im Depot d verwendet wird, während x_k^d angibt, ob Dienst k aus Depot d in der Lösung ist. Das mathematische Modell wird wie folgt formuliert:

$$\min \quad \sum_{d\in D} \sum_{(i,j)\in A^d} c_{ij}^d y_{ij}^d + \sum_{d\in D} \sum_{k\in K^d} f_k^d x_k^d \tag{4.1}$$

$$\sum_{d\in D} \sum_{\{j:(i,j)\in A^d\}} y_{ij}^d = 1 \qquad \forall i \in N \tag{4.2}$$

$$\sum_{d\in D} \sum_{\{i:(i,j)\in A^d\}} y_{ij}^d = 1 \qquad \forall j \in N \tag{4.3}$$

$$\sum_{\{i:(i,j)\in A^d\}} y_{ij}^d - \sum_{\{i:(j,i)\in A^d\}} y_{ji}^d = 0 \qquad \forall d \in D, \forall j \in N^d \tag{4.4}$$

$$\sum_{k\in K^d(i)} x_k^d - \sum_{\{j:(i,j)\in A^d\}} y_{ij}^d = 0 \qquad \forall d \in D, \forall i \in N^d \tag{4.5}$$

$$\sum_{k\in K^d(i,j)} x_k^d - y_{ij}^d = 0 \qquad \forall d \in D, \forall (i,j) \in A^{sd} \tag{4.6}$$

$$\sum_{k\in K^d(i,t^d)} x_k^d - y_{it^d}^d - \sum_{\{j:(i,j)\in A^{ld}\}} y_{ij}^d = 0 \qquad \forall d \in D, \forall i \in N^d \tag{4.7}$$

$$\sum_{k\in K^d(r^d,j)} x_k^d - y_{t^d j}^d - \sum_{\{i:(i,j)\in A^{ld}\}} y_{ij}^d = 0 \qquad \forall d \in D, \forall j \in N^d \tag{4.8}$$

$$x_k^d, y_{ij}^d \in \{0,1\} \quad \forall d \in D,\ \forall k \in K^d, \forall (i,j) \in A^d \tag{4.9}$$

Das Ziel ist, die Summe der Fahrzeug- und Dienstkosten zu minimieren. Die ersten drei Nebenbedingungsmengen (4.2)-(4.4) garantieren einen zulässigen Umlaufplan. Bedingungen (4.5) stellen eine konsistente Überdeckung der Fahrgastfahrten sicher (eine Fahrgastfahrt i ist genau dann durch einen Dienst aus Depot d überdeckt, wenn sie selbst zu diesem Depot zugeordnet wird). Die Nebenbedingungen (4.6) garantieren eine konsistente Kopplung der Verbindungsfahrten im resultierenden

Dienst- und Umlaufplan (eine Verbindung zwischen zwei Fahrten i und j wird genau dann durch einen Dienst aus Depot d überdeckt, wenn die entsprechende Leerfahrt im Umlaufplan vorhanden ist). Schließlich stellen die Nebenbedingungen (4.7)-(4.8) analog zu (4.6) eine konsistente Kopplung für Depotfahrten im resultierenden Umlauf- und Dienstplan dar.

Der Lösungsalgorithmus basiert auf einer Kombination von Column-Generation und Lagrange-Relaxation. Im eingeschränkten Master-Problem werden die Nebenbedingungen (4.4)-(4.8) mit Hilfe der Lagrange Relaxation relaxiert. Das Gesamtproblem zerfällt in ein großes Eindepot-Umlaufplanungsproblem, das mit dem Vorwärts-Rückwärts-Auktionsalgorithmus von Freling (siehe [Freling, 1997]) gelöst wird, und ein triviales Auswahlproblem für x-Variablen. Das Lagrange-Dual-Problem wird mit einem Subgradienten-Verfahren approximativ gelöst. Für die Konstruktion neuer Dienste im Pricing-Problem wird die 2-Phasen-Prozedur von Freling (siehe [Freling, 1997]) benutzt. In der ersten Phase werden Dienststücke durch das Lösen Kürzester-Wege-Probleme zwischen allen kompatiblen Knotenpaaren auf einem speziellen Dienststückerzeugung-Graphen bestimmt. Dieser Graph ist eine Erweiterung von G^d. Die Kosten auf den Kanten entsprechen den reduzierten Kosten und sind so definiert, dass die Kosten eines Pfades gleich den reduzierten Kosten des durch diesen Pfad repräsentierten Dienststückes sind. In der zweiten Phase werden neue Dienste durch die Aufzählung aller möglichen Kombinationen aus den zuvor generierten Dienststücken erzeugt, wobei jede Kombination sowohl alle Dienstregeln erfüllen muss als auch negative reduzierte Kosten aufweisen muss. Das Pricing-Problem wird separat für jedes Depot gelöst. Um eine zulässige Lösung zu berechnen, wird die Originalformulierung mit Diensten, die während des Column-Generation-Verfahrens erzeugt wurden, wieder mit Hilfe der Lagrange-Relaxation gelöst. Allerdings werden hier nur die Nebenbedingungen (4.5)-(4.8) relaxiert, sodass die Lösung des resultierenden Unterproblems einen gültigen Umlaufplan liefert. Dieses Unterproblem ist das Mehrdepot-Umlaufplanungsproblem und ist $\mathcal{NP}$-schwer. Da aber das Subgradienten-Verfahren mit vorhandenen „guten" Multiplikatoren startet, braucht es nur wenige Iterationen, bis eine gute Lösung gefunden wird. Für den resultierenden Umlaufplan wird ein kompatibler Dienstplan durch das Lösen umlaufbasierter Dienstplanungsprobleme für jedes Depot bestimmt.

Huisman testet seinen Ansatz auf realen und künstlich generierten Probleminstanzen. Die Anzahl der Fahrten bei realen Instanzen variierte zwischen 194 und 653 Fahrten und 4 Depots, wobei die meisten Fahrten nur zu einem Depot zugeordnet werden können. Die Lösungszeit betrug mehrere Stunden (genauere Angaben fehlen) schon für kleine Instanzen. Im Vergleich zu der Lösung durch die sequenzielle Vorgehensweise konnten bei dem integrierten Ansatz bis zu 20% der Dienste eingespart werden. Der zweite Testsatz besteht aus künstlich generierten Instanzen

mit bis zu 200 Fahrten und 2 Depots bzw. bis zu 160 Fahrten und 4 Depots. Die maximale Lösungszeit war auf 3 Stunden begrenzt. Dabei betrug die Abweichung der jeweils besten gefundenen Lösung von der errechneten unteren Schranke ca. 5-8%. Der zeitintensivste Teil war das Pricing-Problem, in dem bis zu 85% der gesamten Laufzeit verbracht wurde.

Zusätzlich zu dem oben beschriebenen Verfahren untersucht Huisman einen alternativen Lösungsansatz, der auf der Eindepot-Formulierung von [Haase et al., 2001] basiert. Er erweitert diese Formulierung für den Mehrdepot-Fall und schlägt einen Lösungsalgorithmus vor, der ebenfalls auf einer Kombination von Column-Generation und Lagrange-Relaxation basiert. Die durchgeführten Tests auf realen und künstlich generierten Probleminstanzen zeigen, dass die beiden Ansätze bezüglich der Laufzeit und Lösungsqualität ähnliche Ergebnisse liefern.

Ein weiteres Verfahren zur Behandlung integrierter Umlauf- und Dienstplanungsprobleme mit mehreren Depots wird von [Borndörfer et al., 2004] präsentiert. Die mathematische Formulierung vereinigt ein Mehrgüterfluss-Problem der Umlaufplanung, eine Set-Partitioning-Komponente der Dienstplanung und eine Menge von Kopplungsbedingungen, die eine konsistente Benutzung der Leerfahrten im resultierenden Umlauf- und Dienstplan garantieren. Der Lösungsalgorithmus ist eine Kombination von Column-Generation-Verfahren und Lagrange-Relaxation und hat einige Ähnlichkeiten zur Methode von Freling (siehe [Freling, 1997]). Nach einer Relaxation der Kopplungsbedingungen zerfällt das Gesamtproblem in Unterprobleme, nämlich ein Mehrdepot-Umlaufplanungsproblem, das mit einem effizienten Verfahren von Löbel (siehe [Löbel, 1997, Löbel, 1999]) gelöst wird, und ein Dienstplanungsproblem, das ähnlich wie in [Borndörfer et al., 2001, Borndörfer et al., 2003] behandelt wird. Das Lagrange-Dual-Problem wird mit einer *Proximal-Bundle-Methode* (siehe [Kiwiel, 1995]) näherungsweise gelöst. Der Unterschied dieser Methode im Vergleich zum Subgradienten-Verfahren besteht darin, dass sie viel komplexer und zeitintensiver ist, dafür aber viel bessere Schranken produziert und außerdem primale Informationen über die Variablen liefert. Diese Eigenschaften werden in einem Branch-and-Bound-Algorithmus ausgenutzt. Die Pricing-Phase von Column-Generation wird nur dann ausgeführt, wenn das Stabilitätszentrum der Bundle-Methode sich verändert und die Zielfunktion sich um einen bestimmten Wert verbessert hat. Die beschriebene Lösungsprozedur wird in einen Backtracking-Algorithmus eingebunden. Hier werden die Leerfahrt-Variablen anhand ihrer primalen Information aus der Bundle-Methode iterativ fixiert (und evtl. wieder freigegeben), bis ein kompletter Umlaufplan entsteht. Ein kompatibler Dienstplan kann dann mit einem Algorithmus von [Borndörfer et al., 2003] bestimmt werden. Die Autoren berichten über erfolgreiche Tests auf realen und künstlich generierten Probleminstanzen. Die zwei größten realen Instanzen beinhalten 1414 Fahrten, 3 Fahr-

zeugtypen und 1 Depot bzw. 634 Fahrten, 5 Fahrzeugtypen und 3 Depots. Die Laufzeit betrug entsprechend ca. 125 bzw. 17 Stunden, dabei konnten viel bessere Ergebnisse im Vergleich zu den aktuell gefahrenen Umlauf- und Dienstplänen erzielt werden. Der Testsatz der künstlich generierten Instanzen (verfügbar unter [Huisman, 2005]) bestand aus Problemen mit bis zu 400 Fahrten mit 2 bzw. 4 Depots. Die Lösungszeit betrug hier entsprechend ca. 3,3 bzw. 12 Stunden. Dabei konnten die in [Huisman, 2004] veröffentlichen Ergebnisse übertroffen werden.

[Mesquita et al., 2005] präsentierten einen exakten Branch-and-Price-Ansatz für integrierte Umlauf- und Dienstplanung mit mehreren Depots. Die mathematische Formulierung ist ähnlich zu [Huisman, 2004], allerdings in einer etwas kompakteren Form. Der Lösungsansatz basiert auf einem Column-Generation-Ansatz, der in eine Branch-and-Bound-Prozedur eingebunden ist. Das Master-Problem ist eine LP-Relaxation der Originalformulierung, die mit CPLEX direkt gelöst wird, während neue Spalten in der Pricing Phase mit Hilfe eines Algorithmus zur Bestimmung ressourcenbeschränkter kürzester Wege erzeugt werden. Die Autoren schlagen zwei Branching-Strategien (beide auf Leerfahrt- und Depotkanten) vor. Der Ansatz wird auf künstlich generierten Instanzen (verfügbar unter [Huisman, 2005]) mit nur 80 Fahrten und 4 Depots getestet. Eine Aussage über die Laufzeit fehlt jedoch.

4.3.4 Behandlung großer Probleminstanzen

Im Kontext der Mehrdepot-Umlaufplanung, im Unterabschnitt 4.1.1, wurden bereits Techniken erwähnt, die große Probleme durch ihre Aufteilung in mehrere kleine behandeln. Eine ähnliche Idee verfolgen [de Groot and Huisman, 2006] für integrierte Umlauf- und Dienstplanungsprobleme. Sie schlagen vor, ein großes Problem in mehrere kleinere Unterprobleme aufzuteilen und sie dann mit einem integrierten (oder sequenziellen) Ansatz zu lösen. Eine Möglichkeit besteht darin, die Fahrten erst zu Depots zuzuordnen und danach mehrere Eindepot-Probleme integriert zu lösen. Die Zuordnung zu Depots kann z.B. nach einem geographischen Prinzip erfolgen (jede Fahrt wird dem Depot zugeordnet, das am nächsten zu ihrer Start- oder/und Endhaltestelle liegt) oder aus der Lösung des Mehrdepot-Umlaufplanungsproblems abgeleitet sein. Eine weitere Möglichkeit ist, die Menge der Fahrgastfahrten in kleinere Mengen aufzuteilen und mehrere kleine Umlauf- und Dienstplanungsprobleme mit mehren Depots integriert zu lösen. Eine mögliche Aufteilungsstrategie dafür ist, erst ein MDVSP zu lösen und dann die Fahrten bestimmter Umläufe zu einem Unterproblem zusammenzufassen. Die resultierende Lösung kann noch verbessert werden, indem unterschiedliche Unterprobleme anschließend wieder vereinigt und noch mal gelöst werden (z.B. vereinige alle Umläufe aus unterschiedlichen Unterproblemen nach ihrer Depotzugehörigkeit und löse ein großes Dienstplanungsproblem für jedes Depot). Die vorgeschlagenen Auftei-

lungsstrategien wurden auf realen Probleminstanzen mit bis zu 1372 Fahrten und 6 Depots getestet. Die Ergebnisse hinsichtlich der Anzahl der Dienste waren deutlich besser als die Lösung des sequenziellen Ansatzes und für viele Instanzen besser als die Lösung, die ein integrierter Ansatz auf dem Gesamtproblem (ohne Aufteilung) nach einer vorgegebenen Zeit berechnen konnte. Außerdem war die gesamte Laufzeit, inkl. Aufteilen und Lösen der Unterprobleme, für alle Testinstanzen signifikant kleiner als die Zeit, die für die Lösung des Gesamtproblems mit dem integrierten Ansatz notwendig war (wenn sie nicht beschränkt war).

4.3.5 Integration im Bereich der Flugplanung

Im Bereich der Flugplanung wurden in der letzten Zeit ebenfalls einige Ideen zur Integration der Planungsschritte *Aircraft Rotation* und *Crew Pairing* vorgestellt. Diese Teilprobleme können zwar nicht direkt auf die Umlauf- bzw. Dienstplanung im ÖPNV übertragen werden, besitzen allerdings gewisse gemeinsame Aspekte und ähnliche Lösungstechniken. Eine Interaktion zwischen Aircraft Rotation und Crew Pairing besteht darin, dass die so genannte Sit-Time (die Mindestzeit, die eine Crew zwischen zwei aufeinanderfolgenden Flügen am Boden verbringen muss) davon abhängt, ob die Crew mit demselben Flugzeug weiterfliegt oder nicht. Ein weiterer Unterschied zu der Planung im ÖPNV ist, dass bei der Flugplanung Leerfahrten stark beschränkt sind. Außerdem wird meistens mit einem Planungshorizont von einer Woche statt einem Tag geplant.

[Klabjan et al., 2002] betrachtet Aircraft Rotation und Crew Pairing gemeinsam und benutzt eine ähnliche Idee wie [Haase et al., 2001]. Anstatt erst Aircraft Rotation und dann Crew Pairing sequenziell zu lösen, erweitert er die Set-Partitioning-Formulierung des Crew-Pairing-Problems um zusätzliche Nebenbedingungen, die eine gültige und kompatible Lösung des Aircraft-Rotation-Problems im Nachhinein sicherstellen. Eine ähnliche Formulierung benutzen [Cordeau et al., 2001]. Ihr Lösungsansatz basiert auf Bender-Dekomposition und Column-Generation. Die zulässige Lösung wird mit einem heuristischen Branch-and-Bound-Verfahren bestimmt. Sie zeigen enormes Einsparungspotenzial durch eine integrierte Betrachtung im Vergleich zur sequenziellen Planung. Für eine reale Instanz einer Kanadischen Fluggesellschaft mit 500 Flügen betrug die jährliche Einsparung mehrere Millionen Dollar. [Mercier et al., 2005] verfolgen diesen Lösungsansatz weiter. Sie modifizieren den Dekompositionsalgorithmus und erweitern ihn durch fortgeschrittene Techniken. Dadurch konnte die Laufzeit im Vergleich zu [Cordeau et al., 2001] um Faktor 10 verbessert werden. Außerdem berichten die Autoren, dass ihr Ansatz im Vergleich zu [Cohn and Barnhart, 2003] eine robustere und bessere Lösung in kürzeren Zeit findet.

Ein weiterer Schritt in der Integration einzelner Planungsaufgaben machen [Sandhu and Klabjan, 2005]. Sie betrachten drei Phasen, nämlich *Fleet Assignment, Aircraft Rotation und Crew Pairing* gemeinsam. Im vorgestellten Modell sind Fleet Assignment und Crew Pairing direkt abgebildet, während die Aircraft Rotation, ähnlich wie in [Klabjan et al., 2002], durch zusätzliche Nebenbedingungen implizit berücksichtigt wird. Die Autoren entwickeln zwei Lösungsalgorithmen. Einer basiert auf einer Kombination von Column-Generation und Lagrange-Relaxation und der andere auf einer Bender Dekomposition. Das Pricing Problem wird in beiden Ansätzen mit einer parallelen Version des ressourcenbeschränkten Kürzeste-Wege-Algorithmus (siehe [Klabjan, 2003]) auf einem Cluster mit mehreren Computern gelöst. Als Testsatz wurden reale Instanzen mit 205 bis 942 Flügen und 2-4 Flottengruppen verwendet. Die durchgeführten Tests zeigen, dass die integrierte Betrachtung im Vergleich zur traditionellen Planungsweise für diese Instanzen eine potenzielle Einsparung von über 50 Millionen Dollar jährlich bringen kann.

4.4 Handlungsbedarf

Zu Zeit ist die Nachfrage nach Ansätzen für einen effizienten Ressourceneinsatz seitens der Verkehrsunternehmen so groß wie nie. Dies ist einerseits auf den immer weiter steigenden Wettbewerbsdruck und auf der anderen Seite auf die immer tiefer greifende Kürzung von Subventionen im ÖPNV zurückzuführen. Computergestützte Planungstools auf Basis von OR-Methoden gehören heutzutage in vielen ÖPNV-Unternehmen zu Standardwerkzeugen. Mit der Weiterentwicklung der Computertechnik und mathematischen Optimierung werden solche Werkzeuge immer leistungsfähiger und bieten immer mehr Unterstützung im Planungsprozess der Verkehrsbetriebe. Durch diese Unterstützung werden die ÖPNV-Unternehmen nicht nur durch eine schnelle und effiziente Planung, sondern auch durch mehr Flexibilität und die Möglichkeit, schneller auf geänderte Umweltfaktoren zu reagieren, im offenen Wettbewerb gestärkt.

Die sequenzielle Umlauf- und Dienstplanung wird seit langer Zeit erforscht und ist sehr weit fortgeschritten. Es existiert eine Reihe leistungsstarker Verfahren, die in kommerziellen Planungswerkzeugen integriert sind. Auf der anderen Seite ist es bekannt, dass eine simultane Betrachtung von Umlauf- und Dienstplanung einen weiteren Schritt in Richtung effizienter Ressourceneinsatzplanung geht und zu zusätzlichen Kostenersparnissen führen kann. Außerdem spielt eine simultane Verplanung von Diensten und Umläufen bei den Betrieben im Nachbarsort- und Regionalverkehr eine wichtige Rolle, da dort die herkömmliche sequenzielle Vorgehensweise wegen der besonderen Netzstruktur nur sehr begrenzt einsetzbar ist[3].

[3]das liegt an der speziellen Netzstruktur im Nachbarsort- und Regionalverkehr. Dort gibt es

Wie die vorgestellte Übersicht zeigt, widmen sich immer mehr Wissenschaftler in den letzten Jahren einer simultanen bzw. gekoppelten Behandlung der Umlauf- und Dienstplanung. Dabei wird das komplett integrierte Umlauf- und Dienstplanungsproblem erst seit kurzer Zeit etwas intensiver untersucht und ist nach unserem Wissensstand in keinem kommerziellen Softwarepaket eingebunden. Einer der Gründe dafür ist sicherlich die Tatsache, dass damit nur relativ kleine bzw. mittlere praxisrelevante Probleme gelöst werden konnten.

Im Rahmen der vorliegenden Arbeit soll ein Beitrag zur weiteren Erforschung der simultanen Umlauf- und Dienstplanung geleistet werden. Da die Planungssituationen unterschiedliche Merkmale aufweisen, ist ein einziges Verfahren voraussichtlich nicht für alle Anforderungen geeignet. Das Ziel der Arbeit ist daher die Entwicklung und Erprobung mehrerer Lösungsverfahren zum Lösen integrierter Umlauf- und Dienstplanungsprobleme mit mehreren Depots und einem unterschiedlichen Grad der Integration. Weiterhin sollen die Einsatzgebiete aller Verfahren analysiert und gegeneinander abgegrenzt werden.

Zunächst soll ein vollständig integrierter Ansatz zum Lösen der Umlauf- und Dienstplanungsprobleme mit mehreren Depots entwickelt werden. Die aus der Literatur bekannten Ansätze sollen dabei um neue Modellierungstechniken erweitert werden. Dank dieser Techniken soll eine starke Reduktion der Problemgröße erreicht werden, die eine effiziente Bewältigung von größeren Problemen ermöglicht. Insbesondere soll die Eignung von Time-Space-Netzwerken bei der Modellierung erprobt werden. In diesem Zusammenhang sollen unterschiedliche Alternativen bei der Formulierungen und Lösung einzelnen Phasen des gesamten Lösungsverfahrens untersucht werden.

Da eine integrierte Betrachtung von Umlauf- und Dienstplanung aus Gründen der Komplexität schon für Probleme mittlerer Größe an ihre Grenzen stößt, bleibt eine sequenzielle Abarbeitung der beiden Planungsphasen für viele Verkehrsbetriebe immer noch die einzige praktikable Alternative. Als zweiter Schwerpunkt dieser Arbeit soll daher ein teilintegriertes Verfahren entwickelt werden, das einen Kompromiss zwischen der sequenziellen und simultanen Behandlung der Umlauf- und Dienstplanung darstellt. Durch eine gewisse Kopplung der beiden Planungsprobleme sollen große praxisrelevante Instanzen mit einer signifikant besseren Lösungsqualität als bei der herkömmlichen sequenziellen Planung gelöst werden.

Schließlich soll ein heuristischer Ansatz zum Lösen großer integrierter Umlauf- und Dienstplanungsprobleme entwickelt werden. Dabei sollen die beiden Planungs-

---

im Vergleich zum städtischen Nahverkehr nur wenige oder gar keine Ablösepunkte außerhalb der Depots. Außerdem sind die Ablöseorte oft weit voneinander entfernt, sodass kein Transfer (zu Fuß oder mit anderen Transportmitteln) zwischen ihnen möglich ist. Somit führt eine Dienstplanung auf Basis der zuvor bestimmten Umläufen oft zu einer unzulässigen Lösung und umgekehrt.

phasen zunächst heuristisch verkleinert und dann vollständig integriert gelöst werden. Insbesondere soll dabei eine passende Technik zur Verkleinerung des Lösungsraums gefunden werden.

Eine weitere Anforderung an die zu entwickelten Verfahren ist, dass sie möglichst miteinander kombinierbar sein sollen, um dem Planer ein flexibles Instrument zur Lösung der Umlauf- und Dienstplanungsprobleme mit unterschiedlichen Anforderungen an die Lösungszeit und Lösungsqualität anzubieten.

Kapitel 5

Integrierte Umlauf- und Dienstplanung

In diesem Kapitel wird einer der zentralen, im Rahmen der vorliegenden Arbeit entwickelten Ansätze zur Lösung der Umlauf- und Dienstplanungsprobleme vorgestellt. Der Ansatz basiert auf einem Column-Generation-Verfahren in Verbindung mit der Technik der Lagrange-Relaxation. Der Aufbau des Kapitels spiegelt einzelne Schritte des vorgeschlagenen Column-Generation-Ansatzes wider. Zu jedem Schritt werden mehrere Lösungsalternativen bzw. unterschiedliche Formulierungen untersucht.

Wir präsentieren ein neues Modell für das integrierte Umlauf- und Dienstplanungsproblem mit mehreren Depots. Das zugrunde liegende Netzwerkmodell wird mit einer neuartigen Modellierungstechnik als Time-Space-Netzwerk formuliert, die dank ihrer Struktur zu einer erheblichen Reduktion der Netzwerkgröße und der daraus abgeleiteten mathematischen Formulierung führt.

5.1 Problem-Formulierung

Das *integrierte Mehrdepot-Umlauf- und Dienstplanungsproblem (MD-VCSP)* kann wie folgt definiert werden: Gegeben sei eine Menge zu verplanender Fahrgastfahrten innerhalb eines fest definierten Planungszeitraums. Gesucht wird ein zulässiger Umlauf- und ein dazu kompatibler, zulässiger Dienstplan mit minimalen Gesamtkosten. Die beiden Pläne sind dann kompatibel, wenn sie jede Fahrgastfahrt genau einmal und die Leerfahrten konsistent überdecken. Für die Zulässigkeit der Umlauf- bzw. Dienstpläne gelten die üblichen Anforderungen, nämlich

- ein Umlaufplan ist zulässig, falls jede Fahrgastfahrt von genau einem Fahr-

zeug bedient wird und alle Fahrzeuge am Ende des Tages zum selben Depot zurückkehren, von dem sie gestartet sind;

- ein Dienstplan ist zulässig, falls jedes Dienstelement aus dem Umlaufplan in genau einem Dienst enthalten ist und alle Dienste bezüglich einer Reihe von gesetzlichen, tariflichen sowie technischen Vorschriften und Dienstregeln gültig sind.

In der von uns behandelten Problemstellung treffen wir folgende Annahmen (vgl. [Huisman, 2004]):

- Die Kosten für das MD-VCSP setzen sich aus fahrzeugbezogenen und dienstbezogenen Kosten zusammen. Das primäre Ziel für die Umlaufplanung ist eine Minimierung der Fahrzeuganzahl und für die Dienstplanung eine Minimierung der Dienstanzahl. Zusätzlich können variable Fahrzeugkosten (wie z.B. kilometerbezogene Kosten für die zurückgelegte Strecke oder zeitbezogene Kosten für jede Zeiteinheit, das ein Fahrzeug außerhalb des Depots verbringt) bzw. variable Dienstkosten (z.B. arbeitszeitbezogene Kosten oder Zuschüsse für Überstunden) berücksichtigt werden.

- Ein Fahrzeug wählt zwischen zwei Fahrgastfahrten immer eine *Fahrt über Depot*, wenn sie möglich und aus Kostenpunkten günstiger ist. Eine Fahrt über Depot heißt, dass das Fahrzeug, anstatt an der Haltestelle zu warten, zwischenzeitlich ins Depot einrückt, dort wartet und anschließend zu der Starthaltestelle der zunächst auszuführenden Fahrgastfahrt wieder ausrückt. Beim Warten im Depot fallen typischerweise keine variablen Fahrzeug- und Personalkosten an.

- Jeder Dienst wird einem Depot zugeordnet und kann nur Dienstelemente enthalten, die aus den Umläufen desselben Depots abgeleitet werden. Das heißt aber nicht, dass jeder Dienst im Depot starten und enden muss.

- Die Zulässigkeit eines Dienststückes hängt nur von seiner Länge ab, d.h. sie muss zwischen der zulässigen Mindest- und Höchstdauer liegen.

- Fahrerwechsel sind an jedem zuvor definierten, zulässigen Ablösepunkt erlaubt, d.h. ein Fahrzeug kann im Laufe des Tages von mehreren Fahrern besetzt werden bzw. ein Fahrer kann im Laufe des Tages mehrere Fahrzeuge bedienen.

- Jedes Fahrzeug muss mit einem Fahrer besetzt sein (engl.: *continuous attendance*), wenn es sich außerhalb des eigenen Depots befindet (auch wenn

es steht). Befindet sich ein Fahrzeug im eigenen Depot, dann ist die Anwesenheit eines Fahrers nicht notwendig. Das Warten im Depot ist somit kein Dienstelement.

Aus der letzten Annahme geht hervor, dass ein Depotaufenthalt für Fahrer zwangsweise als Arbeitszeitunterbrechung zählt, d.h. sobald ein Fahrzeug das eigene Depot passiert, gilt das entsprechende Dienststück des Fahrers als beendet. Eine weitere Folgerung dieser Annahme besagt, dass ein Fahrer immer von einem anderen Fahrer abgelöst wird, bevor er seine Pause außerhalb des Depots startet, da das Fahrzeug sonst unbesetzt bliebe.

Weiterhin treffen wir zur besseren Vergleichbarkeit mit existierenden Modellen zunächst die gleiche Annahme wie [Huisman, 2004], nämlich dass als Ablösepunkte Start und Ende jeder Fahrgastfahrt gelten. Somit entspricht jede Fahrgastfahrt gleichzeitig einem Dienstelement. Der allgemeine Fall mit beliebig platzierten Ablösepunkten, also auch innerhalb einer Fahrgastfahrt, wird im Abschnitt 5.7 diskutiert.

5.1.1 Netzwerkmodell

Im Unterabschnitt 4.1.1 wurden unterschiedliche Modellierungen des Umlaufplanungsproblems beschrieben. Dabei liegt den meisten Netzwerkmodellen eine etwa gleiche Netzwerkstruktur zugrunde, in der jede mögliche Fahrtenverknüpfung *(connection)* explizit durch eine Kante abgebildet wird. Solche Netzwerke nennen wir *Connection-basierte Netzwerke (CBN)*. Ein Nachteil dieser Modellierung ist, dass die Anzahl solcher Verbindungskanten quadratisch mit der Anzahl der Fahrten ansteigt, sodass die Netzwerkgröße und damit auch die Größe des mathematischen Modells bei großen praxisrelevanten Instanzen sehr schnell eine kritische Grenze erreicht. In der Literatur wurden zwar einige Ansätze vorgestellt, die die Kantenmenge durch unterschiedliche Heuristiken reduzieren, allerdings engen sie dadurch den Lösungsraum des Problems ein, so dass die Optimalität der gefundenen Umlaufpläne nicht mehr garantiert werden kann.

Eine komplett neue Vorgehensweise das Umlaufplanungsproblem zu modellieren, schlagen [Mellouli and Kliewer, 2002] und [Kliewer et al., 2006,Kliewer, 2005] vor. Sie benutzen eine andere Struktur des zugrunde liegenden Netzwerkes, das so genannte *Time-Space-Netzwerk (TSN)*, in dem die kompatiblen Fahrgastfahrten anstatt durch explizite Verbindungskanten implizit durch Netzwerkpfade verbunden werden. Der entscheidende Vorteil von TSN gegenüber den CBN-Modellen besteht in einer signifikanten Reduktion der Kantenanzahl im Netzwerk, ohne dabei den Lösungsraum zu beschränken. Dies führt zu einer Reduktion der Größe des

mathematischen Modells, so dass es mit Hilfe mathematischer Optimierungssoftware gelöst werden kann.

Das VSP ist ein Bestandteil des integrierten Problems. Sein Netzwerk wird als Basis in Netzwerkmodellen für das VCSP benutzt. Netzwerkmodelle aller im Abschnitt 4.3 vorgestellten Lösungsansätze für VCSP basieren auf „klassischen" Connection-basierten Netzwerken. Somit trifft der oben geschilderte Nachteil dieser Modellierung auch im integrierten Fall zu.

Im Rahmen dieser Arbeit wird zum ersten Mal die Modellierungstechnik aus der Umlaufplanung, das Time-Space-Netzwerk mit einer speziellen Struktur, für das integrierte Umlauf- und Dienstplanungsproblem eingesetzt. Somit besitzt das zugrunde liegende Netzwerk viel weniger Kanten. Sowohl das gesamte mathematische Modell als auch entsprechende Unterprobleme, die durch den gewählten Algorithmus entstehen, werden dadurch kleiner und können einfacher gelöst werden. Im Folgenden wird das TSN-basierte *Planungsnetzwerk* für VCSP beschrieben (vgl. TSN-basiertes Netzwerk für MDVSP in [Mellouli and Kliewer, 2002] und [Kliewer et al., 2006, Kliewer, 2005]).

Das Planungsnetzwerk

In einem TSN repräsentiert jeder Knoten einen bestimmten Zeitpunkt an einem bestimmten Ort, während jede Kante eine Zeit- und ggf. Ortstransformation zwischen den zugehörigen Knoten (also Zeitpunkten und Orten) darstellt. Zunächst wird der Fall mit nur einem Depot und einem Fahrzeugtyp betrachtet.

Jede Fahrgastfahrt wird im Netzwerk durch einen *Fahrtstart-* und einen *Fahrtend-Knoten* (mit zugehörigen Start- und Endzeiten sowie Start- und Endhaltestellen) und eine Kante dazwischen abgebildet. Zu jeder Fahrgastfahrt werden außerdem zwei zusätzliche *Depotknoten* und zwei *Depotkanten* erzeugt. Dabei repräsentiert die erste Kante eine Ausrückfahrt aus dem Depot zu der Fahrgastfahrt und verbindet den entsprechenden Depotknoten (Startpunkt der Ausrückfahrt) mit dem Fahrtstartknoten. Analog repräsentiert die zweite Kante eine Einrückfahrt und verbindet den Fahrtendknoten mit dem entsprechenden Depotknoten (Endpunkt der Einrückfahrt).

Alle Knoten werden nach zugehörigen Haltestellen gruppiert und innerhalb der Gruppe nach zugehöriger Zeit geordnet. Um die kompatiblen Fahrtaktivitäten, die an einer Haltestelle ankommen und von ihr abfahren, miteinander zu verknüpfen, werden alle Knoten innerhalb einer Haltestelle durch *Wartekanten* fortlaufend verbunden (siehe Abbildung 5.1). Analog werden auch Knoten im Depot verknüpft. Der Fluss über die Wartekanten repräsentiert die an der entsprechenden Haltestelle bzw. im Depot wartenden Fahrzeuge, wobei eine Flusseinheit genau einem Fahrzeug entspricht. Zu bemerken ist, dass dabei für jede ankommende Fahrt höchstens

eine Wartekante im TSN existiert. In einem CBN würde dagegen jede ankommende Fahrt mit allen Fahrten, die von dieser Anschlusshaltestelle später abfahren, explizit durch mehrere Verbindungskanten verbunden.

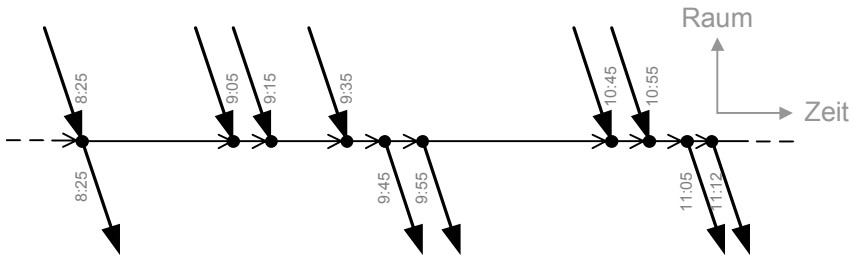

Abbildung 5.1: Netzwerkausschnitt für eine Haltestelle

Die Besonderheit der Busumlaufplanung im ÖPNV besteht im Vergleich zu der ähnlichen Planungsaufgabe im z.B. Flug- oder Schienenverkehr darin, dass ein Fahrzeug sich auch ohne Fahrgäste zwischen Haltestellen (fast uneingeschränkt) bewegen kann, um z.B. den Einsatzort der als nächstes auszuführenden Fahrgastfahrt zu erreichen. Somit sollen auch solche kompatiblen Fahrgastfahrten miteinander verknüpft werden, die keine gemeinsame Anschlusshaltestelle besitzen und trotzdem über eine Leerfahrt nacheinander bedienbar sind. In einem CBN-Modell wird dafür jedes solche Fahrtenpaar explizit durch eine Leerfahrt-Kante verbunden. Das TSN kommt dagegen dank seiner Struktur mit deutlich weniger Leerfahrt-Kanten aus.

Abbildung 5.2 zeigt einen Ausschnitt des TSN und des CBN für ein Problem mit zwölf Fahrten und zwei Haltestellen H_1 und H_2. Wie im Bild zu sehen ist, reicht es im TSN für jede Fahrgastfahrt jeweils nur eine (höchstens eine) Leerfahrt-Kante zu jeder Haltestelle zu erzeugen, um alle kompatiblen Fahrten, die später abfahren, über den Fluss durch die entsprechenden Wartekanten zu erreichen. Außerdem ist es nicht notwendig, für jede Fahrgastfahrt eine Leerfahrt-Kante zu allen Haltestellen zu erzeugen, weil über die Wartekanten an der Ankunftshaltestelle eine spätere, schon existierende Leerfahrt-Kante erreicht und benutzt werden kann, wenn dabei keine ansonsten mögliche Verknüpfung verloren gehen würde. So existiert beispielsweise für die Fahrten t_2 und t_3 im TSN keine explizite Leerfahrt-Kante zur Haltestelle H_2, trotzdem erreichen sie dort alle ihre kompatiblen Fahrten.

Die Regel zur Erzeugung von Leerfahrt-Kanten zwischen zwei Haltestellen H_1 und H_2 lässt sich wie folgt formalisieren: Für jede an der Haltestelle H_1 ankommende Fahrt t_i wird zunächst eine von der Haltestelle H_2 abfahrende Fahrt t_j bestimmt, die zeitlich als erste nach einer Leerfahrt von t_i zu H_2 erreichbar ist (d.h. t_j hat die kleinste Abfahrtszeit von allen zu t_i kompatiblen und von der Haltestelle H_2 abfahrenden Fahrten). Haben mehrere an H_1 ankommende Fahrten die gleiche erste

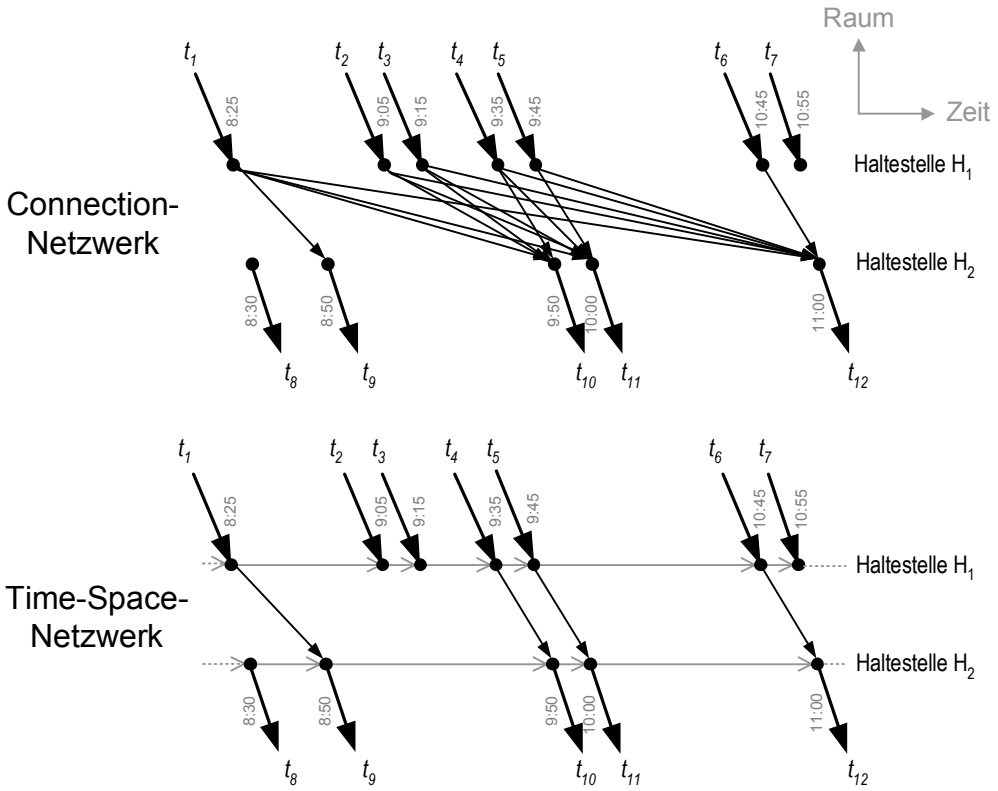

Abbildung 5.2: Leerfahrt-Kanten: CBN vs. TSN

kompatible Fahrt t_j an H_2, dann wird nur zwischen der zeitlich späteren von ihnen und t_j eine Leerfahrt-Kante erzeugt. Somit benötigt das TSN im oberen Beispiel nur 4 Leerfahrt-Kanten (anstatt 16), um dennoch alle kompatiblen Verknüpfungen implizit abzubilden.

Wegen der getroffenen Annahme, dass ein Fahrzeug immer eine *Fahrt über Depot* zwischen zwei Fahrgastfahrten wählt, wenn sie möglich und aus Kostengesichtspunkten günstiger ist, können alle solche Leerfahrt- und Wartekanten aus dem Netzwerk gelöst werden, die länger als eine Fahrt über das Depot sind. Zusätzlich werden auch solche Wartekanten gelöscht, die zwar kürzer als eine Fahrt über Depot sind, aber trotzdem nie benutzt werden. Abbildung 5.3 veranschaulicht zwei Fälle, in denen die Wartekante w_1 zwar kürzer als eine Fahrt über Depot ist, dennoch ist sie überflüssig, da t_1 und t_3 über einen günstigeren Weg über das Depot (über eine *Einrückfahrt-Kante*, eine oder mehrere *Depot-Wartekanten* und anschließende *Ausrückfahrt-Kante*) verbunden werden können.

Sei n die Anzahl der Fahrgastfahrten und m die Anzahl der Haltestellen, wobei unter einer Haltestelle hier nur die Endhaltestellen einer Linie verstanden werden. Die Anzahl solcher Endhaltestellen ist typischerweise viel kleiner als die Anzahl

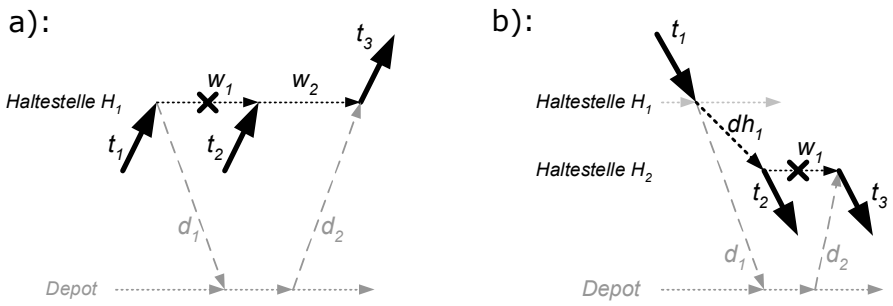

Abbildung 5.3: Löschen überflüssiger Wartekanten

der Fahrgastfahrten. In der vorgeschlagenen Modellierung als TSN wird für jede Fahrgastfahrt maximal eine Verbindungskante zu jeder Haltestelle erzeugt. Somit steigt die Anzahl aller Verbindungskanten in der Größenordnung von $\mathcal{O}(nm)$ an, mit $n >> m$, während diese Anzahl im klassischen Connection-basierten Modell $\mathcal{O}(n^2)$ beträgt. In der Tabelle 5.1 wird die Anzahl der Verbindungskanten beider Modellierungen gegenübergestellt. Es ist deutlich zu erkennen, dass die Kantenanzahl beim TSN etwa linear mit der Anzahl der Fahrgastfahrten steigt und nur einen Bruchteil im Vergleich zur Kantenanzahl in einem äquivalenten CBN beträgt. Für die Instanz mit 2000 Fahrten ist die Größe des Netzwerks bei der alternativen Modellierung ca. 110 Mal (!) kleiner. Ein wichtiger Punkt dabei ist, dass der Lösungsraum dadurch nicht beschränkt wird, d.h. alle möglichen kompatiblen Verbindungen sind implizit vorhanden.

Anzahl Fahrgastfahrten	100	200	400	800	2000
CBN	4.043	16.396	65.788	269.462	1.879.262
TSN	362	946	2.106	4.589	17.086

Tabelle 5.1: Anzahl der Verbindungskanten: CBN vs. TSN

Schließlich wird der letzte Depotknoten mit dem ersten Depotknoten durch die so genannte *Zirkulationsfluss-Kante* verbunden. Das Flussvolumen auf dieser Kante entspricht der Gesamtanzahl der eingesetzten Fahrzeuge. Das resultierende Netzwerk ist ein gerichteter azyklischer Graph. Ein Pfad von dem ersten bis zum letzten Depotknoten repräsentiert einen Umlauf für ein Fahrzeug. Die Abbildung 5.4 zeigt ein Beispiel-Netzwerk für ein Problem mit 6 Fahrten.

Für jede Kante im Netzwerk wird ein Kostensatz definiert. Wir betrachten einen allgemeinen Fall, bei dem die operativen Fahrzeugkosten sich aus einem einsatzzeit- und einem kilometerbezogenen Teil zusammensetzen. Somit ergeben sich für alle Wartekanten außerhalb des Depots nur Einsatzzeitkosten, während Wartekanten im Depot kostenfrei bleiben. Für alle Kanten, die Fahrtaktivitäten darstellen, nämlich

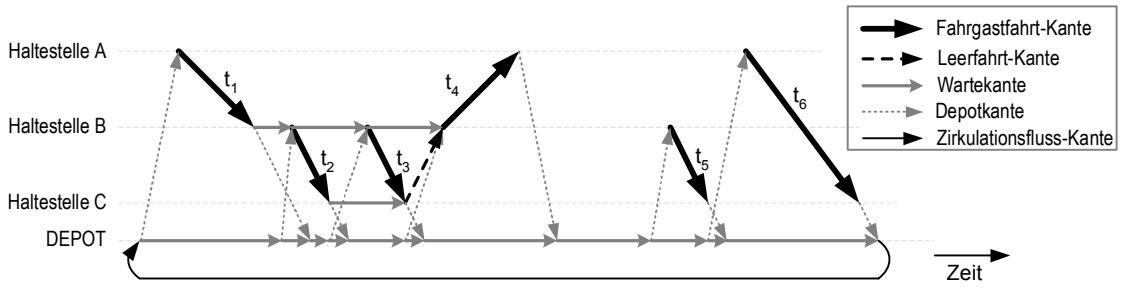

Abbildung 5.4: Beispiel eines Time-Space-Netzwerks

Fahrgastfahrt-, Depot- und Leerfahrt-Kanten, werden die Kosten so definiert, dass sie den operativen Kosten für die Ausführung der repräsentierten Aktivitäten entsprechen. Die Kosten der Zirkulationsfluss-Kante entsprechen den fixen Fahrzeugkosten für die Benutzung eines zusätzlichen Fahrzeugs. Die Kantenkapazitäten sind so definiert, dass Fahrgastfahrt-, Ausrück- und Einrückfahrt-Kanten eine maximale Flusskapazität von 1 besitzen, während für alle anderen Kanten die maximale Flusskapazität auf die maximal erlaubte Fahrzeuganzahl gesetzt wird.

Aus dem Blickwinkel der Dienstplanung repräsentiert jeder Knoten im Planungsnetzwerk einen Ablösepunkt. Dies gilt dank der oben getroffenen Annahme, dass jede Fahrgastfahrt mit einem Ablösepunkt anfängt und endet, und weil im Depot selbst zu jedem Zeitpunkt ein Ablösen stattfinden darf. Weiterhin entspricht jede Kante außerhalb des Depots einem Dienstelement und ein gerichteter Pfad zwischen zwei beliebigen Knoten einem Dienststück. Das Dienststück ist gültig, wenn es alle Anforderungen an die Dienststückgültigkeit, wie z.B. minimale und maximale Dienststücklänge, erfüllt und keine Depot-Wartekanten beinhaltet, da ein Aufenthalt im Depot als Arbeitszeitunterbrechung zählt. Ein Dienst besteht aus einem oder mehreren solchen Dienststücken (Teilpfaden) und ist genau dann gültig, wenn alle Dienstregeln erfüllt sind. Jeder Dienst wird mit Personalkosten versehen.

Bei der integrierten Umlauf- und Dienstplanung werden solche Umläufe und Dienste gesucht, die jede Fahrgastfahrt-Kante genau einmal und alle sonstigen Kanten konsistent abdecken, d.h. falls eine Verbindungs- bzw. Depotkante von einem Fahrzeug benutzt wird, dann muss auch das entsprechende Dienststück in einem der gewählten Dienste enthalten sein. Dabei werden sowohl die Fahrzeugkosten (Kosten des Netzwerkflusses) als auch die Personalkosten (Kosten der Dienste) minimiert.

Handelt es sich um ein Problem mit mehreren Depots, dann wird ein solches Netzwerk (Netzwerkschicht) für jedes Depot erstellt. Jede Fahrgastfahrt wird in diesem Fall durch mehrere Kanten, eine in jeder betroffenen Netzwerkschicht, re-

präsentiert, wobei nur eine davon durch einen Pfad überdeckt werden darf. Es handelt sich um ein Mehrgüterfluss-Problem, in dem mehrere Güter mit minimalen Kosten durch das vorgestellte Netzwerk transportiert werden sollen. Eine Güterart ergibt sich aus der Menge der Fahrzeuge deines bestimmten Depots.

5.1.2 Mathematische Formulierung des MD-VCSP

Sei D die Menge aller Depots und N die Menge aller Fahrgastfahrten aus der Fahrplanmasse, wobei $N^d \subseteq N$ nur die Fahrten enthält, die von Fahrzeugen des Depots $d \in D$ bedient werden können. Für jedes Depot d wird ein oben beschriebenes Planungsnetzwerk $G^d = (V^d, A^d)$ definiert, wobei V^d die Menge aller Knoten und A^d die Menge aller Kanten ist. Sei $\bar{A}^d \subset A^d$ die Menge aller Kanten, die Aktivitäten abbilden, die sowohl ein Fahrzeug als auch einen Fahrer erfordern. Zu $\bar{A}^d$ gehören alle Fahrgastfahrt-, Leerfahrt- und Depotfahrt-Kanten sowie Wartekanten außerhalb der Depots. Sei $A^d(n) : N \rightarrow A^d$ eine Funktionen, die für jede Fahrgastfahrt $n \in N$ die zugehörige Fahrgastfahrt-Kante $(i, j) \in A^d$ aus dem Netzwerkschicht G^d liefert. Kann n nicht aus dem Depot d bedient werden, dann ist $A^d(n) = \emptyset$.

Für jede Kante $(i, j) \in A^d$ werden folgende Fahrzeugkosten c^d_{ij} definiert, entweder fixe Fahrzeugkosten für den Einsatz eines Fahrzeuges aus dem Depot d, wenn (i, j) die Zirkulationsfluss-Kante ist, oder operative Fahrzeugkosten, die beim Durchführen der durch (i, j) zu repräsentierenden Aktivität entstehen. Die maximale Flusskapazität u^d_{ij} auf einer Kante $(i, j) \in A^d$ ist gleich 1 für alle Fahrgastfahrt- und Depotfahrt-Kanten und u^d sonst, wobei u^d die maximale Fuhrparkgröße im Depot d ist.

Sei K^d die Menge aller gültigen Dienste aus dem Depot d. Die Kosten eines Dienstes $k \in K^d$ sind durch f^d_k definiert. Sie sind betriebsabhängig und können z.B. aus fixen, variablen Kosten bzw. Zuschlägen etc. bestehen. Weiterhin sei $K^d(i, j) \subset K^d$ die Menge von Diensten, die das Dienstelement beinhalten, das durch die Kante $(i, j) \in \bar{A}^d$ abgebildet ist. Umgekehrt sei $\bar{A}(k) \subset \bar{A}$ die Menge von Kanten, die die Dienstelemente des Dienstes $k \in K$ repräsentieren.

Eine ganzzahlige Variable y^d_{ij} entspricht dem Fluss auf der Kante $(i, j) \in A^d$ (jede Flusseinheit entspricht einem Fahrzeug, das die entsprechende durch (i, j) zu repräsentierende Aktivität ausführt). Weiterhin besagt eine binäre Entscheidungsvariable x^d_k, ob der Dienst $k \in K^d$ in der Lösung ausgewählt wird. Das integrierte Mehrdepot-Umlauf- und Dienstplanungsproblem kann als ein MIP wie folgt formu-

liert werden:

$$\text{(MD-VCSP)}: \quad \min \quad \sum_{d\in D}\sum_{(i,j)\in A^d} y_{ij}^d c_{ij}^d + \sum_{d\in D}\sum_{k\in K^d} x_k^d f_k^d \tag{5.1}$$

$$\text{s.t.} \quad \sum_{d\in D}\sum_{(i,j)\in A^d(n)} y_{ij}^d = 1 \quad \forall n \in N \tag{5.2}$$

$$\sum_{\{j:(j,i)\in A^d\}} y_{ji}^d - \sum_{\{j:(i,j)\in A^d\}} y_{ij}^d = 0 \quad \forall d \in D, \forall i \in V^d \tag{5.3}$$

$$\sum_{k\in K^d(i,j)} x_k^d - y_{ij}^d = 0 \quad \forall\, d \in D, \forall (i,j) \in \bar{A}^d \tag{5.4}$$

$$y_{ij}^d \text{ ganzzahlig}, \; 0 \le y_{ij}^d \le u_{ij}^d \quad \forall d \in D, \forall (i,j) \in A^d \tag{5.5}$$

$$x_k^d \in \{0;1\} \quad \forall d \in D, \forall k \in K^d \tag{5.6}$$

In der Zielfunktion (5.1) werden die gesamten Fahrzeug- und Personalkosten minimiert. Die Relation zwischen fixen und variablen Kosten sowie zwischen Fahrzeug- und Personalkosten hängt von den Anforderungen und Prioritäten der jeweiligen Optimierungsziele in einem konkreten Betrieb ab. Die Nebenbedingungen (5.2), (5.3) und (5.5) formulieren das MDVSP und stellen einen gültigen Umlaufplan sicher. Durch die *Flussbedingungen* (5.2) (*engl.: flow condition constraints*) wird garantiert, dass für jede Fahrgastfahrt genau eine ihrer Kanten in der optimalen Lösung enthalten ist, d.h. jede Fahrgastfahrt wird genau einmal bedient. Die *Flusserhaltungsbedingungen* (5.3) (*engl.: flow conservation constraints*) stellen sicher, dass für jeden Knoten die Summe aller eingehenden gleich der Summe aller ausgehenden Flüsse ist. Schließlich fordern die Nebenbedingungen (5.5), dass das Flussvolumen auf allen Kanten ganzzahlig ist und die vorgegebenen Kapazitäten nicht übersteigt. Die Kopplungsbedingungen (5.4) (*engl.: linking constraints*) sorgen für eine korrekte Kopplung zwischen resultierenden Umläufen und Diensten. Wird eine Kante im Planungsnetzwerk von einem oder mehreren Fahrzeugen (die Fahrzeugzahl entspricht dem Flussvolumen auf der Kante) benutzt, dann muss auch das dazugehörige Dienstelement in genauso vielen Diensten enthalten sein, d.h. jede Fahrzeugaktivität muss von einem aktiven Fahrer begleitet werden.

Eine Lösung der MIP-Formulierung (5.1)-(5.6) besteht aus einer Flusslösung für MDVSP, die durch die ganzzahligen Flusswerte auf den Netzwerkkanten beschrieben wird, und einer Menge von ausgewählten Diensten. Um aus der gefundenen Flusslösung gesuchte Fahrzeugumläufe abzuleiten, muss sie in Pfade zerlegt werden. Bei der oben vorgestellten Modellierung als Time-Space-Netzwerk sind solche Pfade nicht unbedingt disjunkt, d.h. sie können gemeinsame Kanten und/oder Knoten enthalten. Daher muss in jedem Knoten des Netzwerks eine *Flussdekomposition* durchgeführt werden, d.h. jeder in einen Knoten eingehenden Flusseinheit (falls vorhanden) wird eindeutig eine aus dem Knoten ausgehende Flusseinheit zugeordnet.

Damit der resultierende Umlaufplan mit dem gefundenen Dienstplan kompatibel bleibt, muss die Flussdekomposition „entlang" der Dienststücke realisiert werden, da jedes Dienststück einem Teil von einem Umlauf entspricht. Eine wichtige Eigenschaft des eingeführten Netzwerkmodells ist, dass alle zulässigen Dekompositionen äquivalente Umlaufpläne bezüglich der Kosten liefern. Somit kann aus dem optimalen Lösungsfluss immer ein optimaler und zu den gefundenen Diensten kompatibler Umlaufplan extrahiert werden.

Vergleicht man (MD-VCSP) mit der auf Seite 61 beschriebenen Formulierung von Huisman ([Huisman, 2004]), wird man feststellen, dass (MD-VCSP) „übersichtlicher" und kompakter ist. Man braucht nicht mehr zwischen verschiedenen Kantentypen zu unterscheiden. Somit können die Kopplungsbedingungen viel einfacher modelliert werden. Der entscheidende Unterschied liegt aber in der Anzahl der Variablen und Restriktionen. Da für jede Kante im Netzwerk eine Variable definiert wird, erreicht die Modellgröße von (MD-VCSP) dank der signifikant kleineren Anzahl von Verbindungskanten im Time-Space-Netzwerk (siehe Tabelle 5.1) nur einen Bruchteil von der Größe des Referenzmodells auf Seite 61. Somit können sowohl das Gesamtproblem, als auch die im Lösungsansatz abgeleiteten Unterprobleme viel schneller gelöst werden.

5.1.3 Column-Generation-Lösungsansatz

Das Gesamtmodell ist zu komplex, um direkt gelöst zu werden (schon wegen der hohen Komplexität der einzelnen Unterprobleme). Wir schlagen einen Lösungsansatz vor, der auf der Kombination von Column-Generation und Lagrange-Relaxation basiert (siehe auch [Freling, 1997] und [Huisman, 2004]). Auf die Vorteile dieser Kombination wurde bereits im Abschnitt 3.3 hingewiesen.

Durch die Relaxierung der Kopplungsbedingungen (5.4) zerfällt das Gesamtproblem in zwei Teile, ein Mehrdepot-Umlaufplanungsproblem (y-Teil) und ein Dienstplanungsproblem (x-Teil), die separat gelöst werden können. Die notwendige Kopplung der beiden Teile erfolgt implizit durch die Modifizierung der Zielfunktion. Während der x-Teil ein triviales Auswahlproblem darstellt, in dem Variablen mit negativen reduzierten Kosten in die Lösung ausgewählt werden, ist der y-Teil immer noch ein schwer zu lösendes Optimierungsproblem (MDVSP ist $\mathcal{NP}$-schwer, siehe [Bertossi et al., 1987]).

Aus diesem Grund schlagen wir vor, das Problem in zwei Schritten zu lösen (eine ähnliche Grundidee wird auch in [Huisman, 2004] verfolgt). In der ersten Stufe wird mit Hilfe von Column-Generation eine Menge guter Dienste generiert, wobei neben der Relaxation der Kopplungsbedingungen (5.4) auch das resultierende MDVSP vereinfacht wird, indem eine der beiden weiteren Restriktionsmengen

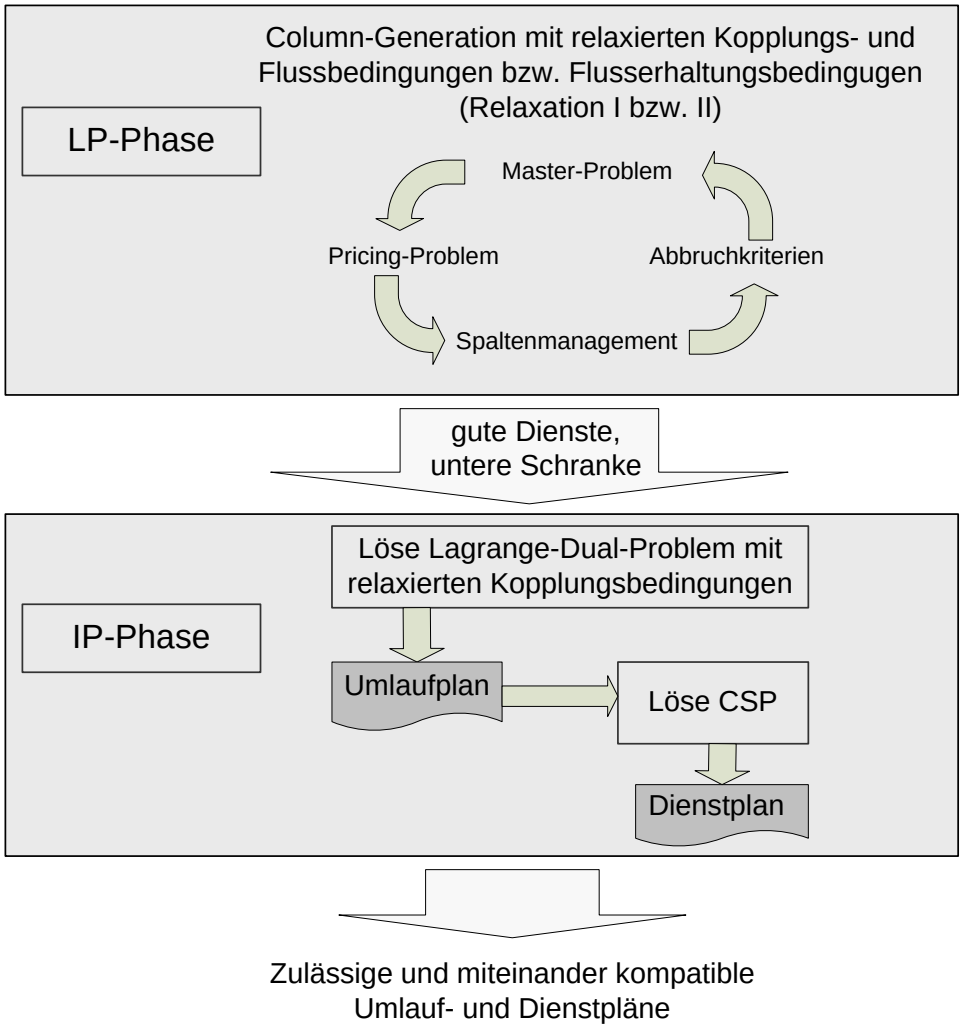

Abbildung 5.5: LP- und IP-Phasen des Lösungsprozesses

(5.2) oder (5.3) ebenfalls relaxiert wird. Je nach verbleibenden Nebenbedingungen reduziert sich der y-Teil zu mehreren kleinen SDVSP (Relaxation I) oder einem großen SDVSP (Relaxation II), die in polynomieller Zeit gelöst werden können. Wir bezeichnen diese Stufe als LP-Phase (siehe Abbildung 5.5). Können während der LP-Phase keine neuen Dienste mit negativen reduzierten Kosten gefunden werden, d.h. die untere Schranke kann nicht mehr verbessert werden, oder ist ein anderes Abbruchkriterium erreicht, wird die LP-Phase beendet.

In der zweiten Stufe des Lösungsansatzes wird eine möglichst gute zulässige Lösung gesucht. Wir bezeichnen diese Phase als IP-Phase (siehe Abbildung 5.5). Zunächst wird das Lagrange-Dual-Problem der „teureren" Relaxation (hier sind nur die Kopplungsbedingungen (5.4) relaxiert) gelöst, wobei man sich nur auf die bis dahin generierten Dienste beschränkt. Diese Relaxation ist deswegen teuer,

weil eines der resultierenden Lagrange-Unterprobleme ein $\mathcal{NP}$-hartes Mehrdepot-Umlaufplanungsproblem ist, was in jeder Iteration des Subgradienten-Verfahrens (bzw. Volume-Algorithmus) gelöst werden muss. Für den resultierenden Umlaufplan wird das traditionelle (umlaufbasierte) Dienstplanungsproblem gelöst. Somit wird eine korrekte Kopplung der beiden Pläne sichergestellt, allerdings kann die Optimalität der Gesamtlösung nicht garantiert werden.

Der grobe Ablauf des vorgeschlagenen Algorithmus kann schematisch folgendermaßen skizziert werden:

Schritt 0: Initialisierung (⇒Abschnitt 5.2)
Löse Umlauf- und Dienstplanung sequenziell, d.h. zuerst MDVSP und dann CSP. Die resultierenden Dienste bilden die Initialmenge der Dienste.
▷ zulässige Gesamtlösung (obere Schranke für MD-VCSP)
▷ initiale Spaltenmenge K'
Schritt 1: Löse eingeschränktes Master-Problem (⇒Abschnitt 5.3)
Löse das Lagrange-Dual-Problem einer Lagrange-Relaxation der ursprünglichen Formulierung mit der aktuellen Spaltenmenge K'.
▷ untere Schranke für MD-VCSP mit K'
▷ Lagrange-Multiplikatoren
Schritt 2: Löse Pricing-Problem (⇒Abschnitt 5.4)
Bestimme Dienste mit negativen reduzierten Kosten.
▷ neue Dienste
Schritt 3: Spaltenmanagement(⇒Abschnitt 5.5)
Erweitere die aktuelle Spaltenmenge K' um neue Dienste mit negativen reduzierten Kosten (falls vorhanden) und lösche ggf. aus K' vorhandene Spalten mit hohen positiven reduzierten Kosten.
▷ aktualisierte Spaltenmenge K'
Schritt 4: Abbruchkriterien
Gehe zum Schritt 5, falls mindestens eins der folgenden Kriterien erfüllt ist:
- keine Dienste mit negativen reduzierten Kosten gefunden,
- die Zahl der Iterationen hat ein Limit T_{max} erreicht,
- keine signifikante Verbesserung der unteren Schranke in der letzten T_ε Iterationen,

sonst gehe zum Schritt 1.
Schritt 5: Finde zulässige Lösung (⇒Abschnitt 5.6)
Finde zulässige Lösung durch eine Lagrange-Heuristik
▷ Umlaufplan
▷ Dienstplan

In den folgenden Abschnitten 5.2 - 5.6 werden die einzelnen Schritte des Algorithmus ausführlich diskutiert.

5.2 Initialisierung durch sequenzielle Planung

Die Hauptschleife von Column-Generation wird mit einer zulässigen Lösung des MD-VCSP initialisiert. Dafür greifen wir auf die sequenzielle Planung zurück und bestimmen eine zulässige Initialmenge von Spalten, indem zunächst das Mehrdepot-Umlaufplanungsproblem gelöst wird und anschließend passende Dienste durch das Lösen eines herkömmlichen Dienstplanungsproblems für berechneten Umläufe bestimmt werden.

5.2.1 Umlaufplanungsproblem

Zuerst müssen die optimalen Umläufe bestimmt werden. Die Formulierung des Mehrdepot-Umlaufplanungsproblem kann von der (MD-VCSP)-Formulierung auf Seite 78 direkt abgeleitet werden. Das MIP besteht aus dem y-Teil der Zielfunktion (5.1) unter der Erfüllung der Nebenbedingungen (5.2), (5.3) und (5.5). Es basiert auf dem oben beschriebenen Time-Space-Netzwerk und kann daher dank seiner kompakten Formulierung direkt mit Hilfe von Standard-Optimierungsbibliotheken wie z.B. MOPS® ([Suhl, 2000]) oder ILOG CPLEX® ([CPLEX, 2003]) exakt optimal gelöst werden (siehe [Kliewer, 2005]). Zuerst wird auf die Ganzzahligkeitsbedingung (5.5) verzichtet und die LP-Relaxation des Problems gelöst. Anschließend wird mit Hilfe eines Branch-and-Bound-Verfahrens eine optimale ganzzahlige Lösung bestimmt. Da aber die LP-Matrix für jedes Depot eine unimodulare Struktur aufweist, ist bereits die Lösung der LP-Relaxation in den meisten Fällen ganzzahlig. Die Lösungszeit des MDVSP liegt für alle getesteten Instanzen im Sekundenbereich und kann im Vergleich zur Gesamtlösungszeit vernachlässigt werden.

Ähnlich zum MD-VCSP liefert die resultierende Lösung der MDVSP-Formulierung eine optimale Flusslösung, die in Pfade zerlegt werden muss, um die gesuchten Fahrzeugumläufe davon abzuleiten. Solche Pfade sind im Time-Space-Netzwerk nicht unbedingt disjunkt, daher ist eine Flussdekomposition in jedem Knoten notwendig. Da das MDVSP separat und unabhängig von der Dienstplanung gelöst wird, genügt es hier, im Gegensatz zur Flussdekomposition beim MD-VCSP, z.B. eine einfache LIFO-Zuordnung von eingehenden und ausgehenden Flusseinheiten in jedem Knoten. Jede zulässige Flussdekomposition der optimalen Flusslösung ergibt einen optimalen Umlaufplan.

5.2.2 Dienstplanungsproblem

Bei der Dienstplanung werden die resultierenden Umläufe zunächst an den vordefinierten Ablösepunkten in atomare Arbeitseinheiten, Dienstelemente, zerlegt. Das

Ziel des Dienstplanungsproblems ist eine kostenminimale Menge von Diensten zu finden, so dass alle Dienstregeln erfüllt sind und jedes Dienstelement in genau einem Dienst enthalten ist. Im Folgenden werden die mathematische Formulierung und die unterschiedlichen Lösungsansätze vorgestellt, die im Rahmen der vorliegenden Arbeit untersucht wurden.

Mathematische Formulierung

Die am häufigsten verwendete Formulierung für CSP ist ein Set-Covering- bzw. Set-Partitioning Problem. Wir bevorzugen die Set-Covering-Formulierung, da sie gewöhnlich einfacher zu lösen ist. Allerdings können dabei *Mehrfachüberdeckungen* im Dienstplan entstehen. Das heißt, dass ein Dienstelement in mehreren Diensten enthalten ist. Dies kann allerdings nur dann auftreten, wenn es aus Kostengründen günstiger ist. Solche Mehrfachüberdeckungen können einfach aufgelöst werden, indem die betreffenden Dienstelemente nur in einem Dienst als auszuführende Arbeit bleiben und in allen anderen Diensten als Transfer (Mitfahrt als Fahrgast) betrachtet werden.

Sei K die Menge aller zulässiger Dienste und I die Menge aller Dienstelemente, die sich durch „das Schneiden" der vorliegenden Umläufe an Ablösepunkten ergeben. Menge $K(i) \in K$ ist die Menge aller Dienste, die das Dienstelement $i \in I$ beinhalten. Seien f_k die Kosten eines Dienstes $k \in K$ und x_k eine binäre Entscheidungsvariable, die angibt, ob Dienst $k \in K$ in der Lösung ist oder nicht. In der allgemeinen Set-Covering-Formulierung des Dienstplanungsproblems wird eine kostenminimale Menge an Diensten gesucht, die jedes Dienstelement mindestens einmal abdecken. Das entsprechende gemischt-ganzzahlige Programm lautet wie folgt:

$$\text{(CSP):} \qquad \min \quad \sum_{k \in K} f_k x_k \tag{5.7}$$

$$\text{s.t.} \quad \sum_{k \in K(i)} x_k \geq 1 \qquad \forall i \in I \tag{5.8}$$

$$x_k \in \{0,1\} \qquad \forall k \in K \tag{5.9}$$

Bei einem Umlaufplanungsproblem mit mehreren Depots wird jeder Umlauf einem Depot zugeordnet. Laut der am Anfang des Kapitels getroffenen Annahme werden auch die Fahrer einem Depot zugeordnet und können nur die Umläufe aus diesem Depot bedienen. Existieren keine globalen depotübergreifenden Nebenbedingungen, die den Dienstmix über mehrere Depots beschränken, dann kann das Dienstplanungsproblem für jedes Depot separat gelöst werden, d.h. es werden nur die für das jeweilige Depot relevanten Dienstelemente und Dienste betrachtet.

Im Falle globaler depotübergreifender Bedingungen muss (CSP) erweitert werden. Sei $K_m \subseteq K$ die Menge aller Dienste, die für eine globale Nebenbedingung

$m \in M$ relevant sind. Der Koeffizient eines Dienstes $k \in K_m$ in der Nebenbedingung $m \in M$ wird durch b_k^m bezeichnet. Seien $\underline{b}^m$ bzw. $\overline{b}^m$ die untere bzw. obere Schranke für eine Nebenbedingung $m \in M$. Folgende mathematische Formulierung erweitert das Modell (5.7)-(5.9) und kann für die Modellierung der meisten globalen Nebenbedingungen genutzt werden:

$$\underline{b}^m \leq \sum_{k \in K_m} b_k^m x_k \leq \overline{b}^m \qquad \forall m \in M \tag{5.10}$$

Im Folgenden werden einige Beispiele für globale Nebenbedingungen skizziert, bei denen nur obere Schranke $\overline{b}^m$ benutzt wird (vgl. [Freling, 1997], S. 63):

- maximaler prozentualer Anteil geteilter Dienste in der Lösung. Sei $0 \leq p \leq 1$ der maximal erlaubte Anteil geteilter Dienste in der Lösung und $K_s \subset K$ die Menge aller geteilten Dienste. Dann wird $\overline{b}^m = 0$, $K_m = K$, $b_k^m = -p$ für alle $k \in K_m \setminus K_s$ und $b_k^m = 1 - p$ für alle $k \in K_s$ gesetzt. Die resultierende Nebenbedingung sieht nach Umformung wie folgt aus: $\sum_{k \in K_s} x_k \leq p \sum_{k \in K} x_k$

- maximale durchschnittliche Arbeitszeit. Sei $\overline{a}$ die maximal erlaubte durchschnittliche Arbeitszeit und a_k die Arbeitszeit im Dienst k. Dann wird $\overline{b}^m = 0$, $K_m = K$, und $b_k^m = a_k - \overline{a}$ für alle $k \in K_m$ gesetzt.Die resultierende Nebenbedingung sieht nach Umformung wie folgt aus: $\sum_{k \in K} a_k x_k \leq \overline{a} \sum_{k \in K} x_k$

- maximal verfügbare Anzahl der Fahrer in jedem Depot, wobei jeder Dienst in einem bestimmten Depot starten und enden muss. Sei $\overline{b}^m$ die maximal verfügbare Anzahl der Fahrer im Depot m und K_m die Menge aller Dienste, die im Depot m starten und enden. Dann wird $b_k^m = 1$ für alle $k \in K_m$ gesetzt. Die resultierende Nebenbedingung sieht wie folgt aus: $\sum_{k \in K_m} x_k \leq \overline{b}^m$ für jedes Depot m.

Column-Generation-Ansatz

Da die Anzahl aller zulässigen Dienste schon bei kleinen Probleminstanzen mehrere Millionen erreichen kann, ist eine direkte Behandlung solcher mathematischen Probleme nicht möglich. Daher wird zur Lösung des Dienstplanungsproblems ein *Column-Generation*-Lösungsansatz eingesetzt, bei dem immer eine relativ kleine Vorauswahl an Spalten (Diensten) explizit behandelt wird und alle anderen implizit betrachtet werden.

Der Algorithmus wird zunächst mit einer *Initialmenge* von Diensten initialisiert, die eine zulässige Überdeckung aller Dienstelemente sicherstellen. Wir wählen diese Initialmenge aus der Menge der Teildienste (Dienste, die nur aus einem Dienststück bestehen) mit Hilfe einer Greedy-Heuristik aus.

Die Hauptschleife vom Column-Generation startet mit dem Lösen des *eingeschränkten Master-Problems* für die Initialmenge der Dienste. Wir untersuchen zwei Formulierungen des Master-Problems:

- *LP-Relaxation.* Dabei wird die Ganzzahligkeitsbedingung (5.9) der Originalformulierung (CSP) ignoriert. Zur Lösung der LP-Relaxation setzen wir den *Sifting-Algorithmus* der Optimierungsbibliothek CPLEX ein ([CPLEX, 2003]).

- *Lagrange-Relaxation.* Dabei werden die Überdeckungsbedingungen (5.8) und wenn vorhanden die globalen Bedingungen (5.10), mit Hilfe der Lagrange-Relaxation relaxiert. Die verbleibende Lagrange-Funktion ist ein triviales Auswahlproblem ohne weitere Nebenbedingungen und kann durch ein Auswählen von Diensten mit negativen reduzierten Kosten gelöst werden. Zur Lösung des Lagrange-Dual-Problems untersuchen wir zwei approximative Verfahren, nämlich das Subgradienten-Verfahren und den Volume-Algorithmus.

In der *Pricing-Phase* von Column-Generation werden mit Hilfe der dualen Informationen aus dem Master-Problem (Schattenpreise im Falle der LP-Relaxation bzw. Lagrange-Multiplikatoren im Falle der Lagrange-Relaxation) neue Dienste mit negativen reduzierten Kosten erzeugt. Dazu verwenden wir eine 2-Phasen-Prozedur, in der zunächst die zulässigen Dienststücke generiert werden. Sie werden nur einmal am Anfang des Verfahrens bestimmt (als alle möglichen, zulässigen Zusammensetzungen von nacheinander folgenden Dienstelementen eines Umlaufs) und deren reduzierten Kosten in jeder Pricing-Phase aktualisiert. Die zulässigen Dienststücke dienen als Eingabe für die zweite Phase, in der daraus die gesuchten Dienste mit negativen reduzierten Kosten konstruiert werden. Bei Dienstarten mit 2 Dienststücken pro Dienst werden dafür alle möglichen, zulässigen Kombinationen aus Dienststücken aufgezählt und die Zulässigkeit überprüft. Bei Dienstarten mit mehr als 2 Dienststücken wird das Pricing-Problem als ein *ressourcenbeschränktes Kürzeste-Wege-Problem* auf einem speziellen *Diensterzeugung-Netzwerk* formuliert.

Ganzzahlige Lösung

Nachdem die Hauptschleife von Column-Generation verlassen wird, muss eine ganzzahlige Lösung bestimmt werden. Dazu untersuchen wir zwei Lösungsansätze, nämlich eine Branch-and-Bound-Methode und eine primale (Such-)Heuristik, die jeweils Vor- und Nachteile aufweisen. Aus Gründen der Performance betrachten wir bei den beiden Methoden nur die Dienste, die nach der letzten Column-Generation-Iteration im eingeschränkten Master-Problem enthalten sind, d.h. es werden keine neuen Dienste während der IP-Phase nachgeneriert.

Das B&B kann für einige Probleme mit einer besonderen Struktur sehr schnell

eine gute oder sogar optimale Lösung finden. Allerdings ist die erforderliche Lösungszeit nicht im Voraus abschätzbar und kann in einem ungünstigen Fall extrem groß sein. In der Praxis wird sie oft begrenzt, sodass die B&B-Suche nach einem vordefinierten Zeitlimit abgebrochen und die bis dahin gefundene ganzzahlige Lösung zurückgegeben wird. Leider kann dabei aber nicht garantiert werden, dass zu diesem Zeitpunkt eine hinreichend gute bzw. überhaupt eine ganzzahlige Lösung gefunden worden ist.

Primale (Such-)Heuristiken starten dagegen mit einer zulässigen Lösung, die nach einem einfachen Schema konstruiert werden kann, und verbessern sie iterativ im Laufe des Verfahrens. Somit kann die Suche jederzeit mit einer zulässigen Lösung abgebrochen werden. Der Nachteil von solchen Heuristiken ist, dass sie keine gute Lösung, auch nach langer Suche, garantieren. Außerdem kann oft keine Aussage darüber getroffen werden, wie gut die aktuell gefundene Lösung wirklich ist, d.h. wie weit sie vom Optimum entfernt ist. In den meisten Fällen wird ein solches Verfahren nach einer vordefinierten Anzahl der Suchiterationen oder nach dem Überschreiten eines Zeitlimits abgebrochen. Auf der anderen Seite heißt das, dass es die vorgegebene Zeit „voll ausschöpft", auch wenn eine gute oder sogar optimale Lösung schon viel früher gefunden wurde.

Unsere Untersuchungen haben ergeben, dass sich keine eindeutige Aussage darüber treffen lässt, welches der beiden IP-Verfahren besser ist. Für einige CSP-Instanzen liefert B&B sehr schnell eine gute ganzzahlige Lösung, andere erweisen sich dagegen als besonders schwierig, sodass das B&B lange Zeit überhaupt keine zulässige Lösung finden kann, während die von uns untersuchte Heuristik auf Basis von Simulated-Annealing schon nach kurzer Zeit eine gute Lösung bestimmt. Leider kann im Voraus nicht eindeutig ersehen werden, ob ein Problem eher schwierig oder leicht für B&B ist, da dies nicht nur von der jeweiligen Problemgröße, sondern auch von der Problemstruktur und von vielen anderen Faktoren abhängt.

Um das Dilemma der Wahl der richtigen IP-Methode zu lösen und die Vorteile beider Verfahren zu vereinen, schlagen wir folgenden hybriden Algorithmus vor: Zuerst wird versucht eine ganzzahlige Lösung mit B&B zu finden, wobei wir dafür ein Zeitlimit einsetzen. Wird in der vorgegebenen Zeit keine zulässige Lösung gefunden bzw. ist sie von schlechter Qualität (die Lücke zur LP-Schranke ist zu groß), dann wird im zweiten Schritt eine Suchheuristik auf Basis von Simulated-Annealing ausgeführt, um eine gültige Lösung zu finden bzw. die vorliegende Lösung zu verbessern. Somit wird die Suchheuristik nur bei Bedarf nachgeschaltet und „belastet" die Lösungszeit von ganz einfachen Problemen nicht.

Das Grundschema unserer Suchheuristik basiert auf der lokalen Suchheuristik von [Jacobs and Brusco, 1995]. In jeder Iteration des Verfahrens wird eine neue Lösung aus der Nachbarschaft der aktuell vorliegenden Lösung ausgewählt. Wird

dabei eine bessere Lösung gefunden, ersetzt sie die alte und die Suche wird fortgeführt. Die Besonderheit von Simulated-Annealing ist es, dass mit einer gewissen Wahrscheinlichkeit auch verschlechternde Lösungen akzeptiert werden. Diese Akzeptanzwahrscheinlichkeit hängt von dem Ausmaß der Verschlechterung ab. Weiterhin wird sie durch den so genannten Temperaturparameter so kontrolliert, dass sie mit fortschreitendem Lösungsprozess gegen Null geht (siehe [Dowsland, 1993] für eine ausführliche Beschreibung der Simulated-Annealing Metaheuristik).

Um eine gute zulässige Startlösung für das Verfahren zu bekommen, gehen wir wie folgt vor: Zunächst wird aus den Diensten einer fraktionalen primalen Lösung eine partielle ganzzahlige Lösung konstruiert. Dabei werden in einem iterativen Verfahren die Dienste, die mindestens ein noch nicht abgedecktes Dienstelement beinhalten, ausgewählt und mit einer gewissen Wahrscheinlichkeit der partiellen Lösung hinzugefügt. Die Akzeptanzwahrscheinlichkeit ist umso höher, je größer der jeweilige fraktionale primale Wert ist. Ist die resultierende partielle Lösung nicht zulässig, d.h. sie deckt nicht alle Dienstelemente ab, dann wird die Zulässigkeit mit Hilfe einer *Konstruktionsheuristik* von [Caprara et al., 1999] wieder hergestellt. Dafür werden Dienste gemäß einer speziellen Bewertung *(score)* sortiert und nacheinander solange der partiellen Lösung hinzufügt, bis sie wieder zulässig wird.

Eine Nachbarschaftslösung bekommt man, indem man aus der aktuell vorliegenden Lösung einen Teil der Dienste durch andere ersetzt. Dafür werden zunächst einige Dienste aus der Lösung gelöscht. Ein Teil davon sind Dienste mit einem schlechten Bewertungswert (errechnet wie bei [Caprara et al., 1999]), der andere Teil ist ein Prozentsatz von Diensten, die aus der verbleibenden Menge zufällig ausgewählt und entfernt werden. Die resultierende partielle Lösung ist höchstwahrscheinlich nicht mehr zulässig. Um die Zulässigkeit wieder herzustellen wird wiederum die Konstruktionsheuristik von [Caprara et al., 1999] benutzt.

Das gesamte Suchverfahren wird mehrmals ausgeführt, wobei jedes Mal von einer anderen primalen Lösung gestartet wird. Die Menge unterschiedlicher primaler Lösungen wird ganz am Anfang mit Hilfe eines Volume-Algorithmus konstruiert.

5.3 Lösung des beschränkten Master-Problems

Zur Lösung des eingeschränkten Master-Problems setzen wir die Technik der Lagrange-Relaxation ein. Dabei werden zwei unterschiedliche Relaxationen der (MD-VCSP)-Formulierung (5.1)-(5.6) untersucht. Zur Lösung der Lagrange-Dual-Probleme untersuchen wir zwei Methoden, das Subgradienten-Verfahren und den Volume-Algorithmus.

5.3.1 Lagrange-Relaxationen

In der Column-Generation-Phase wird neben den Kopplungsbedingungen (5.4) auch eine der beiden Restriktionsmengen (5.2) oder (5.3) mit Hilfe der Lagrange-Relaxation relaxiert. Je nach verbleibenden Nebenbedingungen reduziert sich der y-Teil zu mehreren kleinen SDVSP (Relaxation I) oder einem großen SDVSP (Relaxation II), die in polynomieller Zeit gelöst werden können. Nachfolgend werden die beiden Varianten ausführlicher beschreiben.

Relaxation I
Wir assoziieren Lagrange-Multiplikatoren μ_{ij}^d mit jeder Kopplungsbedingung (5.4) und π_n mit jeder Flussbedingung (5.2). Durch die Lagrange-Relaxation werden die Restriktionen (5.2) und (5.4) aus der Formulierung gestrichen und die Zielfunktion (5.1) wird folgendermaßen angepasst:

$$\begin{aligned}\min \quad & \sum_{d\in D}\sum_{(i,j)\in A^d} y_{ij}^d c_{ij}^d + \sum_{d\in D}\sum_{k\in K^d} x_k^d f_k^d + \\ & + \sum_{d\in D}\sum_{(i,j)\in \bar{A}^d} \mu_{ij}^d \left(y_{ij}^d - \sum_{k\in K^d(i,j)} x_k^d \right) + \sum_{n\in N} \pi_n \left(1 - \sum_{d\in D}\sum_{(i,j)\in A^d(n)} y_{ij}^d \right)\end{aligned}$$

Die Lagrange-Funktion kann wie folgt formuliert werden:

$$\Phi^I(\mu,\pi) \;=\; \Phi_y^I(\mu,\pi) + \Phi_x^I(\mu) + \sum_{n\in N}\pi_n \tag{5.11}$$

mit

$$\Phi_y^I(\mu,\pi) \;=\; \min \quad \sum_{d\in D}\sum_{(i,j)\in A^d} y_{ij}^d \bar{c}_{ij}^d \;, \tag{5.12}$$

$$\sum_{\{j:(j,i)\in A^d\}} y_{ji}^d - \sum_{\{j:(i,j)\in A^d\}} y_{ij}^d = 0 \qquad \forall d\in D, \forall i \in V^d \tag{5.13}$$

$$0 \le y_{ij}^d \le u_{ij}^d \qquad \forall d\in D, \forall (i,j)\in A^d \tag{5.14}$$

und

$$\Phi_x^I(\mu) \;=\; \min \quad \sum_{d\in D}\sum_{k\in K^d} x_k^d \bar{f}_k^d \;, \tag{5.15}$$

$$x_k^d \in \{0;1\} \qquad \forall d\in D, \forall k \in K^d \tag{5.16}$$

wobei

$$\bar{c}_{ij}^d = \begin{cases} c_{ij}^d + \mu_{ij}^d - \pi_n & \text{für (i,j): } (i,j)\in \bar{A}_{ij}^d \text{ und } \exists n\in N : (i,j)\in A^d(n), \\ c_{ij}^d + \mu_{ij}^d & \text{für (i,j): } (i,j)\in \bar{A}_{ij}^d \text{ und } (i,j)\notin A^d(n), \forall n\in N, \\ c_{ij}^d & \text{für (i,j): } (i,j)\notin \bar{A}^d \end{cases}$$

reduzierte Kosten auf Kante $(i,j) \in A^d_{ij}$ und

$$\bar{f}^d_k = f^d_k - \sum_{(i,j)\in\bar{A}(k)} \mu^d_{ij}$$

reduzierte Kosten für Dienst $k \in K^d$ sind.

$\Phi^I_y(\mu,\pi)$ ist für gegebene Vektoren von Lagrange-Multiplikatoren μ und π ein Minimalkostenfluss-Problem (MCFP) bzw. ein separates Minimalkostenfluss-Problem für jedes Depot. Wir verzichten auf die explizite Ganzzahligkeitsbedingung der y-Variablen in (5.14), da jede Lösung von (5.12)-(5.14) dank der ganzzahligen Schranken auch ganzzahlig ist (siehe z.B. [Wolsey, 1998]). Das liegt an der unimodularen Struktur der Inzidenzmatrix. Somit kann $\Phi^I_y(\mu,\pi)$ mit Algorithmen der linearen Programmierung, wie z.B. Simplex-Methode bzw. ihren speziellen, effizienteren Versionen, in polynomieller Zeit gelöst werden.

$\Phi^I_x(\mu)$ stellt für einen gegebenen Vektor von Lagrange-Multiplikatoren μ ein triviales Auswahlproblem dar. Eine optimale Lösung bekommt man, indem man jede Spalte $k \in K^d$ mit negativen reduzierten Kosten in die Lösung auswählt (setze $x^d_k = 1$, wenn $\bar{f}^d_k < 0$ und $x^d_k = 0$ sonst).

Relaxation II

In der zweiten untersuchten Relaxation werden neben der Kopplungsbedingungen (5.4) die Flusserhaltungsbedingungen (5.3) relaxiert. Allerdings machen wir das nicht direkt, da das resultierende Lagrange-Unterproblem kein Flussproblem mehr darstellen würde. Stattdessen schreiben wird das Master-Problem zunächst wie folgt um.

Wir unterteilen die Menge aller Knoten V^d des Netzwerks G^d in Menge der Fahrgastfahrt-Knoten V^d_T und Menge der Depot-Knoten V^d_D, d.h. $V^d = V^d_T \cup V^d_D$. Sei $V^d_+(n) : N \to V^d_T$ eine Funktion, die für eine gegebene Fahrgastfahrt $n \in N$ ihren Startknoten im Netzwerk G^d liefert. Analog gibt die Funktion $V^d_-(n) : N \to V^d_T$ den Endknoten einer Fahrgastfahrt $n \in N$ im Netzwerk G^d zurück.

Die Flussbedingung (5.2) kann alternativ wie folgt umformuliert werden:

$$\sum_{d\in D}\sum_{i\in V^d_+(n)} \left(\sum_{j:(j,i)\in A^d} y^d_{ji} - \sum_{j:(i,j)\in A^d\setminus A^d(n)} y^d_{ij} \right) = 1 \qquad \forall n \in N \qquad (5.2a)$$

$$\sum_{d\in D}\sum_{i\in V^d_-(n)} \left(\sum_{j:(j,i)\in A^d\setminus A^d(n)} y^d_{ji} - \sum_{j:(i,j)\in A^d} y^d_{ij} \right) = -1 \qquad \forall n \in N \qquad (5.2b)$$

Die ursprüngliche Flussbedingung stellt sicher, dass genau eine der Fahrgastfahrt-Kanten, die eine gegebene Fahrgastfahrt $n \in N$ repräsentieren, benutzt wird. In

den Bedingungen (5.2a) und (5.2b) werden die Fahrgastfahrt-Kanten eliminiert. Stattdessen stellt die Bedingung (5.2a) sicher, dass die Summe der Bedarfe aller Knoten, die für eine gegebene Fahrgastfahrt $n \in N$ den Startpunkt repräsentieren, gleich Eins ist. Analog definiert die Bedingung (5.2b) für eine Fahrt $n \in N$ ein Angebot von Eins auf ihren Endknoten. Die Restriktionen (5.2a)-(5.2b) sind äquivalent zu den ursprünglichen Flussbedingungen (5.2).

Weiterhin unterscheiden wir bei der Flusserhaltungsbedingungen (5.3) zwischen solchen für Fahrgastfahrt- und Depot-Knoten:

$$\sum_{\{j:(j,i)\in A^d\}} y^d_{ji} - \sum_{\{j:(i,j)\in A^d\}} y^d_{ij} = 0 \qquad \forall d \in D, \forall i \in V^d_T \tag{5.3a}$$

$$\sum_{\{j:(j,i)\in A^d\}} y^d_{ji} - \sum_{\{j:(i,j)\in A^d\}} y^d_{ij} = 0 \qquad \forall d \in D, \forall i \in V^d_D \tag{5.3b}$$

Somit kann MD-VCSP alternativ durch die Zielfunktion (5.1) und Nebenbedingungen (5.2a), (5.2b), (5.3a), (5.3b), (5.4)-(5.6) formuliert werden. Nun werden in dieser Formulierung die Restriktionen (5.3a) und die Kopplungsbedingungen (5.4) relaxiert.

Wir assoziieren einen Lagrange-Multiplikator μ^d_{ij} mit jeder Kopplungsbedingung (5.4) und κ^d_i mit jeder Flussbedingung (5.3a). Durch die Lagrange-Relaxation werden diese Restriktionen aus der Formulierung gestrichen und die Zielfunktion (5.1) folgendermaßen angepasst:

$$\begin{aligned}\min \quad & \sum_{d\in D}\sum_{(i,j)\in A^d} y^d_{ij}c^d_{ij} + \sum_{d\in D}\sum_{k\in K^d} x^d_k f^d_k + \\ & + \sum_{d\in D}\sum_{(i,j)\in \bar{A}^d} \mu^d_{ij}\left(y^d_{ij} - \sum_{k\in K^d(i,j)} x^d_k\right) + \sum_{d\in D}\sum_{i\in V^d_T} \kappa^d_i \left(\sum_{\{j:(i,j)\in A^d\}} y^d_{ij} - \sum_{\{j:(j,i)\in A^d\}} y^d_{ji}\right)\end{aligned}$$

Die Lagrange-Funktion kann wie folgt formuliert werden:

$$\Phi^{II}(\mu,\kappa) \quad = \quad \Phi^{II}_y(\mu,\kappa) + \Phi^{II}_x(\mu) \tag{5.17}$$

mit

$$\Phi_y^{II}(\mu,\kappa) \;=\; \min \quad \sum_{d\in D}\sum_{(i,j)\in A^d} y_{ij}^d \bar{c}_{ij}^d \;, \tag{5.18}$$

$$\sum_{d\in D}\sum_{i\in V_+^d(n)}\left(\sum_{j:(j,i)\in A^d} y_{ji}^d - \sum_{j:(i,j)\in A^d\setminus A^d(n)} y_{ij}^d\right) - 1 = 0 \qquad \forall n\in N \tag{5.19}$$

$$\sum_{d\in D}\sum_{i\in V_-^d(n)}\left(\sum_{j:(j,i)\in A^d\setminus A^d(n)} y_{ji}^d - \sum_{j:(i,j)\in A^d} y_{ij}^d\right) + 1 = 0 \qquad \forall n\in N \tag{5.20}$$

$$\sum_{\{j:(j,i)\in A^d\}} y_{ji}^d - \sum_{\{j:(i,j)\in A^d\}} y_{ij}^d = 0 \qquad \forall d\in D, \forall i\in V_D^d \tag{5.21}$$

$$y_{ij}^d \text{ ganzzahlig, } 0\le y_{ij}^d \le u_{ij}^d \quad \forall d\in D,\ \forall (i,j)\in A^d \tag{5.22}$$

und

$$\Phi_x^{II}(\mu) \;=\; \min \quad \sum_{d\in D}\sum_{k\in K^d} x_k^d \bar{f}_k^d \;, \tag{5.23}$$

$$x_k^d \in \{0;1\} \qquad \forall d\in D, \forall k\in K^d \tag{5.24}$$

wobei

$$\bar{c}_{ij}^d = \begin{cases} c_{ij}^d + \mu_{ij}^d + \kappa_i^d - \kappa_j^D & \text{für (i,j): } i,j\in V_T^d, \\ c_{ij}^d + \mu_{ij}^d + \kappa_i^d & \text{für (i,j): } i\in V_T^d \text{ und } j\in V_D^d, \\ c_{ij}^d + \mu_{ij}^d - \kappa_j^d & \text{für (i,j): } i\in V_D^d \text{ und } j\in V_T^d, \\ c_{ij}^d & \text{für (i,j): } i\in V_D^d \text{ und } j\in V_D^d \end{cases}$$

reduzierte Kosten auf Kante $(i,j)\in A_{ij}^d$ und

$$\bar{f}_k^d = f_k^d - \sum_{(i,j)\in\bar{A}(k)} \mu_{ij}^d$$

reduzierte Kosten für Dienst $k\in K^d$ sind.

$\Phi_y^{II}(\mu,\kappa)$ stellt für gegebene Vektoren von Lagrange-Multiplikatoren μ und κ ein Minimalkostenfluss-Problem dar (im Gegensatz zu der Relaxation I, in der $\Phi_y^I(\mu,\pi)$ aus $|D|$ kleineren Minimalkostenfluss-Problemen besteht). $\Phi_x^{II}(\mu)$ und $\Phi_x^I(\mu)$ sind identisch.

5.3.2 Lagrange-Dual-Problem

Der Wert der Lagrange-Funktion $\Phi^I(\mu,\pi)$ bzw. $\Phi^{II}(\mu,\kappa)$ ist für gegebene Multiplikatoren eine untere Schranke der Original-Formulierung mit der aktuellen Menge an Diensten. Allerdings sind wir nicht an irgendeiner, sondern an einer möglichst

guten unteren Schranke interessiert. In dem Lagrange-Dual-Problem werden solche Lagrange-Multiplikatoren μ und π bzw. κ gesucht, die die Funktion $\Phi^I(\mu, \pi)$ bzw. $\Phi^{II}(\mu, \kappa)$ maximieren. Wir untersuchen zwei approximative Lösungsverfahren des Lagrange-Dual-Problems, das Subgradienten-Verfahren und den Volume-Algorithmus. Beide Methoden sind iterative Suchverfahren. Ein initialer Multiplikatoren-Vektor wird in jeder Iteration mit Hilfe einer Suchrichtung und einer Schrittweite angepasst, bis es keine bessere Lagrange-Funktion gefunden werden kann oder ein anderes Abbruchkriterium erreicht ist.

Da die beiden Lösungsverfahren nur approximativer Natur sind, sind die resultierenden Lagrange-Multiplikatoren nicht zwangsweise optimal. Das bedeutet, dass das duale Problem unzulässig sein kann. Praktisch heißt das, dass einige Spalten im aktuellen eingeschränkten Master-Problem negative reduzierte Kosten $\bar{f}_k^d$ haben können und somit im nächsten Pricing-Schritt vom Column-Generation-Verfahren erneut generiert werden. Um die duale Zulässigkeit wieder herzustellen, setzen wir folgende Greedy-Heuristik ein (vgl. [Huisman, 2004, S. 20]):

▶ Für jeden Dienst $k \in K^d$ mit $\bar{f}_k^d < 0$:
- berechne $\delta := \frac{\bar{f}_k^d}{|\bar{A}(k)|}$ und
- aktualisiere $\mu_{ij}^d := \mu_{ij}^d + \delta, \quad \forall (i,j) \in \bar{A}(k)$.

In den nächsten zwei Unterabschnitten werden die beiden untersuchten Lösungsverfahren für das Lagrange-Dual-Problem detaillierter beschrieben.

5.3.3 Subgradienten-Verfahren

Das Grundkonzept des Subgradienten-Verfahrens wurde im Unterabschnitt 3.2.2 bereits dargestellt. Die Version, die wir zur Lösung des Lagrange-Dual-Problems benutzen, weist allerdings einige Unterschiede auf. Wir untersuchten unterschiedliche, in der Literatur vorgestellte Modifikationen des Verfahrens. Die endgültige Version beinhaltet solche davon, die sich für die gegebene Problemstellung als sinnvoll erwiesen und zur Performance-Steigerung, wie Qualität der unteren Schranke, Konvergenzverhalten bzw. Anzahl der Iterationen sowohl lokal (im Subgradienten-Verfahren) als auch global (im Column-Generation), beigetragen haben.

Wir unterteilen das Subgradienten-Verfahren in 6 grundlegende Schritte: Nach der einmaligen *Initialisierung* (Schritt 1) wird die Hauptschleife des Verfahrens ausgeführt, in der die Schritte *Lösung der Lagrange-Funktion* (Schritt 2), *Bestimmung der neuen Suchrichtung* (Schritt 3), *Bestimmung der neuen Schrittweite* (Schritt 4) und *Aktualisierung der Multiplikatoren* (Schritt 5) solange nacheinander ausgeführt werden, bis ein *Abbruchkriterium* (Schritt 6) erreicht ist. Im Folgenden werden die einzelnen Schritte detaillierter diskutiert. Dabei beschränken wir uns

nur auf die Betrachtung der Lagrange-Relaxation I. Der Fall mit Benutzung der Lagrange-Relaxation II kann analog formuliert werden.

Initialisierung (Schritt 1)
Zunächst wird hier der Iterationszähler $t = 1$ gesetzt. Weiterhin müssen die Multiplikator-Vektoren μ^1 und π^1 initialisiert werden. Befindet man sich in der ersten Iteration des Column-Generation, dann werden sie mit einem Null-Vektor initialisiert, sonst setzen wir sie gleich den besten Multiplikatoren aus der vorangegangenen Column-Generation-Iteration. Somit startet die Suche nach optimalen Multiplikatoren jedes Mal von einer relativ guten Ausgangsposition *(Warmstart)*.

Als obere Schranke verwenden wir die zulässige Lösung für (MD-VCSP), die bereits in der Initialphase des Column-Generation durch das sequenzielle Lösen des Umlauf- und Dienstplanungsproblem berechnet wurde.

Lösung der Lagrange-Funktion (Schritt 2)
In diesem Schritt wird die Lagrange-Funktion $\Phi^I(\mu^t, \pi^t)$ gelöst, wobei μ^t und π^t die Multiplikator-Vektoren in der Iteration t des Subgradienten-Verfahrens sind.

$\Phi_y^I(\mu^t, \pi^t)$ stellt ein Minimalkostenfluss-Problem (MCFP) dar. Für diese Klasse der Optimierungsprobleme existieren effiziente Algorithmen. Bei der Wahl eines passenden MCFP-Algorithmus muss allerdings beachtet werden, dass $\Phi_y^I(\mu^t, \pi^t)$ in jeder Iteration des Subgradienten-Verfahrens gelöst wird, wobei sich jedes Mal nur der Kostenvektor des Problems ändert, während seine Struktur immer gleich bleibt. Der gesuchte Algorithmus muss also nicht nur ein separates Minimalkostenfluss-Problem, sondern eine Folge solcher Probleme mit änderndem Kostenvektor effizient lösen, d.h. er muss idealerweise in der Lage sein, die Informationen über die in der letzten Iteration errechnete Lösung ausnutzen bzw. von der alten Lösung starten können. Im Rahmen der vorliegenden Arbeit untersuchten wir für diese Zwecke unterschiedliche Varianten des Simplex-Algorithmus aus der Optimierungsbibliothek CPLEX und einen kombinierten Vorwährts-/Rückwärts-Auktions-Algorithmus aus [Freling, 1997]. Laut unserer Untersuchung wird eine Folge von MCFP-Problemen mit erwähnten Eigenschaften am effizientesten mit einem Netzwerk-Simplex-Algorithmus gelöst (auf detaillierte Ergebnisse dieser Analyse wird hier nicht weiter eingegangen). Zum gleichen Ergebnis kommen auch Frangioni und Manca in ihrer vor Kurzem präsentierten, umfangreichen Untersuchung gängiger, effizienter MCFP-Algorithmen unter einer ähnlichen Anforderung (siehe [Frangioni and Manca, 2006]).

$\Phi_x^I(\mu^t)$ stellt für einen gegebenen Vektor von Lagrange-Multiplikatoren μ^t ein triviales Auswahlproblem dar. Eine optimale Lösung bekommt man, indem man jede Spalte $k \in K^d$ mit negativen reduzierten Kosten in die Lösung auswählt (setze

$x_k^d = 1$, wenn $\bar{f}_k^d < 0$ und $x_k^d = 0$ sonst).

Suchrichtung (Schritt 3)
Für die Bestimmung der Suchrichtung d^t wird zunächst ein Subgradient s^t an der Stelle (μ^t, π^t) berechnet. Er repräsentiert eine Richtung, in der die euklidische Distanz zur optimalen Lösung verringert wird. Sei (x^t, y^t) die optimale Lösung der Lagrange-Funktion $\Phi^I(\mu^t, \pi^t)$, dann lässt sich s^t in unserem Fall wie folgt berechnen:

$$s_{ij}^{d\,[t]} = y_{ij}^{d\,[t]} - \sum_{k \in K^d(i,j)} x_k^{d\,[t]} \qquad \forall d \in D, (i,j) \in \bar{A}^d \tag{5.25}$$

$$s_n{}^{[t]} = 1 - \sum_{d \in D} \sum_{(i,j) \in A^d(n)} y_{ij}^{d\,[t]} \qquad \forall n \in N \tag{5.26}$$

Besitzt $\Phi^I(\mu^t, \pi^t)$ eine eindeutige optimale Lösung, dann ist s^t eine echte verbessernde Richtung (*engl.: steepest ascent direction*). Allerdings tritt sehr oft das Gegenteil auf. Besonders $\Phi_x^I(\mu^t)$ ist stark degeneriert: Es gib sehr viele Spalten, deren reduzierte Kosten sehr nah an Null liegen, sodass die Lagrange-Funktion eine sehr große Zahl von nahezu optimalen Lösungen besitzt, die durch Teilmengen dieser Spalten definiert sind. Schon für relativ kleine von uns getestete Instanzen liegt die Zahl der Spalten mit reduzierten Kosten $|\bar{f}_k^d| < 0,001$ in vielen Iterationen des Subgradienten-Verfahrens weit über 1000. Das führt dazu, dass sehr viele Subgradienten bei der Berechnung der Suchrichtung „aktiviert" werden und es keine Verbesserung der Zielfunktion in die echte verbessernde Richtung, sondern nur die euklidische Annäherung an die optimale Lösung garantiert werden kann. Es ist aber bekannt, dass die echte verbessernde Richtung durch die Minimum-Norm der Konvexkombination „aktiver" Subgradienten gegeben ist. Allerdings ist eine exakte Berechnung dieser Konvexkombination sehr zeitintensiv und erfordert das Lösen eines quadratischen Optimierungsproblems. Daher wird in der Praxis aus Effizienzgründen darauf verzichtet. In vielen Versionen des Verfahrens wird die Suchrichtung d^t einfach gleich dem Subgradienten s^t gesetzt, was wegen des Zickzack-Verhaltens zu einer sehr langsamen Konvergenz führen kann.

In unserer Version des Verfahrens verwenden wir eine alternative Formel zur Berechnung der Suchrichtung d^t. Aber zunächst betrachten wir, wie die Berechnung von Subgradienten modifiziert werden kann. Um das oben geschilderte Problem zu umgehen, verwenden wir bei der Berechnung der Subgradienten s^t anstatt des Lösungsvektor x seine Minimum-Norm, die allerdings heuristisch, mit Hilfe einer Prozedur von [Caprara et al., 1999] bestimmt wird. Sei $S \subseteq K$ die Menge von Spalten, deren reduzierte Kosten kleiner 0,001 sind, und $I(S) = \bigcup_{j \in S} I_j$ die Menge von Zeilen, die durch Spalten aus S überdeckt sind, wobei eine Spalte j die Zeilen I_j überdeckt. Die Spalten in S werden zunächst anhand ihrer reduzierten Kosten

absteigend sortiert und dann in dieser Reihenfolge auf Redundanz überprüft und ggf. aus S gelöscht. Die Spalte $j \in S$ ist redundant, wenn $I(S) = I(S \setminus \{j\})$ ist. Die in der resultierenden, nicht-redundanten Menge S verbleibenden Spalten bestimmen den neuen Lösungsvektor $\bar{x}$, der nun anstatt x bei der Berechnung der Subgradienten in (5.25) und (5.26) verwendet wird.

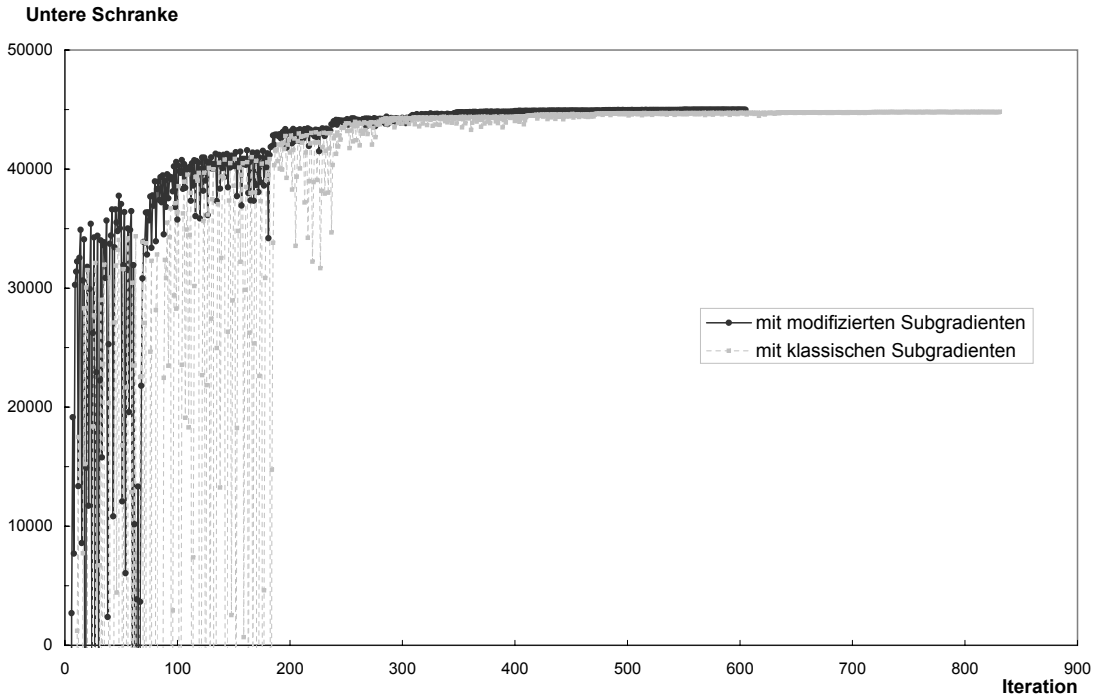

Abbildung 5.6: Klassische vs. modifizierte Berechnung von Subgradienten

Unserer Untersuchung zufolge trägt die Verwendung von modifizierten Subgradienten zu einer schnelleren Konvergenz des Subgradienten-Verfahrens bei. Die Abbildung 5.6 zeigt einen typischen Verlauf der Lagrange-Funktion während des Verfahrens mit und ohne Modifizierung der Subgradienten für eine Instanz mit 100 Fahrten und 4 Depots. Wie in der Abbildung deutlich zu erkennen ist, kann dank der Anpassung der Subgradienten eine bessere Konvergenz erreicht und das Zickzack-Verhalten in früheren Iterationen verringert werden. Somit terminiert das Subgradienten-Verfahren viel früher und oft mit einer besseren unteren Schranke.

Ein weiterer Punkt, der einer Verbesserung bedarf, ist die Formel zur Bestimmung der Suchrichtung. In der Literatur werden unzählige Alternativen der ursprünglichen Formel $d^t = s^t$ erforscht und im Hinblick auf Konvergenzverhalten und Lösungsqualität bewertet. In vielen Untersuchungen hat sich die Erkenntnis durchgesetzt, dass es günstig ist, bei der Neuberechnung von d^t die Suchrichtung der letzten Iteration zu berücksichtigen (vgl. [Crowder, 1976]):

$$d^t = \theta^t d^{t-1} + (1 - \theta^t) s^t, \tag{5.27}$$

wobei θ die Gewichtung der letzten Suchrichtung bestimmt und in der Standardvariante auf einen konstanten Wert $0 \leq \theta \leq 1$ gesetzt wird (sog. *Crowder Regel)*. In [Camerini et al., 1975] schlagen die Autoren eine ähnliche Update-Formel vor:

$$d^t = \theta^t d^{t-1} + s^t, \tag{5.28}$$

mit einer variablen, selbstjustierenden Wahl von θ^t:

$$\theta^t = \begin{cases} \|s^t\|/\|d^{t-1}\|, & \text{falls } s^t d^{t-1} \leq 0, \\ 0, & \text{sonst.} \end{cases} \tag{5.29}$$

Diese so genannte *modifizierte Camerini-Fratta-Maffioli*-Regel (CFM-Regel) garantiert, dass die neue Suchrichtung d^t mindestens so gut wie der Subgradient s^t ist. Sie führte auch in unseren Untersuchungen zu guten Ergebnissen. Die Abbildung 5.7 zeigt einen typischen Verlauf der Lagrange-Funktion während des Verfahrens unter Benutzung der klassischen Update-Formel und der CFM-Regel zur Berechnung der Suchrichtung (gleiche Testinstanz wie in der Abbildung 5.6). Wie in der Abbildung deutlich zu erkennen ist, kann hier ähnlich zur Modifizierung der Subgradienten (siehe Abbildung 5.6) eine bessere Konvergenz erreicht werden.

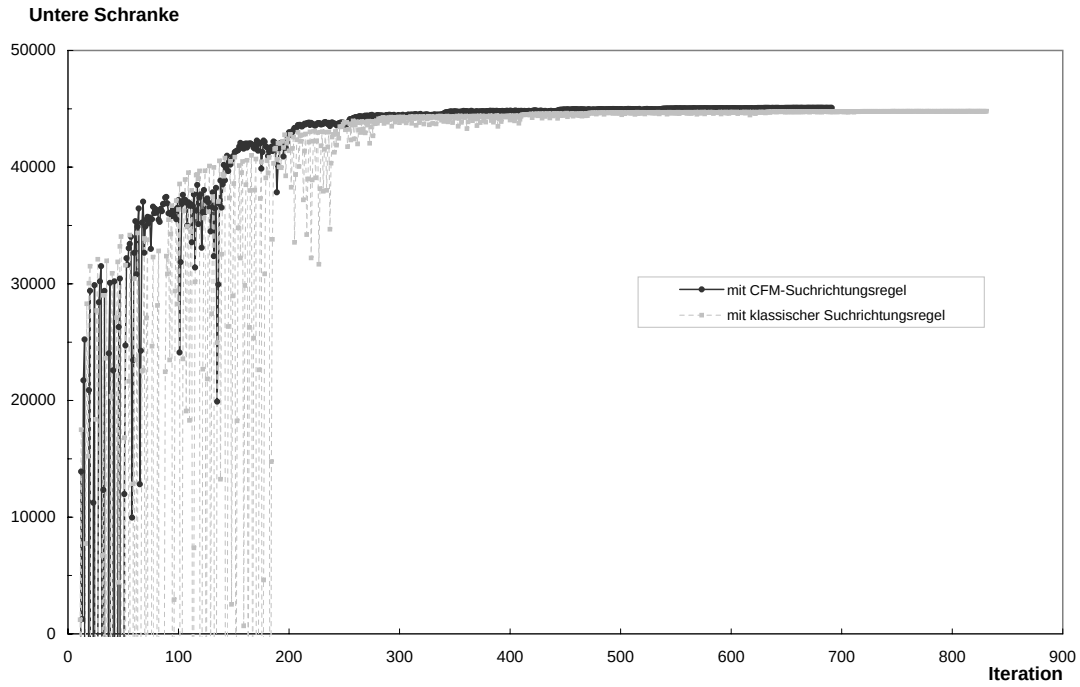

Abbildung 5.7: Klassische vs. CFM-Regel zur Berechnung der Suchrichtung

Die beiden vorgestellten Modifikation sind miteinander kombinierbar und führen zusammen, wie die Abbildung 5.8 veranschaulicht, zu noch besseren Ergebnissen. Neben einer schnelleren Konvergenz des Subgradienten-Verfahrens lässt sich

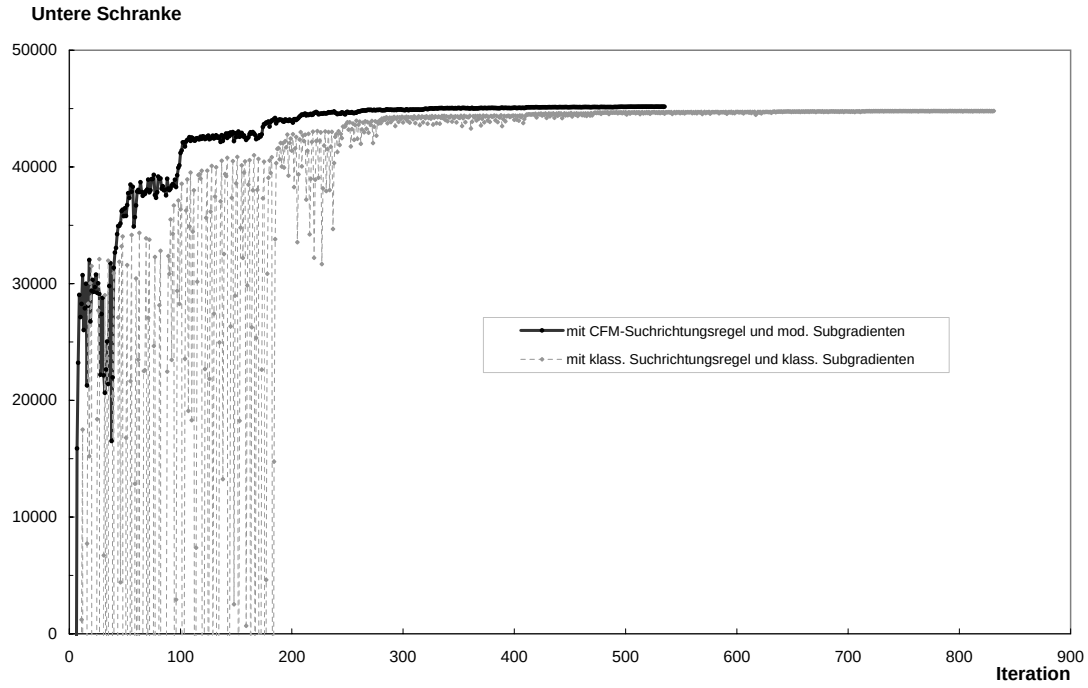

Abbildung 5.8: Kombination aus modifizierten Subgradienten und CFM-Regel zur Berechnung der Suchrichtung

dabei oft eine bessere untere Schranke bestimmen. In den meisten Fällen führt das dazu, dass der gesamte Column-Generation-Prozess schneller konvergiert und nicht nur weniger Zeit pro Iteration, sondern auch weniger Iterationen benötigt.

Schrittweite (Schritt 4)
Auch für die Berechnung der Schrittweite w^t gibt es verschiedene Varianten. In der Praxis wird sie meistens nach einer geometrischen Reihe unter Berücksichtigung der euklidischen Entfernung zur optimalen Lösung bestimmt. Sei z_{UB} eine obere Schranke für (MD-VCSP) und $z_{LR}(\mu^t, \pi^t)$ der aktuelle Zielfunktionswert der Lagrange-Funktion $\Phi^I(\mu^t, \pi^t)$, dann wird die Schrittweite wie folgt berechnet:

$$w^t = \lambda \frac{z_{UB} - z_{LR}(\mu^t, \pi^t)}{\|d^t\|^2}. \tag{5.30}$$

Dabei ist $0 \leq \lambda \leq 2$ ein Skalierungsparameter, der während des Verfahrens in der Regel verkleinert wird, wenn in einer Reihe von aufeinander folgenden Iterationen keine Verbesserung des Zielfunktionswertes erreicht wird (engl.: *stalling*). In unseren Tests hat sich folgende Strategie für λ durchgesetzt: Am Anfang wird λ mit 2 initialisiert und jedes Mal halbiert, wenn der Zielfunktionswert sich während der letzten 10 Iterationen nicht verbessert hat.

Weiterhin benutzen wir folgende, in [Beasley, 1993, S. 271] vorgeschlagene Ver-

feinerung der Schrittweitenberechnung:

$$\text{setze } d_{ij}^{d\,[t]} = 0, \quad \text{falls } \mu_{ij}^{d\,[t]} = 0 \text{ und } d_{ij}^{d\,[t]} < 0.$$

Diese Anpassung der Suchrichtungen wird vor der Berechnung der euklidische Distanz $\|d^t\|^2$ in (5.30) durchgeführt. Der Hintergedanke ist folgender: Ist $\mu_{ij}^{d\,[t]} = 0$ und $d_{ij}^{d\,[t]} < 0$, dann bekommt auch $\mu_{ij}^{d\,[t+1]}$ im nächsten Schritt den Wert Null, da die μ-Multiplikatoren nicht negativ sein dürfen (siehe Formel zur Aktualisierung der Lagrange-Multiplikatoren (5.31))[1]. Allerdings würde $d_{ij}^{d\,[t]}$ in die Berechnung von $\|d^t\|^2$ miteinfließen, obwohl der damit assoziierte Multiplikator $\mu_{ij}^{d\,[t+1]}$ unverändert bleibt.

Als weitere Modifikation wird oft vorgeschlagen, den Wert der oberen Schranke z_{UB} in (5.30) mit 1,05 zu multiplizieren. Dadurch sollte bei einer Annäherung an die optimale Lösung eine zu kleine Wahl von w^t verhindert werden, da sonst viele unnötige Iterationen durchführt werden können. Allerdings erwies sich dieser Vorschlag in unseren Tests als ungünstig. Der Grund dafür ist, dass der Wert der Lagrange-Funktion kaum an die obere Schranke herankommt. Das liegt daran, dass die obere Schranke einer sequenziellen Lösung der Umlauf- und Dienstplanung entspricht.

Aktualisierung der Lagrange-Multiplikatoren (Schritt 5)
Nachdem nun die Suchrichtungen d^t und die Schrittweite w^t berechnet worden sind, werden die Lagrange-Multiplikatoren wie folgt aktualisiert:

$$\mu_{ij}^{d\,[t+1]} = \max\left(0;\ \mu_{ij}^{d\,[t]} + w^t\, d_{ij}^{d\,[t]}\right) \qquad \forall d \in D, (i,j) \in \bar{A}^d \tag{5.31}$$

$$\pi_n{}^{[t+1]} = \pi_n{}^{[t]} + w^t d_n{}^{[t]} \qquad \forall n \in N \tag{5.32}$$

Dabei dürfen μ-Multiplikatoren nur nichtnegative, während π-Multiplikatoren beliebige Werte annehmen, da wir bei der Formulierung der Lagrange-Relaxation das „=”-Zeichen in Nebenbedingungen (5.4), mit denen μ assoziiert sind, durch „$\geq$” ersetzt haben.

Abbruchkriterien (Schritt 6)
Ist keines der Abbruchkriterien erfüllt, wird der Iterationszähler $t = t + 1$ gesetzt und mit dem Schritt 2 fortgefahren. Als Abbruchbedingungen sind in unserer Version des Verfahrens folgende Kriterien umgesetzt:

1.) keine der relaxierten Restriktionen ist verletzt: $s^t = 0$,

[1] Diese Verfeinerung gilt nicht für Suchrichtungen $d_n{}^{[t]}$, die mit π-Multiplikatoren assoziiert sind, da die π-Multiplikatoren auch einen negativen Wert annehmen können (siehe (5.32)).

2.) euklidische Distanz zum Optimum ist klein: $\|d^t\| < 0,001$,

3.) $\lambda < 0,005$,

4.) Iterationslimit erreicht: $t > 1000$,

5.) obere Schranke $\approx$ untere Schranke: $z_{UB} - z_{LR}(\mu^t, \pi^t) < 0,01$.

In den meisten Fällen wird das Subgradienten-Verfahren wegen der Erfüllung der Abbruchbedingung 3 verlassen. Von weiteren Iterationen sind dann keine großen Verbesserungen der unteren Schranke z_{LR} zu erwarten.

5.3.4 Volume-Algorithmus

Volume-Algorithmus wird in [Barahona and Anbil, 2000] als eine Erweiterung des klassischen Subgradienten-Verfahrens präsentiert. Er besitzt bessere Abbruchbedingungen und berechnet zusätzlich zu der dualen Lösung auch eine Approximation der primalen Lösung. Allerdings weisen die Autoren darauf hin, dass nicht bei allen Optimierungsproblemen der Einsatz des Volume-Algorithmus zum Erfolg führt. Gute Ergebnisse werden bei Problemklassen mit einer bestimmten Struktur beobachtet. Im Rahmen dieser Arbeit untersuchen wir, wie weit der Volume-Algorithmus für unsere Problemstellung eingesetzt werden kann. Außerdem werden einige in der Literatur vorgestellte Varianten und Erweiterungen der ursprünglichen Version (z.B. [Bahiense et al., 2002], [Barahona and Anbil, 2002]) untersucht.

Das Grundkonzept des Volume-Algorithmus wurde bereits im Unterabschnitt 3.2.3 erläutert. Der Ablauf besteht aus den gleichen Schritten wie das Subgradienten-Verfahren, die sich allerdings inhaltlich unterscheiden. Im Folgenden wird die Variante des Algorithmus vorgestellt, die sich für unsere Problemstellung als günstig erwiesen hat. Wir bezeichnen als $(\bar{\mu}, \bar{\pi})$ die aktuell besten Lagrange-Multiplikatoren, d.h. solche, mit denen die beste untere Schranke erreicht wird. Weiterhin sei $(\bar{x}, \bar{y})$ der primale Lösungsvektor.

Initialisierung (Schritt 1)
Ähnlich zum Subgradienten-Verfahren wird der Iterationszähler $t = 1$ gesetzt. In der ersten Iteration des Column-Generation-Verfahrens werden $(\bar{\mu}, \bar{\pi})$ mit einem Null-Vektor initialisiert, sonst setzen wir sie gleich den besten Multiplikatoren aus der vorangegangenen Column-Generation-Iteration $\mu_1 = \bar{\mu}$ und $\pi_1 = \bar{\pi}$.

Lösung der Lagrange-Funktion (Schritt 2)
Dieser Schritt ist identisch mit dem entsprechendem Schritt des Subgradienten-Verfahrens (siehe Seite 93). Zusätzlich werden die besten Multiplikatoren aktualisiert, wenn die neue untere Schranke besser als die bisher beste ist:

$$\text{Ist } z_{LR}(\mu^t, \pi^t) > z_{LR}(\bar{\mu}, \bar{\pi}), \text{ dann aktualisiere } \bar{\mu} = \mu^t \text{ und } \bar{\pi} = \pi^t.$$

Suchrichtung (Schritt 3)
Hier werden zunächst Subgradienten s^t an der Stelle (μ^t, π^t) wie in (5.25) und (5.26) berechnet. Bei der Berechnung der Suchrichtung d^t werden auch bei Volume-Algorithmus die früheren Suchrichtungen mitberücksichtigt:

$$d^t = \alpha s^t + (1 - \alpha) d^{t-1}. \tag{5.33}$$

Für die Bestimmung des Gewichtungsfaktors $0 \leq \alpha \leq 1$ benutzen wir die Prozedur, die in [Barahona and Anbil, 2000] dafür vorgeschlagen wird und ursprünglich aus dem Konjugierten-Subgradient-Verfahren (*engl.: conjugate subgradient method*) [Wolf, 1975] stammt. Zunächst wird ein solches α_{opt} berechnet, das das eindimensionale quadratische Problem $\|\alpha s^t + (1 - \alpha) d^{t-1}\|$ minimiert (es kann als quadratische Gleichung formuliert und entsprechend gelöst werden). Ist $\alpha_{opt} < 0$, dann wird $\alpha = \alpha_{max}/10$ und sonst $\alpha = \min\{\alpha_{opt}, \alpha_{max}\}$ gesetzt. Dabei starten wir mit $\alpha_{max} = 0{,}1$ und verringern es im Laufe des Verfahrens mit $\alpha_{max} = \max\{\alpha_{max}/2, 10^{-5}\}$, wenn die Verbesserung der aktuell besten unteren Schranke $z_{LB}(\bar{\mu}, \bar{\pi})$ in der letzten 80 Iterationen weniger als 1% betrug. Somit wird die Genauigkeit der primalen Lösung zum Schluss erhöht.

Mit dem gleichen Gewichtungsfaktor α werden auch die primalen Lösungswerte bestimmt:

$$\bar{y} = \alpha y^t + (1 - \alpha)\bar{y} \qquad \text{und} \tag{5.34}$$

$$\bar{x} = \alpha x^t + (1 - \alpha)\bar{x} \quad , \tag{5.35}$$

wobei (x^t, y^t) die optimale Lösung der Lagrange-Funktion $\Phi^I(\mu^t, \pi^t)$ ist. Somit wird die Approximation der primalen Lösung $(\bar{x}, \bar{y})$ als eine konvexe Kombination der lokalen Lösungen aus jeder Iteration berechnet. Diese Überlegung basiert auf der Dualitätstheorie (siehe [Barahona and Anbil, 2000]).

Schrittweite (Schritt 4)
Vor der Berechnung der Schrittweite setzen wir zunächst, ähnlich zum Subgradienten-Verfahren, solche $d_{ij}^{d\,[t]}$ auf Null, für die sich die zugehörigen Lagrange-Multiplikatoren nicht ändern, d.h. wenn $\bar{\mu}_{ij}^d = 0$ (beachte geänderte Formel zur Aktualisierung der Multiplikatoren) und $d_{ij}^{d\,[t]} < 0$. Die neue Schrittweite wird wie folgt bestimmt:

$$w^t = \lambda \frac{T - z_{LR}(\bar{\mu}, \bar{\pi})}{\|d^t\|^2}. \tag{5.36}$$

Wie in [Barahona and Anbil, 2002] empfohlen, benutzen wir im Zähler anstatt der Differenz zwischen der oberen und unteren Schranke die Differenz zwischen

einem speziellen Zielwert T und der bisher besten unteren Schranke $z_{LR}(\bar{\mu}, \bar{\pi})$. Wir starten mit einem kleinen Ziel und erhöhen es auf $1,05\, z_{LR}(\bar{\mu}, \bar{\pi})$ jedes Mal, wenn $z_{LR}(\bar{\mu}, \bar{\pi}) \geq 0,95T$ ist.

Bei der Bestimmung des Skalierungsparameters λ werden drei Typen von Iterationen *rot, gelb und grün* unterschieden. Je nach der Iteration wird λ wie folgt errechnet:

- Ist $z_{LR}(\mu^t, \pi^t) < z_{LR}(\bar{\mu}, \bar{\pi})$,
  - dann wird solche Iteration *rot* genannt. Nach einer Sequenz von 10 aufeinander folgenden roten Iteration wird der Skalierungsparameter verringert: $\lambda = \max\{0,67\lambda\,;\ 0{,}0005\}$;
- sonst berechne $D := s^t d^{t-1}$
  - ist $D < 0$, wird solche Iteration als *gelb* bezeichnet. Bei zwei aufeinander folgenden gelben Iterationen wird der Skalierungsparameter leicht vergrößert: $\lambda = \min\{1,1\lambda\,;\ 2\}$,
  - sonst nennen wir die Iteration *grün* und verdoppeln den Skalierungsparameter: $\lambda = \min\{2\lambda\,;\ 2\}$.

Wie man sieht, kann hier im Gegensatz zum Subgradienten-Verfahren der Skalierungsparameter λ im Laufe des Verfahrens auch wieder ansteigen und somit die Schrittweite erhöhen.

Aktualisierung der Lagrange-Multiplikatoren (Schritt 5)
Die Aktualisierung der Lagrange-Multiplikatoren geschieht im Volumen-Algorithmus im Gegensatz zum Subgradienten-Verfahren nicht ausgehend von den Multiplikatoren der vorangegangenen Iteration, sondern ausgehend von den bisher besten Multiplikatoren (vgl. (5.31)-(5.32)):

$$\mu_{ij}^{d\,[t+1]} = \max\left(0;\ \bar{\mu}_{ij}^{d} + w^t\, d_{ij}^{d\,[t]}\right) \qquad \forall d \in D, (i,j) \in \bar{A}^d \tag{5.37}$$

$$\pi_n{}^{[t+1]} = \bar{\pi}_n + w^t d_n{}^{[t]} \qquad \forall n \in N \tag{5.38}$$

Das heißt, die Aktualisierung der Multiplikatoren findet nur statt, wenn eine bessere untere Schranke im Schritt 2 gefunden wird, ansonsten geht die Suche in der Nachbarschaft des aktuellen *Stabilitätszentrums* $(\bar{\mu}, \bar{\pi})$ weiter. Diese Vorgehensweise wird in [Barahona and Anbil, 2002] vorgeschlagen. Die Autoren imitieren damit die Bundle-Methode [Lemaréchal, 1989], ohne dabei in jeder Iteration ein quadratisches Optimierungsproblem lösen zu müssen.

Abbruchkriterien (Schritt 6)
Sei $z(\bar{x}, \bar{y})$ primaler Zielfunktionswert, der sich durch das Einsetzen der primalen

Lösung $(\bar{x}, \bar{y})$ in die Zielfunktion (5.7) der (MD-VCSP)-Formulierung ergibt. Der Volume-Algorithmus terminiert, falls mindestens eine der folgenden Abbruchbedingungen erfüllt wird:

1.) Differenz zwischen der primalen Lösung und unterer Schranke ist kleiner als 1% und jede relaxierte Nebenbedingung ist maximal um 0,02 verletzt:

$$\left|\frac{z(\bar{x},\bar{y}) - z_{LR}(\bar{\mu},\bar{\pi})}{z_{LR}(\bar{\mu},\bar{\pi})}\right| < 0,01 \quad \text{und} \quad s^t_{\max} < 0,02$$

wobei $s^t_{\max} = \max\left\{ s_{ij}^{d\,[t]} : \forall d \in D, (i,j) \in \bar{A}^d \quad \text{und} \quad |s_n{}^{[t]}| : \forall n \in N \right\}$ ist,

2.) Verbesserung von $z_{LR}(\bar{\mu},\bar{\pi})$ in der letzten 100 Iterationen ist kleiner als 0,1%,

3.) Iterationslimit ist erreicht: $t > 1000$,

4.) obere Schranke $\approx$ untere Schranke: $UB - z_{LR}(\bar{\mu},\bar{\pi}) < 0,01$,

5.4 Lösung des Pricing-Problems

Nachdem das eingeschränkte Master-Problem, wie im vorangegangenen Abschnitt beschrieben, gelöst wird, werden in der Pricing-Phase neue Dienste gesucht, die die Lösung des aktuellen Problems verbessern können. Jeder Dienst besteht aus einem oder mehreren Dienststücken. Ein Dienststück repräsentiert dabei die Arbeit, die ein Fahrer auf einem Fahrzeug am Stück ausführt, d.h. ohne sie durch einen Fahrerwechsel oder eine gesetzlich vorgeschriebene Pause unterbrechen zu müssen. Bei dem CSP (siehe Unterabschnitt 5.2.2) kann die Menge der Dienststücke durch „das Schneiden" der zuerst definierten Umläufe in gültige Abschnitte gebildet werden. Bei dem integrierten Umlauf- und Dienstplanungsproblem sind dagegen die Umläufe im Voraus nicht bekannt, so dass viel mehr Freiheitsgrade bei der Bildung der Dienststücke existieren. Die Anzahl der zulässigen Dienststücke und Dienste ist bei VCSP viel größer als bei CSP.

Freling schlägt vor, die gesuchten Dienste in zwei Schritten zu generieren (siehe [Freling, 1997]). Dabei werden zuerst zulässige Dienststücke generiert, die dann als Grundlage für die zweite Phase verwendet werden, wo sie zu Diensten kombiniert werden. Wir verfolgen diese Vorgehensweise in unserem Ansatz. Die Menge zulässiger Dienststücke wird mit Hilfe eines Algorithmus zur Berechnung kürzester Wege in einem speziellen Dienststückerzeugungsgraphen konstruiert (siehe Unterabschnitt 5.4.1). Für die Erstellung der Dienste schlagen wir zwei Methoden vor. Bei Dienstarten mit zwei Dienststücken pro Dienst werden dafür alle möglichen,

zulässigen Kombinationen aus Dienststücken aufgezählt und die Zulässigkeit überprüft. Dabei wurde eine spezielle Datenstruktur entwickelt, die die zeitintensive Zulässigkeitsüberprüfung der Dienste in jede Iteration überflüssig macht und somit zu einer signifikanten Laufzeitverbesserung des gesamten Verfahrens führt (siehe Unterabschnitt 5.4.2). Bei Dienstarten mit mehr als zwei Dienststücken pro Dienst werden die gesuchten Dienste mit Hilfe eines Algorithmus zur Berechnung ressourcenbeschränkter kürzester Wege auf einem speziellen Diensterzeugungsgraphen konstruiert (siehe Unterabschnitt 5.4.3).

5.4.1 Erzeugung von Dienststücken

Zunächst definieren wir einen *Dienststückerzeugungsgraphen* $\widetilde{G} = (\widetilde{V}, \widetilde{A})$, der von dem für VCSP zugrunde liegenden Planungsnetzwerk G (siehe Unterabschnitt 5.1.1) direkt abgeleitet wird, indem alle Kanten, die nicht Bestandteil eines Dienststückes sein können, aus G gelöscht werden, d.h. $\widetilde{V} = V$ und $\widetilde{A} = \bar{A}$. Zu solchen Kanten zählen die Zirkulationskante und die Kanten, die Aufenthalte im Depot repräsentieren, d.h. Wartekanten im Depot, da jeder Depotaufenthalt als Dienststückunterbrechung gilt (siehe Annahme und Folgerung auf Seite 71). Somit repräsentiert jeder gerichteter Pfad zwischen zwei Knoten im $\widetilde{G}$ ein potentielles Dienststück[2]. Da jeder Knoten in $\widetilde{G}$ einen Zeitpunkt im Raum repräsentiert, kann die Dauer eines Dienststückes als die Differenz zwischen den Zeitpunkten seines letzten und seines ersten Knotens errechnet werden. Ein Dienststück ist gültig, wenn seine Dauer zwischen den vordefinierten Mindest- und Höchstdauer liegt (siehe Annahme auf Seite 70). Das entsprechende Knotenpaar bezeichnen wir als *kompatibel*.

Die Menge gültiger Dienststücke wird durch alle mögliche Pfade zwischen jedem kompatiblen Knotenpaar in $\widetilde{G}$ repräsentiert. Allerdings ist es nicht notwendig alle Dienststücke zu erzeugen, da wir nur daran interessiert sind, mindestens einen Dienst mit negativen reduzierten Kosten zu finden, um das Column-Generation-Verfahren fortsetzen zu können. Auf der anderen Seite, ist das Ziel des gesamten Verfahrens eine solche Lösung zu finden, die nicht mehr verbessert werden kann, d.h. für die kein neuer Dienst mit negativen reduzierten Kosten existiert. Aus diesem Grund schlägt Freling vor, für jedes kompatible Knotenpaar nur ein Dienststück zu generieren und zwar das, mit den kleinsten reduzierten Kosten (siehe [Freling, 1997]). Er bewies, dass diese Auswahl an Dienststücken ausreichend ist, um das Column-Generation-Verfahren mit neuen Diensten fortzusetzen bzw. es ggf. abzubrechen, wenn daraus keine Dienste mit negativen reduzierten Kosten gebildet werden können. Zwar basiert der Dienststückerzeugungsgraph von Freling

[2] $\widetilde{G}$ ist im Gegensatz zu G nicht zusammenhängend, somit existiert nicht zwischen jedem zeitlich kompatiblen Knotenpaar ein gerichteter Weg.

auf einem Connection-basierten Netzwerk, aber seine Idee lässt sich auch auf unsere TSN-basierte Modellierung übertragen.

Ersetzt man die Kosten auf Kanten im $\widetilde{G}$ durch ihre reduzierte Kosten:

$$\bar{c}_{ij}^{d} = c_{ij}^{d} - \mu_{ij}^{d} \qquad \forall (i,j) \in \widetilde{A},$$

wobei μ die mit den Kopplungsbedingungen (5.4) assoziierten Lagrange-Multiplikatoren sind, dann entsprechen die Kosten eines Pfades in $\widetilde{G}$ den reduzierten Kosten des Dienststückes, das dadurch repräsentiert wird (die Kosten eines Pfades setzten sich aus Kosten der darin enthaltenen Kanten zusammen). Der Pfad mit den kleinsten reduzierten Kosten kann somit mit einem Kürzeste-Wege-Algorithmus bestimmt werden. Da man aber an den besten Dienststücken für jedes kompatible Knotenpaar interessiert ist, setzt man einen Algorithmus zur Berechnung kürzester Wege zwischen allen Knotenpaaren (*engl.: all-pair-shortest-path-algorithm*) ein, wie z.B. *Floyd/Warshall-Verfahren* oder *Johnson-Algorithmus* ([Cormen et al., 2000]).

Handelt es sich um eine Mehrdepot-Variante des VCSP, dann besteht das ursprüngliche Planungsnetzwerk G aus jeweils einer unabhängigen Netzwerkschicht G^d für jedes Depot $d \in D$. In diesem Fall wird für jede dieser Netzwerkschichten G^d ein Dienststückerzeugungsgraph $\widetilde{G^d}$ definiert und darauf ein Kürzeste-Wege-Problem zwischen allen Knotenpaaren gelöst.

5.4.2 Erzeugung von Diensten durch Aufzählung

In der zweiten Phase des Verfahrens werden aus zuvor gefundenen Dienststücken Dienste erstellt. Bei Dienstarten mit zwei Dienststücken pro Dienst verfolgen wir den Vorschlag von Huisman (siehe [Huisman, 2004, S. 85]), in dem potentielle Dienste durch die Aufzählung aller Kombinationen von Dienststücken gebildet werden. Erfüllt so eine Kombination alle erforderlichen Dienstregeln entspricht sie einem zulässigen Dienst. Besitzt dieser Dienst negative reduzierte Kosten, dann kann er dem eingeschränkten Master-Problem hinzugefügt werden.

Die Überprüfung eines potentiellen Dienstes auf die Zulässigkeit ist relativ zeitintensiv. Die Anzahl potentieller Dienste beträgt im allgemeinen $\mathcal{O}(N^4)$, wobei N die Anzahl der Fahrten ist[3]. Somit kann das Pricing Problem sehr viel Zeit in Anspruch nehmen. Dies bestätigt auch die Analyse von bereits existierenden Ansätzen zur integrierten Umlauf- und Dienstplanung (siehe Abschnitt 4.3). Der Anteil des Pricing-Teils beträgt in Column-Generation-basierten Algorithmen aus der Literatur zwischen 60% und 90% von der Gesamtlaufzeit.

[3]Die Menge potentieller Diensten beträgt $\mathcal{O}(|P|^2)$, wobei |P| die Anzahl gültiger Dienststücke ist. Da zwischen jedem kompatiblen Knotenpaar ein Dienststück erzeugt werden kann, erreicht die Anzahl der Dienststücke die Größenordnung $\mathcal{O}(N^2)$.

Im Rahmen dieser Arbeit wurde eine neue Methode entwickelt, die für Probleminstanzen bis zu einer gewissen Größe und mit Dienstarten mit maximal zwei Dienststücken das Pricing-Problem signifikant schneller löst. Die Hauptidee basiert auf der Beobachtung, dass für die Zulässigkeit eines Dienstes, unter getroffenen Annahmen, der eigentliche Verlauf seiner Dienststücke absolut irrelevant ist. Entscheidend ist lediglich wann und wo die Dienststücke starten und enden. Alle für die Zulässigkeit relevanten Informationen wie Dienstlänge, Arbeitszeit, Pausendauer, Dienststart- und ende etc. können aus den Start- und Endknoten (mit ihren Merkmalen wie Zeit und Ort) der Dienststücke, die einen Dienst bilden, abgeleitet werden. Somit bestimmen nur diese Knoten, ob ein Dienst gültig ist oder nicht.

Die Dienststücke werden als kürzeste Wege zwischen kompatiblen Knotenpaaren im Dienststückerzeugungsgraphen konstruiert. Ist ein Dienst, der durch kompatible Knotenpaare seiner Dienststücke bestimmt ist, gültig in einer Iteration des Column-Generation-Verfahrens, dann ist er gültig in jeder Iteration, auch wenn die Verläufe seiner Dienststücke sich ändern. Somit ist es ausreichend nur einmal (am Anfang des Verfahrens) zu überprüfen, welche Kombinationen aus kompatiblen Knotenpaaren gültigen Diensten entsprechen. Nun werden in jeder Iteration des Column-Generation-Verfahrens nur reduzierte Kosten für gültige Kombinationen überprüft, wogegen die mehrfache zeitintensive Zulässigkeitsüberprüfung überflüssig wird.

Allerdings ist die Zahl solcher Kombinationen von kompatiblen Knotenpaaren, die gültige Dienste repräsentieren, so groß, dass es schon für relativ kleine Problemfälle technisch nicht mehr möglich ist, sie alle explizit zu merken. Wir entwickelten eine spezielle Datenstruktur, die eine große Menge solche Kombinationen implizit abbilden kann. Die Idee basiert auf der Tatsache, dass jeder Knoten in vielen kompatiblen Knotenpaaren als Start- oder Endknoten vorkommen kann. Für einen Knoten i im Dienststückerzeugungsgraphen existieren bis zu $|\widetilde{V}-1|$ kompatible Knotenpaare mit i als Endknoten, d.h. es existieren bis zu $|\widetilde{V}-1|$ Dienststücke, die den gleichen Endpunkt haben und sich nur in der Dauer unterscheiden. Bildet einer dieser Dienststücke den ersten Teil einer gültigen Kombination, dann bilden auch viele andere Dienststücke mit gleichem Endknoten mit hoher Wahrscheinlichkeit ebenfalls eine gültige Kombination mit dem gleichen End-Dienststück. Wir nutzen diese Beobachtung aus, um ein Bündel anstatt jeder einzelnen gültigen Kombination zu speichern.

Sei P die Menge aller gültigen Dienststücke, die als kürzeste Wege aus kompatiblen Knotenpaaren konstruiert wurden. Wir erstellen zwei Sortierungen S^s und S^e von P, einmal nach Starthaltestelle und Startzeit und einmal nach Endhaltestelle und Endzeit. Seien p_i und p_j zwei Dienststücke, die eine gültige Kombination (d.h. einen zulässigen Dienst) bilden, wobei $S^e(p_i)$ bzw. $S^s(p_j)$ ihre Positionen in der

Sortierung S^e bzw. S^s sind. Sowohl der rechte als auch der linke Nachbar von p_j in S^s bilden ebenfalls mit großer Wahrscheinlichkeit eine gültige Kombination mit p_i. Seien p_k und p_m solche Dienststücke, dass alle p_h mit $S^s(p_h) \in [S^s(p_k), S^s(p_m)]$ eine gültige Kombination mit p_i bilden. Wir nennen $[S^s(p_k), S^s(p_m)]$ das zulässige *Dienstend-Intervall* für p_i in S^s. Jedes Dienststück kann mehrere gültige Dienstend-Intervalle haben, die disjunkt sind. Analog dazu kann auch statt des ersten Dienststückes ein zulässiges *Dienststart-Intervall* $[S^e(p_v), S^e(p_w)]$ in S^e für alle Dienste aus $[S^s(p_k), S^s(p_m)]$ definiert werden, sodass alle p_i mit $S^e(p_i) \in [S^e(p_v), S^e(p_w)]$ und alle p_j mit $S^s(p_j) \in [S^s(p_k), S^s(p_m)]$ eine gültige Kombination darstellen. Wir nennen diese Datenstruktur eine *Dienstsequenz*. Sie impliziert ein Bündel von $(S^e(p_w) - S^e(p_v)) \times (S^s(p_m) - S^s(p_k))$ gültigen Diensten, wobei sie nur 4 ganzzahlige Werte (Randpositionen der Dienststart- und Dienstend-Itervalle) zu speichern braucht. Um den Speicherverbrauch weiter zu reduzieren, erweitern wir das Konzept der Dienstsequenz, indem alle Sequenzen mit gleichem Dienstend-Intervall in S^s zu einer einzigen erweiterten Dienstsequenz vereint werden. Somit referenziert eine erweiterte Dienstsequenz ein Dienstend-Intervall in S^s und mehrere dazu kompatible Dienststart-Intervalle in S^e. Die Abbildung 5.9 zeigt eine grafische Darstellung einer erweiterten Dienstsequenz mit zwei Dienststart-Intervallen und einem Dienstendintervall.

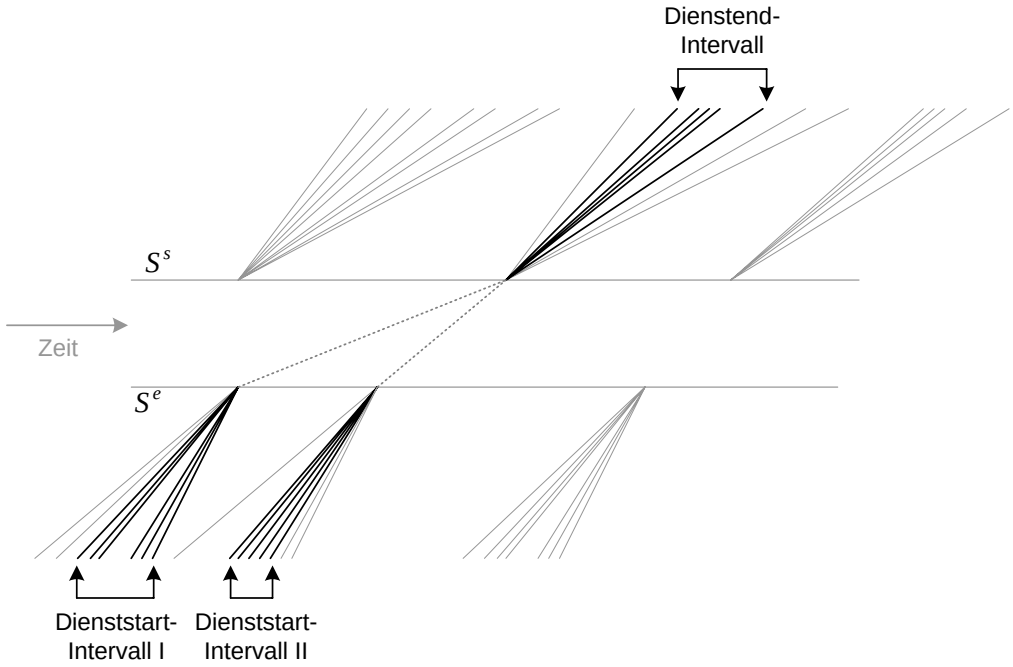

Abbildung 5.9: Beispiel einer Dienstsequenz mit zwei Dienststart-Intervallen

Tabelle 5.2 zeigt die Anzahl erweiterter Sequenzen und die Anzahl dadurch implizit abgebildeter gültiger Dienste für 3 unterschiedliche Probleminstanzen mit

100, 200 und 400 Fahrten. Außerdem sind in der Tabelle die maximale und die durchschnittliche Sequenzlänge (Anzahl implizierten Diensten pro Sequenz) sowie der für die Speicherung der Sequenzen erforderliche Speicherbedarf abgebildet.

Anzahl Fahrgastfahrten	100	200	400
Anzahl erweiterter Dienstsequenzen	6.884	34.662	143.312
Anzahl implizit abgebildeter Dienste	2.583.884	62.030.373	1.104.643.589
Maximale Sequenzlänge	18.990	213.180	2.044.840
Durchschnitt. Sequenzlänge	375	1.789	7.707
Speicherbedarf	1,2 MB	19 MB	220 MB

Tabelle 5.2: Dienstsequenzen: Kennzahlen für drei Probleminstanzen

Werden Dienstsequenzen am Anfang des Column-Generation-Verfahrens aufgebaut, kann das Pricing-Problem in jeder Iteration schnell gelöst werden, da die durch Sequenzen implizierten gültigen Dienste nur darauf überprüft werden müssen, ob sie negative reduzierte Kosten aufweisen. Die teure Zulässigkeitsüberprüfung wird somit nur einmal und nicht in jeder Iteration durchgeführt.

Bei mehreren Depots wird sowohl die Konstruktion der Dienstsequenzen als auch die Suche nach Diensten mit negativen reduzierten Kosten separat für jedes Depot durchgeführt. Das Gleiche gilt, wenn mehrere Dienstarten vorhanden sind und ein Dienstmix angestrebt ist bzw. Kapazitäten für Dienstarten vorhanden sind. Gibt es dagegen keine Einschränkung für Zusammensetzung der Dienstarten in der Lösung, dann entfällt die Notwendigkeit identische Dienste, die sich nur in Dienstart und ggf. Kosten unterscheiden, in das Master-Problem hinzufügen. In diesem Fall wird jede Kombination nur einmal konstruiert. Ist sie für mehrere Dienstarten zulässig, wird sie der Dienstart zugeordnet, wo sie die geringsten Kosten verursacht.

Wie Tabelle 5.2 zeigt, wächst der Speicherbedarf für die Speicherung von Sequenzen überproportional mit der Größe des Problems. Übersteigt er den zur Verfügung stehenden virtuellen Speicher, kann eine Strategie benutzt werden, die jeweils nur die notwendigen Sequenzen (z.B. für ein gegebenes Depot) in den Hauptspeicher lädt und alle anderen auf die Festplatte auslagert. Eine weitere Möglichkeit ist, nur für einen Teil des gesamten Problems die Sequenzen zu konstruieren und für den anderen Teil das Pricing-Problem auf konventionelle Weise zu lösen (d.h. durch Zulässigkeitsüberprüfung in jeder Iteration).

5.4.3 Erzeugung von Diensten durch RCSP

Sind mehr als zwei Dienststücke pro Dienst erlaubt, dann ist der enumerative Ansatz nicht praktikabel, da die Anzahl zulässiger Dienste exponentiell wächst, so-

dass sie sich nicht in akzeptabler Zeit vollständig aufzählen lassen. Daher wird in der Literatur oft das Problem zum Finden zulässiger Dienste mit negativen reduzierten Kosten als Netzwerkfluss-Problem mit Ressourcen bzw. *(ressourcen) beschränkte Kürzeste-Wege-Problem* (engl.: *(resource) constrained shortest path problem, RCSP*) formuliert[4].

In der Literatur werden in solchen Netzwerken zur Dienstgenerierung (*Dienstgenerierungsnetzwerke*) Dienstelemente bzw. Dienststücke durch Knoten repräsentiert. Außerdem stellen zwei speziellen Knoten, nämlich Quelle und Senke, den Dienstanfang und -ende dar. Kompatible Knoten werden durch Kanten verbunden, die je nach Definition Fahrtausführung, Fahrtunterbrechung, Fahrzeug-/Fahrerortwechsel oder Pause bedeuten. Die Kanten, die jeden Knoten mit Quelle bzw. Senke verbinden, repräsentieren Dienstvorbereitungs- bzw. Dienstabschlusszeiten. Dienste sind durch Pfade von der Quelle zur Senke abgebildet. Die Kosten auf den Kanten sind so definiert, dass die Kosten des Pfades den reduzierten Kosten des zugehörigen Dienstes entsprechen. Die für die Dienstregeln relevanten Größen wie z.B. Dienstlänge, Pausendauer, Lenkzeit etc. werden als Ressourcen modelliert. Für jede Kante und ggf. jeden Knoten wird ein Verbrauch an Ressourcen festgelegt. Der Ressourcenverbrauch eines Pfades ist durch die Gesamtverbrauche aller darin enthaltener Kanten und Knoten definiert. Ein Dienst ist gültig, falls der Ressourcenverbrauch des zugehörigen Pfades für jede relevante Ressource im zulässigen Bereich liegt.

Die Wahl, ob Dienstelemente (siehe z.B. [Haase et al., 2001] und [Borndörfer et al., 2003]) oder Dienststücke (siehe z.B. [Freling, 1997] und [Desrochers and Soumis, 1989]) als Knoten abgebildet werden, ist ein Kompromiss zwischen der Netzwerkgröße und der Anzahl relevanten Ressourcen. Die zweite Variante geht von einer Menge zulässiger Dienststücke aus, die im Vorfeld erzeugt wurden. Somit wird die Anzahl der relevanten Ressourcen um diejenige reduziert, die für die Gültigkeitsprüfung der Dienststücke verantwortlich sind. Dies kann die Suche nach gültigen Pfaden vereinfachen, da die Problemkomplexität unter anderem von der Anzahl der Ressourcen abhängt. Allerdings ist diese Alternative nur eingeschränkt einsetzbar, da die Diensterzeugungsnetzwerke wegen der großen Anzahl gültiger Dienste sehr groß werden können.[5]

In unserem RCSP-basiertem Ansatz zur Dienstgenerierung benutzen wir eine al-

[4] Anmerkung: In Rahmen der vorliegenden Arbeit beschränken wir uns auf Dienstarten mit maximal zwei Dienststücken pro Dienst. Dennoch möchten wir an dieser Stelle einen kurzen, zusammenfassenden Überblick über die entwickelten Verfahren geben, auch wenn RCSP-basierte Pricing nicht zu den Schwerpunkten dieser Arbeit gehört. Für eine detaillierte Beschreibung der vorgestellten Modellierung verweisen wir auf [Steinzen et al., 2006].

[5] z.B. in [Freling, 1997] werden Dienststücke durch Knoten im Diensterzeugungsnetzwerk repräsentiert. Alle kompatiblen Dienststücke, die einen Dienst bilden können, werden explizit mit Kanten verbunden. Ist N die Anzahl der Fahrgastfahrten, dann hat die Menge der Knoten/Dienststücke die Größenordnung $\mathcal{O}(N^2)$ und die Menge der Kanten $\mathcal{O}(N^4)$.

ternative Modellierung des Diensterzeugungsnetzwerks, das auf einem Time-Space-Netzwerk basiert. Zunächst wird eine Menge gültiger Dienststücke, so wie im Unterabschnitt 5.4.1 beschrieben, generiert. Beachte, dass es dabei ausreicht, maximal ein Dienststück pro Fahrgastfahrt-Paar zu betrachten. Basierend auf dieser Menge der Dienststücke wird das Diensterzeugungsnetzwerk aufgebaut. Allerdings repräsentieren Knoten nicht die Dienststücke selbst, sondern ihre Start- und Endzeiten und Orte. Die Kanten im Netzwerk repräsentieren Dienststücke (verbinden Startzeitknoten mit Endzeitknoten), Pausen (verbinden Endzeitknoten mit Startzeitknoten) oder Vorbreitungs- bzw. Abschlusszeiten eines Dienstes (verbinden Quelle mit Startknoten bzw. Endknoten mit Senke). Auf jeder Kante werden Ressourcenverbrauche für die relevante Ressourcen definiert. Ein ressourcenbeschränkter Pfad von Quelle zur Senke repräsentiert einen gültigen Dienst, wobei die Kosten auf den Kanten so gesetzt sind, dass die Kosten des Pfades den reduzierten Kosten des abzubildenden Dienstes entsprechen. Für die Suche nach gültigen Diensten mit negativen reduzierten Kosten wird ein Algorithmus der dynamischen Programmierung benutzt.

Da die Anzahl der möglichen Start- und Endzeit bzw. Orte für alle Dienststücke durch die Anzahl der Fahrgastfahrten begrenzt ist, besitzt das vorgeschlagene Diensterzeugungsnetzwerk (nur) $\mathcal{O}(N)$ Knoten und (nur) $\mathcal{O}(N^2)$ Kanten, wenn N die Anzahl der Fahrgastfahrten ist (die Kantenmenge der von Freling vorgeschlagene Modellierung besitzt die Größenordnung von $\mathcal{O}(N^4)$). Durch eine starke Netzwerkverkleinerung können mit diesem Ansatz Pricing-Probleme für komplizierte Dienstarten mit mehreren Dienststücken pro Dienst (getestet mit bis zu 10 Dienststücken pro Dienst) effizient gelöst werden (siehe [Steinzen et al., 2006]).

5.5 Spaltenmanagement

Spaltenmanagement ist ein wichtiger Bestandteil des Column-Generation-Ansatzes. Eine intelligente Auswahlstrategie zur Erweiterung des eingeschränkten Master-Problems sorgt dafür, dass der gesamte Column-Generation-Prozess schneller die angestrebte untere Schranke erreicht und somit schneller terminiert. Auf der anderen Seite soll die Größe des eingeschränkten Master-Problems nicht zu stark steigen, damit es in akzeptabler Zeit gelöst werden kann. Für die vorgestellten Verfahren zur Lösung von sowohl sequenziellen als auch integrierten Umlauf- und Dienstplanungsproblemen wurden unterschiedliche Auswahlstrategien zur Verwaltung des eingeschränkten Master-Problems untersucht.

5.5.1 Erweiterung des eingeschränkten Master-Problems

Um den Column-Generation-Prozess fortzusetzen (bzw. zu beenden) ist es ausreichend, einen neuen Dienst mit negativen reduzierten Kosten dem eingeschränkten Master-Problem hinzuzufügen (bzw. zu beweisen, dass es keinen solchen Dienste gibt). Diese Strategie, auch als *Single Pricing* bekannt, führt allerdings dazu, dass die Verbesserung der Zielfunktion zwischen den Iterationen oft nur minimal ist und das gesamte Verfahren sehr viel Zeit in Anspruch nimmt. Daher werden in der Praxis oft anstatt nur einem gleich mehrere Dienste mit negativen reduzierten Kosten in jedem Pricing-Schritt dem Master-Problem hinzugefügt. Diese Variante, auch als *Multiple Pricing* bekannt, wird in dem vorgeschlagenen Column-Generation-Ansatz benutzt.

Die Menge der Dienste mit negativen reduzierten Kosten wird im Pricing-Schritt bestimmt (siehe Abschnitt 5.4). Allerdings ist sie üblicherweise so groß (besonders während der ersten Pricing-Schritte), dass bei kompletter Aufnahme aller Dienste in das eingeschränkte Master-Problem es sehr schnell eine kritische Größe erreicht und dann nicht mehr bzw. sehr zeitaufwendig gelöst werden kann. Daher schlagen wir vor, die Anzahl der neuen Dienste pro Iteration zu begrenzen, damit die Größe des eingeschränkten Master-Problems besser kontrolliert werden kann. Bei mehreren Dienstarten erwies es sich außerdem als sinnvoll, auch die Anzahl neuer Dienste pro Dienstart in jeder Iteration zu begrenzen. Wie stark die Begrenzung ist, hängt von der Problemgröße, Problemstruktur und Lösungszeit für das Master-Problems ab und kann entsprechend individuell eingestellt werden.

Bei der Auswahl der Dienste ist man natürlich an solchen mit kleinsten reduzierten Kosten interessiert, da sie potenziell mehr zur Verbesserung der Zielfunktion bringen können. Im besten Fall würde man alle gültigen Dienste anhand ihrer reduzierten Kosten (aufsteigend) sortieren und dann K^{max} besten nehmen, wobei K^{max} die maximale Anzahl der neuen Dienste pro Column-Generation-Iteration ist. Allerdings hat dieser Vorgehensweise zwei Nachteile, zum einen müssen dafür alle gültigen Dienste explizit erzeugt werden und zum anderen kann der Aufwand für das Sortieren sehr groß sein (üblicherweise $\mathcal{O}(n \log n)$, wenn n die Anzahl aller neuen Dienste ist). Stattdessen benutzen wir eine alternative Strategie, die in unseren Experimenten sehr gute Ergebnisse gezeigt hat. In jeder Iteration i des Column-Generation-Ansatzes definierten wir einen Schwellenwert $R_i^0 < 0$ und suchen zunächst nur nach Diensten mit reduzierten Kosten im Intervall $(-\infty, R_i^0)$. Wird dabei die maximale Anzahl K^{max} erreicht, dann wird die Suche abgebrochen und der Schwellenwert für die nächste Iteration um θ_R herabgesetzt $R_{i+1}^0 = R_i^0 - \theta_R$ (somit ist die Suche in der nächsten Iteration etwas restriktiver). Tritt das Gegenteil ein, d.h. die Anzahl der neuen Dienste mit reduzierten Kosten im Intervall $(-\infty, R_i^0)$ ist kleiner als K^{max}, dann wird ein neuer Schwellenwert $R_i^1 = \min(0, R_i^0 + \theta_R)$

definiert. Danach wird die Suche wiederholt, aber diesmal werden Dienste mit reduzierten Kosten im Intervall $[R_i^0, R_i^1)$ gesucht. Das Ganze wird solange wiederholt, bis K^{max} neue Dienste gefunden wurden oder das Akzeptanzintervall leer ist (d.h. $R_i^{k-1} = R_i^k = 0$). In der nächsten Column-Generation-Iteration wird mit dem alten Schwellenwert angefangen, d.h. $R_{i+1}^0 = R_i^k$. Die Wahl von R_1^0 und θ_R ist ein Kompromiss zwischen der Qualität der neuen Spalten und Anzahl der Suchläufe und kann je nach Probleminstanz bzw. Anforderungen an die Lösung individuell gewählt werden.

Ein weiterer wichtiger Punkt neben der Begrenzung der Anzahl neuer Dienste ist ihre gute Verteilung sowohl örtlich als auch zeitlich. Bekommt beispielsweise eine Fahrgastfahrt sehr niedrige reduzierte Kosten, dann werden im Pricing-Schritt vorzugsweise solche Dienste gefunden, die diese Fahrt beinhalten, da sie dadurch insgesamt kleine reduzierte Kosten bekommen. Dies ist allerdings nicht erwünscht, da von all diesen Diensten höchstens einer in der Lösung ausgewählt wird. Um diese Situation zu vermeiden, wird die Suche nach neuen Diensten modifiziert. Zum einen fangen wir die Suche jedes Mal mit einer neuen, zufällig ausgewählten Starthaltestelle. Zum anderen wird auch die Suche innerhalb einer erweiterten Sequenz (siehe Unterabschnitt 5.4.2) randomisiert. Zusätzlich engen wir den Suchraum ein, indem die Anzahl neuer Dienste, die innerhalb einer erweiterten Sequenz sowohl pro Dienststück als auch insgesamt gefunden werden, begrenzt wird. Diese beiden Limits verfallen (d.h. der Suchraum wird komplett geöffnet), sobald pro Iteration weniger als K^{max} neuen Spalten gefunden werden konnten.

In unseren Experimenten haben die beiden Modifikationen der Suche (Randomisierung und schrittweise Eröffnung des Suchraumes) zu einer enormen Verringerung der Gesamtanzahl der Column-Generation-Iterationen (für einige Probleminstanzen bis zu 90 % weniger Iterationen) und damit zu einer deutlichen Beschleunigung des gesamten Verfahrens geführt.

5.5.2 Verkleinerung des eingeschränkten Master-Problems

Trotz der Begrenzung der maximalen Anzahl neuer Dienste pro Column-Generation-Iteration kann das eingeschränkte Master-Problem eine Größe erreichen, ab der es nicht mehr in akzeptabler Zeit zu lösen ist. Daher definieren wir eine maximale Größe K_{RMP}^{max} und eine Zielgröße K_{RMP}^{target} für das Master-Problem. Sobald die Anzahl der Spalten in dem eingeschränkten Master-Problem K_{RMP}^{max} übersteigt, werden $K_{RMP}^{max} - K_{RMP}^{target}$ Spalten mit den schlechtesten reduzierten Kosten aus dem Master-Problem entfernt. Die richtige Wahl von K_{RMP}^{max} und K_{RMP}^{target} ist ein Kompromiss zwischen der Anzahl der Iteration und Lösungszeit pro Iteration und hängt von der Größe, Struktur bzw. weiteren Eigenschaften der Probleminstanz sowie Anfor-

derungen an den Lösungsprozess ab.

Neben dem beschriebenen Verfahren zur Verkleinerung des eingeschränkten Master-Problems wurde eine weitere Strategie untersucht, in der in jeder Iteration von Column-Generation diejenige Dienste, die zuletzt hinzugefügt wurden, aber nach dem Lösen des eingeschränkten Master-Problems hohe reduzierte Kosten bekommen haben, wieder aus dem Problem gelöscht. Die Absicht war, dadurch die Größe des eingeschränkten Master-Problems noch stärker und gezielter schon am Anfang zu verkleinern. Allerdings erwies sich diese Strategie als weniger erfolgreich. Sie führte zu mehr Iterationen und demzufolge zu längerer Laufzeit als die oben beschrieben Variante mit K_{RMP}^{max} und K_{RMP}^{target}.

5.6 Ganzzahlige Lösung

Wie im Unterabschnitt 5.1.3 bereits skizziert, ist der gesamte Lösungsansatz für MD-VCSP zweistufig. Die LP-Phase (Abschnitte 5.3 - 5.5) dient im Prinzip dazu, eine möglichst gute untere Schranke für das Gesamtproblem zu finden und eine Menge von Spalten/Diensten zu generieren, die zum Finden einer guten zulässigen Lösung für MD-VCSP relevant sind. Können dabei keine neuen Dienste mit negativen reduzierten Kosten gefunden werden, d.h. die unter Schranke kann nicht mehr verbessert werden, oder ist ein anderes Abbruchkriterium erreicht, dann wird die LP-Phase beendet.

In der zweiten Stufe des Lösungsansatzes, der IP-Phase, wird eine möglichst gute zulässige Lösung gesucht. Zunächst wird das Lagrange-Dual-Problem der „teureren" Relaxation, in der nur die Kopplungsbedingungen (5.4) relaxiert sind, mit Hilfe des Subgradienten-Verfahrens (bzw. Volume-Algorithmus) gelöst. Diese Relaxation ist deswegen teuer, weil eines der resultierenden Lagrange-Unterprobleme ein $\mathcal{NP}$-hartes Mehrdepot-Umlaufplanungsproblem ist, was in jeder Iteration des Subgradienten-Verfahrens (bzw. Volume-Algorithmus) gelöst werden muss. Allerdings basiert das MDVSP auf dem gemeinsamen Time-Space-Netzwerk und kann dank seiner kompakten Formulierung schnell gelöst werden (siehe Unterabschnitt 5.2.1). Außerdem braucht das Subgradienten-Verfahren (bzw. Volume-Algorithmus) nicht so viele Iterationen, da es bereits mit guten Lagrange-Multiplikatoren aus der LP-Phase startet.

Ein optimaler Umlaufplan ist nur lokal optimal und führt bekanntlich nicht unbedingt dazu, dass die Gesamtlösung des Umlauf- und Dienstplanungsproblems optimal bzw. gut ist. Viel häufiger kann durch eine „Verschlechterung" des optimalen Umlaufplans ein viel besserer Dienstplan erreicht werden, sodass die Gesamtlösung besser ist. In jeder Iteration des Subgradienten-Verfahrens (bzw. Volume-

Algorithmus) wird die Kostenfunktion des MDVSP durch die implizite Betrachtung von Kopplungsbedingungen schrittweise so modifiziert, dass bei der nächsten Lösung derjenige Umlaufplan die besten Kosten bekommt, der für eine bessere gesamtoptimale Lösung sorgt. Anschließend wird für den resultierenden Umlaufplan das umlaufbasierte Dienstplanungsproblem (siehe Unterabschnitt 5.2.2) gelöst. Somit wird eine korrekte Kopplung der beiden Pläne sichergestellt, wobei die Optimalität der Lösung nicht garantiert werden kann.

Falls die Differenz zwischen der gefundenen Lösung und der unteren Schranke zu groß ist, verfolgen wir einen Vorschlag aus [Huisman, 2004] und berechnen einen Dienstplan nicht nur für den resultierenden Umlaufplan, sondern für n letzte Umlaufpläne. Damit wird die Wahrscheinlichkeit erhöht, eine gute gesamtoptimale Lösung zu finden. Außerdem untersuchten wir zwei weiteren Strategien, um die Lösungsqualität zu verbessern. Bei der ersten Strategie wird nach T^{IP}_{start} Anlauf-Iterationen des Subgradienten-Verfahrens (bzw. Volume-Algorithmus) in der IP-Phase alle T^{IP}_{freq} Iterationen ein passender Dienstplan für den aktuellen Umlaufplan gesucht. Dabei wird das umlaufbasierte Dienstplanungsproblem nur sehr grob, aber dafür schnell gelöst. Bei der zweiten Strategie berechnen wir eine gültige Lösung auch während der LP-Phase und zwar nur einmal pro Column-Generation-Iteration, am Ende des Subgradienten-Verfahrens (bzw. Volume-Algorithmus). Allerdings muss dafür erst das Mehrdepot-Umlaufplanungsproblem für die aktuelle Kostenfunktion gelöst werden. Ein häufigeres Berechnen einer zulässigen Lösung kostet zwar Zeit, kann sich aber positiv auf die gesamte Laufzeit auswirken, da die obere Schranke bei dem Subgradienten-Verfahren (bzw. Volume-Algorithmus) ständig verbessert wird, und es somit schneller terminieren kann.

Während der IP-Phase werden keine neuen Dienste für das MD-VCSP nachgeneriert. Das Lagrange-Dual-Problem wird mit den Diensten gelöst, die in der LP-Phase gefunden wurden. Als weitere Maßnahme zur Verbesserung der Lösungsqualität, kann allerdings bei der Lösung von umlaufbasierten Dienstplanungsproblemen in der IP-Phase auf diese Einschränkung verzichtet werden.

Anstatt einer sequenziellen Lösung der Umlauf- und Dienstplanungsprobleme während der IP-Phase kann ein adaptiver teilintegrierter Ansatz verwendet werden, der in Kapitel 6 vorgestellt wird. Der alternative Ansatz ist zwar etwas zeitintensiver, führt aber fast immer zu einer besseren Lösung.

Zur Zeit der Erstellung der vorliegenden Arbeit wurde die IP-Phase durch eine sukzessive Reduktion der Problemgröße erweitert. Unter vielen untersuchten Reduktionstechniken, hat sich ein Verfahren als besonders erfolgreich durchgesetzt. Dabei wird die in [Holmberg and Yuan, 2000] präsentierte Idee der α-*Fixierung* verfolgt. Dieses Fixierungsschema wird auf die Zuordnung von Fahrgastfahrten zu Depots angewandt: Wird eine Fahrgastfahrt im Laufe des Subgradienten-Verfahrens

der IP-Phase sehr häufig einem bestimmten Depot zugeordnet, dann wird sie zu diesem Depot fest fixiert. Dadurch wird das Mehrdepot-Umlauf- und Dienstplanungsproblem nach und nach in mehrere Eindepot-Umlauf- und Dienstplanungsprobleme aufgeteilt, die viel einfacher gelöst werden können. Diese auf α-Fixierung basierte Technik hat sehr gute Ergebnisse sowohl hinsichtlich der Lösungszeit als auch hinsichtlich der Lösungsqualität gezeigt[6].

5.7 Allgemeiner Fall: beliebige Ablösemöglichkeit

Im Kapitel 5 vorgestellte Modelle und Lösungsansätze gelten unter den am Anfang des Kapitels getroffenen Annahmen (siehe Seite 70). Eine dieser Annahmen unterstellt, dass als Ablösepunkte Start und Ende jeder Fahrgastfahrt gelten. Außerdem gilt das eigene Depot zu jedem Zeitpunkt als Ablösepunkt. Diese Einschränkung wurde zunächst zwecks besserer Vergleichbarkeit mit existierenden Modellen (siehe [Huisman, 2004]) und Einfachheit der Darstellung eingeführt. Der allgemeine Fall aus der Praxis sieht allerdings auch beliebig platzierte Ablösepunkte vor. So können sich die Fahrer beispielsweise auch während einer Fahrgastfahrt an zuvor definierten Stellen bzw. zu zuvor definierten Zeitpunkten ablösen. Auf der anderen Seite können auch bestimmte Start- und Endpunkte der Fahrgastfahrten als Ablösemöglichkeit nicht zugelassen sein. Ein Extremfall dafür, der im Regionalverkehr oft vorkommt, ist die Benutzung des eigenen Depots als einzige Ablösemöglichkeit.

In diesem Abschnitt wird diskutiert, wir die vorgestellte Modellierung und die präsentierten Lösungsansätze aus den vorangegangenen Abschnitten für den allgemeinen Fall mit beliebigen Ablösemöglichkeiten erweitert werden kann.

Im Prinzip betrifft die allgemeine Erweiterung nur den Dienstplanungsteil des integrierten Umlauf- und Dienstplanungsproblems. Somit kann die im Unterabschnitt 5.1.1 eingeführte Modellierung beibehalten werden. Gleichzeitig definieren wir für jede Fahrgastfahrt-Kante im Planungsnetzwerk die Menge ihrer Dienstelement-Abschnitte. Ein *Dienstelement-Abschnitt* ist der Teil einer Fahrgastfahrt zwischen zwei aufeinanderfolgenden Punkten aus der Menge: Start der Fahrgastfahrt, Ablösepunkte innerhalb der Fahrgastfahrt und Ende der Fahrgastfahrt. Zur Erinnerung: Unter einem Dienstelement verstehen wir eine nicht mehr zu unterteilende Arbeitsaufgabe, die aus einer Folge von Fahrgast- und Leerfahrten zwischen zwei

[6]Die beschriebene Fixierungsschema war nicht als Bestandteil der vorliegenden Arbeit geplant. Zum Zeitpunkt der Erstellung der vorliegenden Arbeit war eine weitgehende Validierung noch nicht abgeschlossen. Die an dieser Stelle vorgestellte Zusammenfassung der gewonnenen Erkenntnisse dient der Vollständigkeit des behandelten Themas. Für eine detaillierte Beschreibung der untersuchten Reduktionstechniken und numerischen Ergebnissen wird auf zukünftige Veröffentlichungen des DS&OR-Lab der Universität Paderborn verwiesen.

aufeinander folgenden Ablösepunkten in einem Umlauf besteht. Somit kann jedes Dienstelement aus mehreren Dienstelement-Abschnitten bestehen. Darf am Anfang und am Ende jeder Fahrgastfahrt abgelöst werden (unabhängig davon, ob Ablösepunkte zwischendrin erlaubt sind), dann entspricht jedes Dienstelement einem Dienstelement-Abschnitt. Die Abbildung 5.10 veranschaulicht den Zusammenhang beider Begriffe.

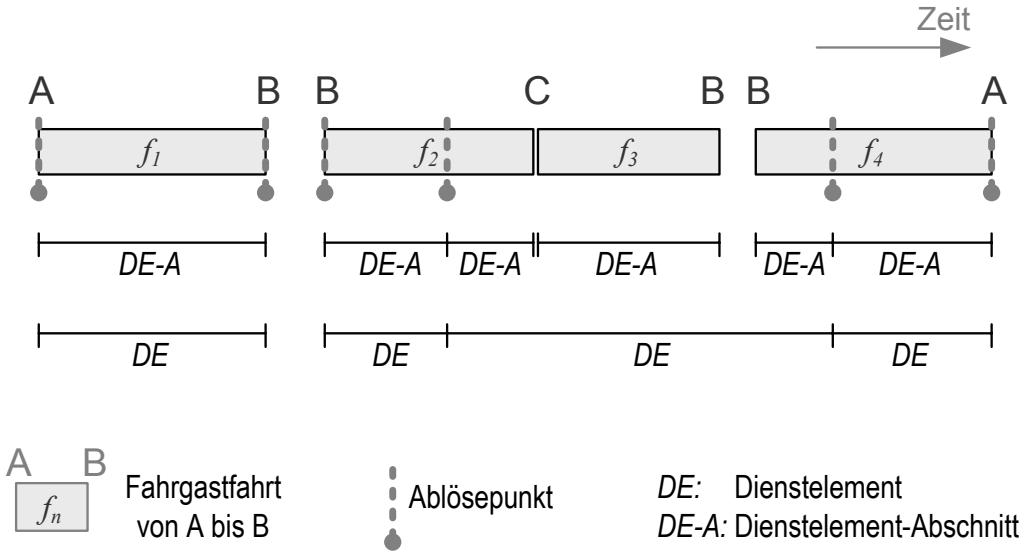

Abbildung 5.10: Dienstelemente und Dienstelement-Abschnitte

Im allgemeinen Fall besteht ein Dienst nicht mehr zwangsläufig aus kompletten Fahrgastfahrten, sondern kann auch nur Teile davon (Dienstelement-Abschnitte) beinhalten. Allerdings kann diese Situation nur am Anfang oder am Ende seiner Dienststücke auftreten. Sei Γ_{ij}^d die Anzahl der Dienstelement-Abschnitte der Fahrgastfahrt, die mit der Kante $(i,j) \in \bar{A}^d$ assoziiert ist. Die Menge aller Dienste, die den Dienstelement-Abschnitt $p \in \{1,\ldots,\Gamma_{ij}^d\}$ der Kante $(i,j) \in \bar{A}^d$ beinhalten, sei durch $K^d(i,j)_p$ definiert. In der mathematischen Formulierung (5.1)-(5.6) auf Seite 78 für das integrierte Umlauf- und Dienstplanungsproblem müssen lediglich die Kopplungsbedingungen (5.4) wie folgt angepasst werden:

$$\sum_{k \in K^d(i,j)_p} x_k^d - y_{ij}^d = 0 \quad \forall\, d \in D, \forall (i,j) \in \bar{A}^d, \forall p \in \{1,\ldots,\Gamma_{ij}^d\} \tag{5.39}$$

Somit existiert nicht mehr eine Kopplungsbedingung pro Kante, sondern eine Kopplungsbedingung pro Dienstelement-Abschnitt.Durch die Änderung der Kopplungsbedingungen müssen auch die Lagrange-Relaxationen des eingeschränkten Master-Problems angepasst werden. Allerdings sind diese Anpassungen trivial und werden hier nicht näher diskutiert.

Die Generierung neuer Dienste im Pricing-Problem des Column-Generation-Verfahrens bedarf ebenfalls einiger Änderungen. Allerdings betrifft das nur Phase, in der zunächst eine Menge von Dienststücken gebildet wird (siehe Unterabschnitt 5.4.1). Dazu wird der Dienststückerzeugungsgraph wie folgt modifiziert: Besteht eine Kante aus mehreren Dienstelement-Abschnitten, so wird sie an den Ablösepunkten in mehrere Kanten geteilt. Weiterhin unterscheiden wir zwischen Knoten, die gleichzeitig einen Ablösepunkt darstellen und Knoten an denen kein Ablösen erlaubt ist. Die Abbildung 5.11 zeigt einen Abschnitt des modifizierten Dienststückerzeugungsgraphen für das Beispiel aus dem Bild 5.10.

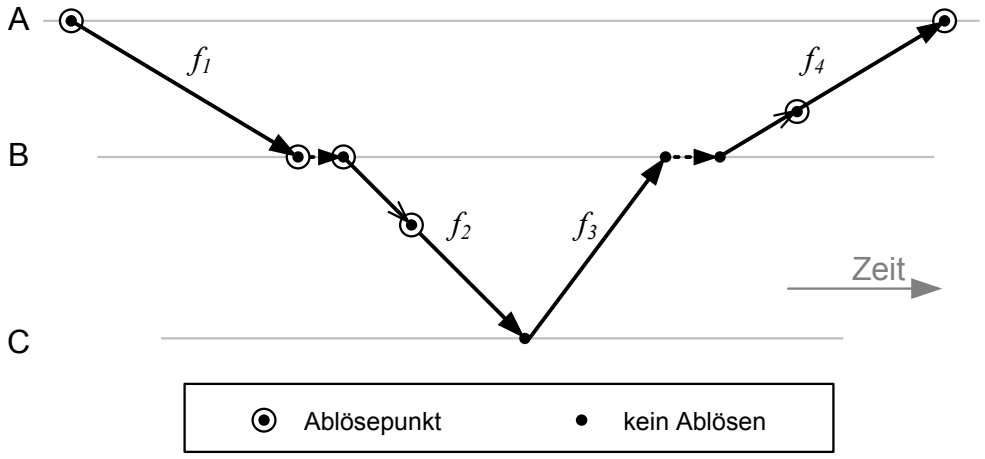

Abbildung 5.11: Anpassung des Dienststückerzeugungsgraphen

Die Suche nach den Dienststücken wird nun so angepasst, dass die kürzesten Wege nur zwischen zwei Ablösepunkt-Knoten gesucht bzw. akzeptiert werden (vgl. Unterabschnitt 5.4.1).

5.8 Numerische Ergebnisse

In diesem Abschnitt werden die durchgeführten Tests zu dem in diesem Kapitel vorgeschlagenen integrierten Lösungsansatz für Umlauf- und Dienstplanungsprobleme diskutiert. Zum einen werden hier die unterschiedlichen Phasen des gesamten Lösungsverfahrens anhand einer Reihe von Testläufen validiert; zum anderen die vorgeschlagenen Alternativen bei der Modellierung und Lösung der einzelnen Schritte gegenüber gestellt. Das Ziel ist eine Auswahl von Methoden bzw. Einstellungen zu bestimmen, die für die benutzten Probleminstanzen einen möglichst guten Kompromiss zwischen Lösungszeit und Lösungsqualität ermöglichen.

Die Menge der Testfälle besteht aus drei Klassen mit je 5 Testinstanzen, die

sich aus den künstlich generierten ECOPT-Instanzen von Dennis Huisman zusammensetzen (siehe A.2). Dabei wurden Probleminstanzen mit vier Depots ausgewählt, die je nach Klasse 80, 160 bzw. 320 Fahrgastfahrten beinhalten. Wir lassen fünf Dienstarten zu: Teildienst, Frühdienst, Tagesdienst, Spätdienst und geteilter Dienst. Die genaue Spezifikation und Eigenschaften dieser Dienstarten sind im Anhang A.1 dieser Arbeit zu finden. Es gelten alle am Anfang des Kapitels getroffenen Annahmen (siehe Seite 70). Die Kostenfunktion ist wie folgt definiert: fixe Kosten von 1000 Kosteneinheiten für jeden Dienst und jeden Umlauf und variable Kosten von 1 Kosteneinheit für jede Minute, die ein Fahrzeug außerhalb des eigenen Depots verbringt.

Der VCSP-Solver wurde in der Programmiersprache C# implementiert und mit .NET Framework der Version 2.0 unter Windows XP kompiliert. Alle in diesem Abschnitt dargestellten Testergebnisse wurden auf einem Dell OptiPlex GX620 Personalcomputer mit einem Pentium IV 3,4 GHz Prozessor und 2 GB RAM ausgeführt.

5.8.1 Master Problem

Zunächst wird das eingeschränkte Master-Problem untersucht. Wie bereits beschrieben relaxieren wir die Kopplungsbedingungen (5.4) der Originalformulierung (5.1)-(5.6) des MD-VCSP mit Hilfe der Lagrange-Relaxation, sodass sie in zwei Teile zerfällt: Ein triviales Auswahlproblem (x-Teil) und ein immer noch $\mathcal{NP}$-hartes Mehrdepot-Umlaufplanungsproblem (y-Teil). Als weitere Vereinfachung wurde in Unterabschnitt 5.3.1 vorgeschlagen, das schwer zu lösende MDVSP-Unterproblem weiter zu relaxieren. Dabei wurden zwei Lagrange-Relaxationen diskutiert, die je nach relaxierten Nebenbedingungen den y-Teil zu mehreren kleinen SDVSP (Relaxation I) oder einem großen SDVSP (Relaxation II) reduzieren. Die somit resultierenden Eindepot-Umlaufplanungsprobleme können in polynomieller Zeit gelöst werden. Zur Lösung des Lagrange-Dual-Problems schlugen wir zwei Lösungsverfahren vor, nämlich das weit verbreitete Subgradienten-Verfahren (siehe Unterabschnitt 5.3.3) und eine relative neue Methode, den Volume-Algorithmus (siehe Unterabschnitt 5.3.4).

Aus der Kombination zweier Relaxationen und zweier Lösungsalgorithmen ergeben sich folgende vier Varianten für das eingeschränkte Master-Problem:

LR1+S: Relaxation I und Subgradienten-Verfahren

LR1+V: Relaxation I und Volume-Algorithmus

LR2+S: Relaxation II und Subgradienten-Verfahren

LR2+V: Relaxation II und Volume-Algorithmus

Für alle durchgeführten Tests gilt:

- Column-Generation terminiert, wenn der Zielfunktionswert sich in den letzten 10 Iterationen weniger als um 2% verbessert hat.
- Das Pricing-Problem wird für jedes Depot separat gelöst. Die maximale Anzahl neuer Spalten, die dem eingeschränkten Master-Problem in jeder Iteration hinzugefügt werden, ist auf 20000 pro Depot begrenzt.
- Für die Suche nach neuen Diensten wird die 2-Phasen-Prozedur mit Aufzählung der Dienste über die erweiterten Dienstsequenzen eingesetzt (siehe Unterabschnitt 5.4.2).
- Sobald die Größe des Master-Problems $K_{\mathrm{RMP}}^{\max}$ Spalten erreicht hat, wird sie auf $K_{\mathrm{RMP}}^{\mathrm{target}}$ Spalten reduzierten. Für die unterschiedliche Problemklassen gelten folgende Werte:
  bei 80 Fahrten: $K_{\mathrm{RMP}}^{\max} = 150.000$, $K_{\mathrm{RMP}}^{\mathrm{target}} = 70.000$
  bei 160 Fahrten: $K_{\mathrm{RMP}}^{\max} = 200.000$, $K_{\mathrm{RMP}}^{\mathrm{target}} = 100.000$
  bei 320 Fahrten: $K_{\mathrm{RMP}}^{\max} = 250.000$, $K_{\mathrm{RMP}}^{\mathrm{target}} = 120.000$
- Die verwendete Variante des Subgradienten-Verfahrens beinhaltet alle im Unterabschnitt 5.3.3 vorgeschlagenen Erweiterungen und Verbesserungen.
- Alle Umlaufplanungsprobleme werden mit Hilfe der Netzwerk-Simplex-Implementierung aus der Optimierungsbibliothek CPLEX (Version 9.1.3) gelöst. Dabei werden alle Einstellungen des Solvers bei ihren Standardwerten belassen.

Die maximale Anzahl von Column-Generation-Iterationen ($T_{\mathrm{CG}}^{\max}$) sei zunächst auf 80 und die maximale Anzahl der Iteration im Subgradienten-Verfahren bzw. Volume-Algorithmus ($T_{\mathrm{LR}}^{\max}$) auf 1000 begrenzt. In Tabelle 5.3 sind die Ergebnisse der Vergleichstests zusammengefasst. Alle Werte sind Durchschnitte über 5 Instanzen je Gruppe. In der ersten Spalten ist die beste erreichte untere Schranke angegeben. Die nächsten drei Spalten zeigen Laufzeit (in Minuten), die für das Lösen des Master-Problems (t_{RMP}) bzw. Pricing-Problem (t_{pricing}) benötigt wurde, sowie deren Summe (t_{gesamt}). Dabei handelt es sich bei t_{pricing} sowohl um die Zeit für das eigentliche Pricing-Problem (Suche nach neuen Spalten) als auch um die Zeit für das Initialisieren der erweiterten Dienstsequenzen (siehe Unterabschnitt 5.4.2). Die letzten beiden Spalten zeigen die Anzahl der Column-Generation-Iterationen (Anzahl gelöster Lagrange-Dual-Probleme) sowie die kumulierte Anzahl aller Iterationen im Subgradienten-Verfahren bzw. Volume-Algorithmus (Anzahl gelöster Lagrange-Unterprobleme).

	LP	Laufzeit			T_{CG}	T_{LR}
		t_{RMP}	t_{pricing}	t_{gesamt}		
80 Fahrten						
LR1+S	27.366	4,10	0,13	4,23	16,2	7616
LR1+V	27.577	6,75	0,13	6,88	18,4	10286
LR2+S	27.402	4,73	0,13	4,86	15,4	7665
LR2+V	27.597	6,38	0,13	6,51	16,8	9347
160 Fahrten						
LR1+S	42.793	18,8	2,8	21,6	28,6	15659
LR1+V	43.196	25,4	2,5	27,9	29,8	18096
LR2+S	42.888	21,1	2,3	23,4	21,6	13674
LR2+V	43.213	25,0	2,5	27,5	26,4	15184
320 Fahrten						
LR1+S	71.953	70,0	35,4	105,4	51,2	29948
LR1+V	72.793	138,5	27,9	166,4	52,4	43031
LR2+S	72.039	172,8	41,4	214,2	48,8	33809
LR2+V	72.333	145,3	36,8	182,1	49,4	28953

Tabelle 5.3: Relaxation I vs. Relaxation II und Subgradienten-Verfahren vs. Volume-Algorithmus

Wie aus der Tabelle 5.3 zu sehen ist, schneidet die Variante **LR1+S** (Lagrange-Relaxation I in Verbindung mit Subgradienten-Verfahren) sowohl hinsichtlich der Lösungsqualität als auch hinsichtlich der Laufzeit am besten ab. Bei der Gegenüberstellung beider Relaxationen wird ersichtlich, dass die zweite Relaxation (ein großes SDVSP) eine gleichwertige oder schlechtere LP-Lösung als die Erste (viele kleine SDVSP) liefert, dagegen aber deutlich mehr Zeit für das Lösen der Unterprobleme beansprucht. Dies gilt insbesondere bei großen Problemfällen. Der Volume-Algorithmus konnte sich gegenüber dem bewähren Subgradienten-Verfahren weder in puncto Lösungsqualität noch in der Laufzeit durchsetzen. Allerdings sollte man dabei nicht außer Acht lassen, dass die eingesetzte Variante des Subgradienten-Verfahrens durch zahlreiche Erweiterungen stark verbessert wurde und mit einem klassischen Ansatz nicht vergleichbar ist.

Vergleicht man die Zeit, die für das Lösen des Master-Problems und des Pricing-Problems benötigt wurde, dann stellt man fest, dass der Anteil der Zeit für das Pricing-Problem in der Gesamtlaufzeit eher eine untergeordnete Rolle spielt. Zur Erinnerung: in nahezu allen Berichten über das integrierte Lösen von Umlauf- und Dienstplanungsprobleme war diese Phase des Column-Generation-Ansatzes die teuerste hinsichtlich der Laufzeit (siehe Abschnitt 4.3). Dies bekräftigt die vorgeschlagene Vorgehensweise bei der Generierung neuer Dienste (siehe Unterabschnitt

5.4.2).

Da die Variante LR1+S alle anderen Kombinationen klar überlegen ist, betrachten wir sie im weiteren Verlauf als einzige Alternative für das Lösen des Master-Problems.

Um die Lösungszeit für das Master-Problem zu reduzieren, untersuchen wir zwei weitere Maßnahmen. Erstens wird die Anzahl der gelösten Lagrange-Unterprobleme pro Column-Generation-Iteration stärker begrenzt, d.h. die maximale Anzahl an Iterationen im Subgradienten-Verfahren ($T_{\mathrm{LR}}^{\max}$) wird reduziert. Zweitens setzen wir das Limit für die Anzahl der Column-Generation-Iterationen ($T_{\mathrm{CG}}^{\max}$) herab. Die Tabelle 5.4 veranschaulicht, wie diese Maßnahmen sich auf die Qualität der Lösung sowie die Laufzeit auswirken.

	LP	Laufzeit			K_{CG}	K_{LR}
		t_{RMP}	t_{pricing}	t_{gesamt}		
160 Fahrten						
$T_{CG}^{max} = 80,\ T_{LR}^{max} = 1000$	42.793	18,8	2,8	21,6	28,6	15659
$T_{CG}^{max} = 30,\ T_{LR}^{max} = 1000$	42.808	15,4	2,5	17,9	23,2	12804
$T_{CG}^{max} = 80,\ T_{LR}^{max} = 600$	42.815	16,3	2,8	19,1	28,8	13437
$T_{CG}^{max} = 30,\ T_{LR}^{max} = 600$	42.823	15,9	2,6	18,5	26,0	12549
$T_{CG}^{max} = 80,\ T_{LR}^{max} = 400$	42.818	13,5	2,9	16,4	30,2	11283
$T_{CG}^{max} = 30,\ T_{LR}^{max} = 400$	42.824	11,3	2,5	13,8	24,4	9346
320 Fahrten						
$T_{CG}^{max} = 80,\ T_{LR}^{max} = 1000$	71.953	70,0	35,4	105,4	51,2	15659
$T_{CG}^{max} = 50,\ T_{LR}^{max} = 1000$	72.067	65,4	33,6	99,0	47,4	12804
$T_{CG}^{max} = 30,\ T_{LR}^{max} = 1000$	72.124	41,5	28,7	70,2	30,0	12549
$T_{CG}^{max} = 80,\ T_{LR}^{max} = 600$	71.974	55,9	35,0	90,9	50,6	13437
$T_{CG}^{max} = 50,\ T_{LR}^{max} = 600$	72.089	45,6	32,8	78,4	46,2	11283
$T_{CG}^{max} = 30,\ T_{LR}^{max} = 600$	72.108	37,5	29,8	67,3	30,0	9346
$T_{CG}^{max} = 80,\ T_{LR}^{max} = 400$	72.057	49,7	37,9	87,6	54,2	11283
$T_{CG}^{max} = 50,\ T_{LR}^{max} = 400$	72.102	45,9	35,3	81,2	50,0	9346
$T_{CG}^{max} = 30,\ T_{LR}^{max} = 400$	72.144	27,5	29,2	56,7	30,0	9346

Tabelle 5.4: Unterschiedliche Variationen von $T_{\mathrm{CG}}^{\max}$ und $T_{\mathrm{LR}}^{\max}$

Alle Werte sind Durchschnittswerte über 5 Instanzen je Gruppe. Um die endgültige LP-Werte vergleichen zu können, wird am Ende des Column-Generation-Verfahrens eine zusätzliche Iteration ausgeführt, in der das maximale Iterationslimit für das Subgradienten-Verfahren wieder auf $T_{\mathrm{LR}}^{\max} = 1000$ angehoben wird[7].

[7]Der LP-Wert (die untere Schranke für die ganzzahlige Lösung) zwischen den Column-Generation-Iterationen verläuft monoton fallend. Im Gegensatz dazu wird innerhalb einer

Die Ergebnisse zeigen, dass der Zielfunktionswert sich nach wenigen Column-Generation-Iterationen stabilisiert und nur sehr langsam (wenn überhaupt) verbessert. Vergleicht man die LP-Werte bei 80, 50 und 30 Iterationen (mit einem festen $T_{\text{LR}}^{\max}$) miteinander, stellt man fest, dass der Unterschied nur gering ist. Somit kann die Gesamtlaufzeit durch einen frühzeitigen Abbruch des Column-Generation-Verfahrens ohne signifikante Qualitätsverluste spürbar reduziert werden. Dieses Phänomen, auch bekannt als *tailing-off*, tritt u.a. aufgrund der starken Degeneration des Problems auf. Das Verfahren macht am Anfang große Fortschritte, braucht aber sehr lange um die finale Dualitätslücke zu schließen. Eine ähnlich Sättigungs- bzw. Stagnationsphase ist auch beim Subgradienten-Verfahren zu beobachten (siehe unterschiedliche Läufe mit einem festen $T_{\text{CG}}^{\max}$). Deswegen kann auch hier eine vergleichbare Strategie mit schärferen Abbruchbedingungen angewendet werden.

Wie aus der Tabelle zu sehen ist, kann durch die Einschränkung der beiden Iterationslimits die Gesamtlaufzeit bis auf die Hälfte reduziert werden. Eine geeignete Wahl von $T_{\text{CG}}^{\max}$ und $T_{\text{LR}}^{\max}$ stellt eine der Möglichkeiten dar, den gesamten Lösungsansatz für unterschiedliche Anforderungen bzgl. der Lösungszeit bzw. -qualität flexibel zu gestalten.

Für die weiteren Testläufe in dieser Arbeit verwenden wir die schnellere Variante mit $T_{\text{CG}}^{\max} = 30$ und $T_{\text{LR}}^{\max} = 400$.

5.8.2 Ganzzahlige Lösung

In der IP-Phase des gesamten Lösungsansatzes wird eine möglichst gute zulässige Lösung gesucht (siehe Abbildung 5.5). Dazu werden in der Originalformulierung (5.1)-(5.6) nur die Kopplungsbedingungen (5.4) mit Hilfe der Lagrange-Relaxation relaxiert. Das MD-VCSP zerfällt dabei in ein triviales Auswahlproblem für den x-Teil und ein Mehrdepot-Umlaufplanungsproblem für den y-Teil. Die Kopplung zwischen Umläufen und Diensten erfolgt implizit über die Lagrange-Multiplikatoren der relaxierten Kopplungsbedingungen. Das Lagrange-Dual-Problem lösen wir mit Hilfe des Subgradienten-Algorithmus, wobei in jeder Iteration ein $\mathcal{NP}$-hartes MDVSP gelöst werden muss. Allerdings basiert das MDVSP auf dem gemeinsamen Time-Space-Netzwerk und kann dank seiner kompakten Formulierung schnell gelöst werden. Außerdem starten wir mit bereits guten Lagrange-Multiplikatoren aus der LP-Phase, sodass der Subgradienten-Algorithmus schneller terminiert.

Column-Generation-Iteration das Lagrange-Dual-Problem gelöst, wobei der duale Zielfunktionswert an die primale untere Schranke ansteigt. Wird das Subgradienten-Verfahren frühzeitig abgebrochen, kann es passieren, dass der duale Wert noch viel tiefer von der eigentlichen unteren Schranke für die ganzzahlige Lösung liegt. Um die endgültige LP-Werte vergleichen zu können, wird am Ende der Column-Generation eine zusätzliche Iteration ausgeführt, in der die maximale Anzahl der Iterationen für Subgradienten-Verfahren wieder auf $T_{\text{LR}}^{\max} = 1000$ angehoben wird.

Der Unterschied zu der LP-Phase besteht darin, dass in jeder Iteration des Subgradienten-Algorithmus durch das Lösen des MDVSP ein zulässiger und für die aktuelle Kostenfunktion optimaler Umlaufplan berechnet wird. Allerdings ist ein optimaler Umlaufplan nur lokal optimal und führt bekanntlich nicht unbedingt dazu, dass die Gesamtlösung des Umlauf- und Dienstplanungsproblems optimal bzw. gut ist. Deswegen wird in jeder Iteration des Subgradienten-Verfahrens die Kostenfunktion für MDVSP durch die Lagrange-Multiplikatoren (d.h. durch die implizite Betrachtung der Kopplungsbedingungen) so schrittweise modifiziert, dass der resultierende Umlaufplan zu einer besseren Gesamtlösung führt. Anschließend wird für diesen Umlaufplan das klassische umlaufbasierte Dienstplanungsproblem gelöst und ein passender Dienstplan gebaut. Somit wird eine korrekte Kopplung der beiden Pläne sichergestellt, wobei die Optimalität der Gesamtlösung nicht garantiert werden kann.

Um die Wahrscheinlichkeit für das Finden einer guten Gesamtlösung zu erhöhen, lösen wir das umlaufbasierte Dienstplanungsproblem nicht nur für den letzten Umlaufplan, sondern auch zwischendurch. Dazu werden zwei Strategien untersucht:

$\mathbf{UB^{IP}}(T_{\mathbf{freq}})$: In der IP-Phase wird alle T_{freq} Iterationen des Subgradienten-Algorithmus für den aktuellen Umlaufplan ein passender Dienstplan bestimmt. Dabei wird das umlaufbasierte Dienstplanungsproblem grob, aber dafür schnell gelöst.

$\mathbf{UB^{LP+IP}}(T_{\mathbf{freq}})$: In IP-Phase wie $\text{UB}^{\text{IP}}(T_{\text{freq}})$. Zusätzlich wird in der LP-Phase nach jeder Column-Generation-Iteration eine zulässige Lösung berechnet. Dazu muss zunächst ein gültiger Umlaufplan durch das Lösen des MDVSP für die aktuelle Kostenfunktion (d.h. mit den optimalen Lagrange-Multiplikatoren) bestimmt werden. Anschließt wird das umlaufbasierte Dienstplanungsproblem gelöst.

In Tabelle 5.5 sind sowohl die beiden Strategien als auch die unterschiedliche Wahl von $T_{\text{freq}} = \{50, 10, 5, 2\}$ dargestellt. Bei den durchgeführten Tests gilt:

- Alle Annahmen und Einstellungen aus dem vorgehenden Unterabschnitt 5.8.1.
- Alle MDVSP sind mit dem *Barrier-Algorithmus* aus der Optimierungsbibliothek CPLEX (Version 9.1.3 mit Standardeinstellungen) gelöst.
- Zum Lösen des Master-Problems wird die Kombination **LR1+S** (Relaxation I und das Subgradienten-Verfahren) eingesetzt.
- Die maximale Anzahl von Iteration ist in Column-Generation auf 30 und im Subgradienten-Verfahren auf 400 begrenzt.

- Das umlaufbasierte Dienstplanungsproblem wird mit Hilfe des Column-Generation-Ansatzes in Verbindung mit Lagrange-Relaxation und Subgradienten-Verfahren gelöst. Dabei ist der Freiheitsgrad für die Konstruktion möglicher Dienste nur durch den zugrundeliegenden Umlaufplan und die üblichen Dienstregel und nicht durch die für MD-VCSP generierte Dienstmenge begrenzt (siehe unten). Für das Finden einer ganzzahligen Lösung wird der im Unterabschnitt 5.2.2 beschriebene hybride Ansatz (Branch-and-Bound im Verbindung mit der primalen Suchheuristik auf Basis der Simulated-Annealing) eingesetzt.

	t_{LP}		t_{IP}		t_{gesamt}	LP	IP	Anzahl	
UB-Strategie	ges.	UB	ges.	UB				Fzg.	Dienste
160 Fahrten									
$\text{UB}^{\text{IP}}(50)$	18,3	-	18,0	0,8	36,3	42915	48289	13,4	31,2
$\text{UB}^{\text{IP}}(10)$	18,3	-	21,7	3,5	40,0	42587	47640	13,4	30,6
$\text{UB}^{\text{IP}}(5)$	18,3	-	26,4	7,2	44,9	42742	46856	13,4	30,0
$\text{UB}^{\text{IP}}(2)$	18,3	-	33,3	15,7	51,6	42535	46623	13,4	29,8
$\text{UB}^{\text{LP+IP}}(50)$	19,2	3,5	12,4	0,8	31,6	43185	46742	13,4	30,0
$\text{UB}^{\text{LP+IP}}(10)$	19,2	3,5	15,3	3,4	34,4	43396	46752	13,4	30,0
$\text{UB}^{\text{LP+IP}}(5)$	19,2	3,5	19,2	6,8	38,8	43264	46369	13,4	29,6
$\text{UB}^{\text{LP+IP}}(2)$	19,2	3,5	27,4	15,7	46,6	43260	46187	13,4	29,4
320 Fahrten									
$\text{UB}^{\text{IP}}(50)$	58,8	-	55,6	3,0	114,4	72923	79985	23,2	52,6
$\text{UB}^{\text{IP}}(10)$	58,8	-	69,2	15,4	128,0	72935	79460	23,2	52,0
$\text{UB}^{\text{IP}}(5)$	58,8	-	80,2	30,0	139,0	72940	78877	23,2	51,4
$\text{UB}^{\text{IP}}(2)$	58.8	-	131,8	77,4	190,6	72915	78332	23,2	51,0
$\text{UB}^{\text{LP+IP}}(50)$	90,3	26,6	42,2	2,8	132,5	73192	78203	23,2	51,0
$\text{UB}^{\text{LP+IP}}(10)$	90,3	26,6	53,0	13,2	143,3	73204	78032	23,2	50,8
$\text{UB}^{\text{LP+IP}}(5)$	90,3	26,6	66,4	26,0	156,7	73199	77926	23,2	50,6
$\text{UB}^{\text{LP+IP}}(2)$	90,3	26,6	101,6	62,0	191,9	73121	77854	23,2	50,6

Tabelle 5.5: Unterschiedliche Strategien zur Berechnung einer zulässiger Lösung.

In der Tabelle wird die durchschnittliche Gesamtlaufzeit t_{gesamt} sowie die Zeit für die LP- und IP-Phase t_{LP} und t_{IP} in Minuten dargestellt, wobei für die letzten beiden neben der Gesamt-Phasenzeit (*ges.*) auch die darin enthaltene Zeit für die Berechnung der zulässigen Lösung (*UB*) extra angegeben wird. Weiterhin zeigt die Tabelle die Durchschnitte (über 5 Instanzen) für die erreichten LP- und IP-Werte und für die Anzahl der Umläufe und Dienste in den resultierenden Plänen.

Vergleichen wir zunächst wie die unterschiedlichen Frequenzen T_{freq} für die Berechnung einer zulässigen Lösung auf die Lösungsqualität bzw. Laufzeit auswirken.

Aus der Tabelle wird deutlich, dass eine häufigere Generierung eines Dienstplans zu dem jeweils aktuellen Umlaufplan zwar bis zu 50% mehr Laufzeit kostet, aber auch hilft, eine bessere Gesamtlösung zu finden. Somit kann durch die Wahl einer passenden Frequenz T_{freq} eine je nach Anforderungen für eine konkrete Situation geeignete Balance zwischen der Lösungsqualität und Laufzeit eingestellt werden.

Vergleicht man die beiden Strategien $\text{UB}^{\text{IP}}(T_{\text{freq}})$und $\text{UB}^{\text{LP+IP}}(T_{\text{freq}})$ miteinander, stellt man fest, dass die zweite Strategie zwar zusätzliche Laufzeit für die Berechnung zulässiger Lösungen während der LP-Phase in Anspruch nimmt, kompensiert jedoch diese Zeit zum größten Teil in der IP-Phase. Der Grund dafür ist eine ständig verbessernde obere Schranke, die in dem Subgradienten-Verfahren für das Update der Lagrange-Multiplikatoren benutzt wird, was die duale Suche schon während der LP-Phase genauer macht. Aber auch die IP-Phase startet somit mit besseren Lagrange-Multiplikatoren und einer besseren oberen Schranke. Dies führt dazu, dass das Subgradienten-Verfahren in der IP-Phase sowohl schneller terminiert als auch häufig eine bessere Lösung finden kann. Aus der Tabelle wird deutlich, dass die Gesamtlaufzeiten bei beiden Strategien in etwa vergleichbar sind. Allerdings ist die resultierende Lösung bei $\text{UB}^{\text{LP+IP}}(T_{\text{freq}})$ besser als bei $\text{UB}^{\text{IP}}(T_{\text{freq}})$. Somit empfehlen wir auch während der LP-Phase, in jeder Column-Generation-Iteration für jeweils beste Lagrange-Multiplikatoren eine zulässige Gesamtlösung zu berechnen.

Während der IP-Phase werden keine neuen Dienste für das MD-VCSP nachgeneriert. Das Lagrange-Dual-Problem wird mit den Diensten gelöst, die in der LP-Phase gefunden wurden. Eine Möglichkeit diese Einschränkung zu schwächen ist die Idee Column-Generation in die IP-Phase einzubinden. Dabei besteht die IP-Phase ähnlich zu der LP-Phase aus mehreren Hauptiterationen, zwischen denen neue Spalten gesucht und dem eingeschränkten Master-Problem hinzugefügt werden. In jeder Iteration wird das Lagrange-Dual-Problem gelöst, was im Vergleich zur LP-Phase viel mehr Zeit in Anspruch nimmt, da eins der Lagrange-Unterprobleme das $\mathcal{NP}$-harte MDVSP ist. In unseren Tests hat sich diese Strategie nicht bewährt. Sowohl die untere Schranke als auch der IP-Wert konnte kaum verbessert werden, was in keiner Relation zu deutlich höherem Zusatzaufwand für das mehrfache Lösen des Lagrange-Dual-Problems in der IP-Phase steht.

Eine weitere Möglichkeit die oben beschriebene Einschränkung zu schwächen, ist bei der Bestimmung einer zulässigen Lösung das umlaufbasierte Dienstplanungsproblem nicht nur auf vorhandene Spalten einzuschränken. Praktisch heißt das, dass beim Lösen des CSP der Freiheitsgrad für die Konstruktion möglicher Dienste nur durch den zugrundeliegenden Umlaufplan und die üblichen Dienstregel und nicht durch die für MD-VCSP generierte Dienstmenge begrenzt ist. Durch diese Vorgehensweise konnte die Qualität der resultierenden Lösung signifikant verbessert werden. In Tabelle 5.6 werden die vorgeschlagene Strategie (*CSP: frei*) und die

eingeschränkte Variante (*CSP: eingeschränkt*) für die Berechnung der zulässigen Lösung gegenübergestellt.

	CSP: eingeschränkt		CSP: frei	
UB-Strategie	t_{gesamt}	Dienste	t_{gesamt}	Dienste
160 Fahrten				
$\text{UB}^{\text{IP}}(50)$	36,0	32,0	36,3	31,2
$\text{UB}^{\text{IP}}(10)$	38,8	31,8	40,4	30,6
$\text{UB}^{\text{IP}}(5)$	40,5	31,8	44,9	30,0
$\text{UB}^{\text{IP}}(2)$	42,3	31,4	51,6	29,8
$\text{UB}^{\text{LP+IP}}(50)$	35,2	31,6	31,6	30,0
$\text{UB}^{\text{LP+IP}}(10)$	31,6	31,2	34,4	30,0
$\text{UB}^{\text{LP+IP}}(5)$	32,4	30,8	38,8	29,6
$\text{UB}^{\text{LP+IP}}(2)$	37,8	30,4	46,6	29,4
320 Fahrten				
$\text{UB}^{\text{IP}}(50)$	113,8	53,4	114,4	52,6
$\text{UB}^{\text{IP}}(10)$	124,8	53,0	128,0	52,0
$\text{UB}^{\text{IP}}(5)$	128,7	52,6	139,0	51,4
$\text{UB}^{\text{IP}}(2)$	169,6	52,0	190,6	51,0
$\text{UB}^{\text{LP+IP}}(50)$	128,8	53,0	132,5	51,0
$\text{UB}^{\text{LP+IP}}(10)$	133,4	52,4	143,3	50,8
$\text{UB}^{\text{LP+IP}}(5)$	135,6	52,2	156,7	50,6
$\text{UB}^{\text{LP+IP}}(2)$	148,3	51,8	191,9	50,6

Tabelle 5.6: Freie vs. eingeschränkte CSP bei der Berechnung zulässiger Lösung.

Es ist deutlich zu erkennen, dass die freie Variante für CSP zwar mehr Rechenzeit beansprucht, da der Lösungsraum für die Dienstkonstruktion deutlich größer ist, ist aber bezüglich der Lösungsqualität (Anzahl der Dienste) weit überlegen. Im weiteren Verlauf der Arbeit wird nur diese Strategie eingesetzt.

5.9 Zusammenfassung

In diesem Kapitel wurde einer der zentralen, in Rahmen der vorliegenden Arbeit entwickelten Ansätze zur Verplanung von Umläufen und Diensten vorgestellt. Der Ansatz basiert auf einem Column-Generation-Verfahren in Verbindung mit Lagrange-Relaxation.

Wir präsentierten eine neue Formulierung des integrierten Umlauf- und Dienstplanungsproblems mit mehreren Depots. Das zugrunde liegende Netzwerkmodell wurde zum ersten Mal mit einer neuartigen Modellierungstechnik als Time-Space-

Netzwerk formuliert, die dank ihrer Struktur zu einer erheblichen Reduktion der Netzwerkgröße führt. Demzufolge besitzt die davon abgeleitete mathematische Formulierung des Problems wesentlich weniger Entscheidungsvariablen als die vergleichbaren Formulierungen aus der Literatur. Weiterhin konnten viele netzwerkbasierten Unterprobleme, die im Ansatz zu lösen sind, viel schneller und effizienter bewältigt werden.

Der entwickelte Lösungsansatz besteht aus zwei Phasen, einer LP-Phase und einer IP-Phase. Zunächst wird in der Hauptschleife des Column-Generation-Verfahrens (Abschnitte 5.3 - 5.5) eine möglichst gute untere Schranke für die Gesamtlösung ermittelt und eine Menge von Spalten/Diensten generiert, die für das Finden einer guten zulässigen Lösung für MD-VCSP in der IP-Phase relevant sind. Im eingeschränkten Master-Problem wird die mathematische Formulierung mit Hilfe einer Lagrange-Relaxation vereinfacht. Dabei werden erstens die Kopplungsbedingungen relaxiert, die die Umlauf- und Dienstplanungskomponenten zusammenführen und für eine konsistente Überdeckung der Fahrten sorgen. Die notwendige Kopplung der beiden Teile erfolgt nun implizit durch die Modifikation der Zielfunktion mit Lagrange-Multiplikatoren. Zweitens wird auch eines der beiden resultierenden Unterprobleme, nämlich das Mehrdepot-Umlaufplanungsproblem, vereinfacht, indem eine seiner beiden Restriktionsmengen ebenfalls mit Hilfe der Lagrange-Relaxation eliminiert wird. Je nach verbleibenden Nebenbedingungen reduziert es sich zu mehreren kleinen SDVSP (Relaxation I) oder einem großen SDVSP (Relaxation II), die jeweils in polynomieller Zeit gelöst werden können. Unseren Tests zufolge kann das eingeschränkte Master-Problem unter Anwendung Relaxation I schneller gelöst werden. Der Unterschied wird insbesondere bei größeren Probleminstanzen deutlich.

Zur Lösung des resultierenden Lagrange-Dual-Problems wurden zwei Lösungsalgorithmen untersucht. Der erste ist das weit verbreitete Subgradient-Verfahren. Wir untersuchten unterschiedliche in der Literatur vorgestellte Modifikationen des Verfahrens sowohl einzeln als auch miteinander kombiniert. Die endgültig eingesetzte Version beinhaltet davon solche, die zur deutlichen Performance-Steigerung, wie Qualität der unteren Schranke, Konvergenzverhalten bzw. Anzahl der Iterationen sowohl im Verfahren selbst als auch im gesamten Column-Generation-Ansatz, beigetragen haben. Der zweite untersuchte Algorithmus, der zur Lösung des eingeschränkten Master-Problems eingesetzt wurde, ist der Volume-Algorithmus. Wir untersuchten, wie gut diese relativ neue Methode für unsere Problemstellung mit dem alt bewährten Subgradienten-Verfahren konkurrieren kann. Dabei wurden einige in der Literatur vorgestellten Varianten und Erweiterungen der ursprünglichen Version untersucht. Allerdings zeigen die durchgeführten Tests, dass der Volume-Algorithmus sich weder bezüglich der Lösungsqualität noch Lösungszeit gegen die fortgeschrittene Version des Subgradienten-Verfahren durchsetzten konnte.

Zur Lösung des Pricing-Problems wurde ein zweistufiges Verfahren angewendet. Zunächst wird eine Menge zulässiger Dienststücke mit Hilfe eines Algorithmus zur Berechnung kürzester Wege auf einem speziellen Dienststückerzeugungsgraphen konstruiert. Anschließend werden daraus gültige Dienste durch Aufzählung möglicher Dienststück-Kombinationen zusammengebaut. Die Überprüfung, ob eine Kombination einem zulässigen Dienst entspricht, ist teuer, da dafür alle relevanten Dienstregeln überprüft werden müssen. Um diesen Zusatzaufwand in jeder Column-Generation-Iteration zu vermeiden, entwickelten wir eine spezielle Datenstruktur, die erweiterte Dienstsequenz, die in der Lage ist, eine große Menge solcher Kombinationen explizit abzubilden. Dadurch kann die Gültigkeitsüberprüfung nur einmal am Anfang des Verfahrens durchgeführt werden. Danach erfolgt die Suche nach Diensten mit negativen reduzierten Kosten in jeder Iteration von Column-Generation nur unter zulässigen Kombinationen. Durch diese Maßnahme kann die Zeit für das Pricing-Problem drastisch reduziert werden. Allerdings muss man sagen, dass diese Methode nur für Dienstarten angewendet werden kann, die aus maximal zwei Dienststücken bestehen, was aber in vielen Verkehrsbetrieben der Fall ist.

Ein weiterer Punkt, der in diesem Kapitel untersucht wurde, ist das Spaltenmanagement. Üblicherweise existieren, besonders in den ersten Iterationen, sehr viele Spalten mit negativen reduzierten Kosten, die das eingeschränkte Master-Problem potentiell verbessern können. Wir schlugen eine Strategie vor, die, ohne einen großen Zusatzaufwand zu verursachen, nur die erfolgversprechendsten Spalten findet und dem Master-Problem vorschlägt. Außerdem wurden Ideen diskutiert, die durch die Einführung von Zufallskomponenten in den Suchalgorithmus für eine bessere örtliche und zeitliche Verteilung bei der Menge von Spalten-Kandidaten sorgen. Die beiden Maßnahmen haben in unseren Experimenten zu einer enormen Verringerung der Gesamtanzahl der Column-Generation-Iterationen und damit zu einer erheblichen Beschleunigung des gesamten Verfahrens geführt.

In der IP-Phase des Lösungsprozesses wird eine möglichst gute ganzzahlige Lösung für das MD-VCSP gesucht. Dafür werden in der Originalformulierung nur die Kopplungsbedingungen mit Hilfe der Lagrange-Relaxation relaxiert. Das MD-VCSP zerfällt dabei in ein triviales Auswahlproblem und ein $\mathcal{NP}$-hartes MDVSP. Die Kopplung zwischen Umläufen und Diensten erfolgt implizit über die Lagrange-Multiplikatoren der relaxierten Kopplungsbedingungen. Das Lagrange-Dual-Problem wird mit einer fortgeschrittenen Variante der Subgradienten-Algorithmus gelöst. In der IP-Phase wird in jeder Iteration ein zulässiger Umlaufplan als Lösung eines der Lagrange-Unterprobleme bestimmt. Wir benutzen diese Lösung, um darauf ein umlaufbasiertes Dienstplanungsproblem zu lösen. Die beiden Pläne ergeben eine zulässige Gesamtlösung, deren Optimalität allerdings nicht garantiert werden kann.

Um die Wahrscheinlichkeit für das Finden einer guten Gesamtlösung zu erhöhen, kann das umlaufbasierte Dienstplanungsproblem nicht nur für den letzten Umlaufplan, sondern auch für mehrere Umlaufpläne, die im Laufe der IP-Phase bestimmt werden, ausgeführt werden. Wie oft eine zulässige Lösung damit berechnet wird, ist ein Kompromiss zwischen der Laufzeit und der Lösungsqualität, was von den Anforderungen im konkreten Fall abhängt. Außerdem hat sich die Strategie, eine zulässige Lösung schon während der LP-Phase zu berechnen, als sinnvoll erwiesen. Die dafür in der LP-Phase zusätzlich beanspruchte Rechenzeit konnte durch eine kürzere IP-Phase zum größten Teil kompensiert werden.

Beim Lösen der umlaufbasierten Dienstplanungsprobleme zur Berechnung zulässiger Gesamtlösungen wurde die Variante untersucht, bei der der Freiheitsgrad für die Konstruktion möglicher Dienste nicht durch die für MD-VCSP generierte Dienstmenge begrenzt war. Durch diese Erweiterung konnte die Qualität der resultierenden Lösung signifikant verbessert werden.

Kapitel 6

Adaptive Teilintegration von Umlauf- und Dienstplanung

Die am weitesten verbreitete Vorgehensweise der Verplanung von Umläufen und Diensten ist streng sequenziell. Zuerst werden Umläufe für Fahrzeuge festgelegt und erst danach die passenden Dienste dazu. Das grenzt den Lösungsraum für das Finden guter Dienste bzw. eines effizienten Dienstplans erheblich ein, da die Fahrzeugrouten fest vorgegeben sind. Im Vergleich zu dieser umlaufbasierten Dienstplanung bietet eine simultane Betrachtung der beiden Planungsprobleme viel mehr planerische Freiheitsgrade, insbesondere bei der Bildung der Dienste. Hinzu kommt, dass die Personalkosten in der Regel die fahrzeugbezogenen Betriebskosten dominieren. Allerdings stößt die integrierte Betrachtung aus Komplexitätsgründen schon für Probleme mittlerer Größe an ihre Grenze. Daher bleibt eine sequenzielle Abarbeitung der Umlauf- und Dienstplanungsprobleme für viele Verkehrsbetriebe immer noch die einzige praktikable Alternative.

Einen Kompromiss zwischen der sequenziellen und simultanen Behandlung der beiden Planungsprobleme schaffen Methoden, die eine gewisse Kopplung der beiden Probleme anstreben. Zum Beispiel werden bei der Bildung der Umläufe einige Aspekte der Dienstplanung mitberücksichtigt, sodass der resultierende Umlaufplan etwas „dienstplantauglicher" wird. Für eine Übersicht über teilintegrierte Umlauf- und Dienstplanung siehe Unterabschnitt 4.3.1.

In Rahmen der vorliegenden Arbeit wurde ein Ansatz entwickelt, der trotz einer sequenziellen Vorgehensweise eine gewisse Interaktion zwischen der Umlauf- und Dienstplanung erlaubt (siehe auch [Gintner et al., 2005b] und [Gintner et al., 2007]). Anstatt eines einzigen Umlaufplans wird bei der Dienstplanung ein Bündel von optimalen bzw. gleichwertigen Umlaufplänen implizit betrachtet. Dafür werden die beiden Planungsprozesse aneinander gekoppelt. Diese Kopplung kann auch indirekt passieren, indem ein zunächst vorgegebener Umlauf nachträglich so angepasst wird

(ohne dass die fahrzeugbezogenen Kosten sich ändern), dass auf seiner Basis ein besserer Dienstplan möglich wird. Wir klassifizieren den entwickelten Ansatz als *adaptive Teilintegration von Umlauf- und Dienstplanung*.

Der Rest des Kapitels ist wie folgt aufgebaut. Zunächst wird der neue Ansatz im Hinblick auf die Interaktion zwischen der Umlauf- und Dienstplanung detailliert beschrieben (Abschnitt 6.1). Danach werden das modifizierte Dienstplanungsproblem (Abschnitt 6.2) gefolgt von der Prozedur zur nachfolgenden Ableitung des „richtigen" Umlaufplans (Abschnitt 6.3) vorgestellt. Im Abschnitt 6.4 wird die Idee diskutiert, wie die neuartige Dienstplanung von der Umlaufplanung entkoppelt und selbstständig gelöst werden kann. Neben einer direkten Anwendung der vorgestellten Methode zur Verplanung von Umläufen und Diensten kann sie auch zur Lösung der Unterprobleme im vollständig integrierten Lösungsansatz aus dem Kapitel 5 eingesetzt werden (Abschnitt 6.5). Schließlich präsentieren wir die Ergebnisse der durchgeführten Validierung für den entwickelten Ansatz (Abschnitt 6.6) und eine Zusammenfassung des Kapitels (Abschnitt 6.7).

6.1 Interaktion zwischen Umlauf- und Dienstplanung

Es ist bekannt, dass eine streng voneinander getrennte Betrachtung der Umlauf- und Dienstplanungsprobleme den Lösungsraum stark eingrenzt. Eine Interaktion der beiden Prozesse kann zu einer besseren Gesamtlösung führen. Denkbar wären dabei Methoden, die während einer Planungsphase gewisse Anforderungen der darauf folgenden Phase mitberücksichtigen (siehe z. B. [Kliewer et al., 2006]). Eine weitere Möglichkeit zur Interaktion wäre die in einer Phase generierte Lösung bei Bedarf in der darauf folgenden Phase nachträglich anzupassen.

Den Impuls für die Entwicklung des vorgestellten Ansatzes gab die Beobachtung, dass die Umlaufpläne sehr häufig nicht eindeutig sind, d.h. es existiert oft mindestens ein alternativer Umlaufplan mit gleichen Kosten (siehe Unterabschnitt 6.1.1). Im Prinzip kann für jede dieser Alternativen ein zugehöriger Dienstplan berechnet werden. Die Kombination mit den besten Gesamtkosten bestimmt die gesuchte Lösung. Allerdings ist dieser Prozess sehr zeitintensiv und außerdem ist es nicht klar, wie die zahlreichen Alternativen von dem gegebenen Umlaufplan abgeleitet werden sollen.

In dem präsentierten Verfahren werden die Lösungsprozesse für die Umlauf- und Dienstplanung miteinander gekoppelt. Trotz des einmaligen Lösens des Dienstplanungsproblems werden dabei als Vorlage mehrere (wenn vorhanden) optimale Umlaufpläne implizit betrachtet. Der resultierende Dienstplan bestimmt nachträglich

den dazu passenden optimalen Umlaufplan.

6.1.1 Mehrdeutigkeit von Umlaufplänen

In diesem Unterabschnitt wird anhand eines Beispiels gezeigt, wie eine einfache Anpassung eines Umlaufplans zu einer besseren Dienstplanlösung führen kann.

Sei ein Umlaufplan mit zwei Umläufen gegeben, die nun mit Diensten besetzt werden sollen, siehe Abbildung 6.1. Das erste Fahrzeug startet mit einer Ausrückfahrt d_1 aus dem Depot zur Haltestelle A, führt eine Fahrgastfahrt f_1 aus, fährt leer von C nach B (dh_1), wartet dort eine Weile und nach Ausführung von zwei weiteren Fahrgastfahrten f_2 und f_3 kehrt es wieder ins Depot zurück. Das zweite Fahrzeug rückt aus (d_3), bedient die Fahrgastfahrten f_4 und f_5 und kehrt wieder heim (d_4). Die maximale ununterbrochene Lenkzeit betrug 4:30 Stunden und die minimale Pausendauer 30 Minuten. Somit kann das erste Fahrzeug nicht komplett von einem Fahrer gefahren werden. Somit sind für die Bedienung der beiden Umläufe insgesamt drei Dienste notwendig.

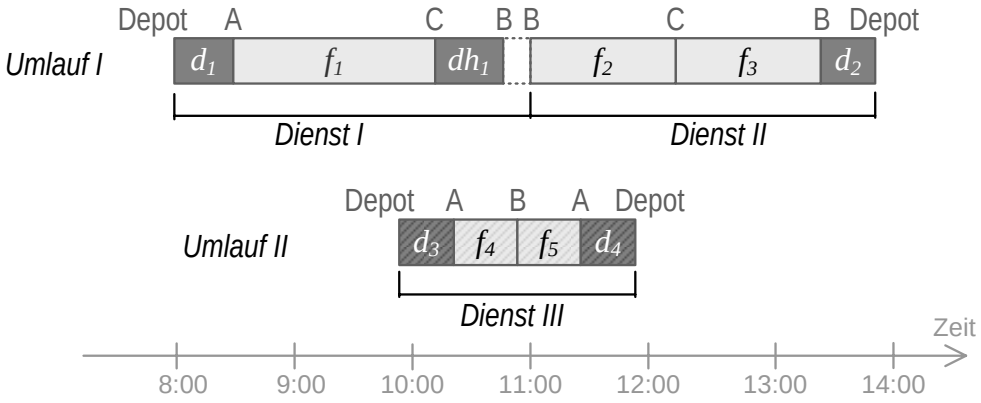

Abbildung 6.1: Zwei Beispiel-Umläufe mit darauf geplanten Diensten

Bei einer genauen Betrachtung des Umlaufplans stellt man fest, dass es einen Zeitpunkt kurz vor 11:00 Uhr gibt, an dem beide Fahrzeuge an der Haltestelle B sind. Würde man ab diesem Zeitpunkt die Verläufe der beiden Umläufe vertauschen, dann erhielte man einen alternativen Umlaufplan mit den gleichen Kosten, da die Vertauschung keine zusätzlichen Kosten, wie Leerfahrten oder Wartezeiten, hervorruft[1]. Als Nebeneffekt stellt sich heraus, dass der alternative Umlaufplan nur mit insgesamt zwei Diensten befahrbar ist (siehe Abbildung 6.2).

[1] Zur Erinnerung: Unter einer Fahrgastfahrt verstehen wir eine Linienfahrt zwischen zwei Endhaltestellen (mit planmäßigen Zwischenhalten an vorgesehenen Haltestellen). Somit sind alle Fahrgäste spätestens beim Erreichen der Endhaltestelle befördert.

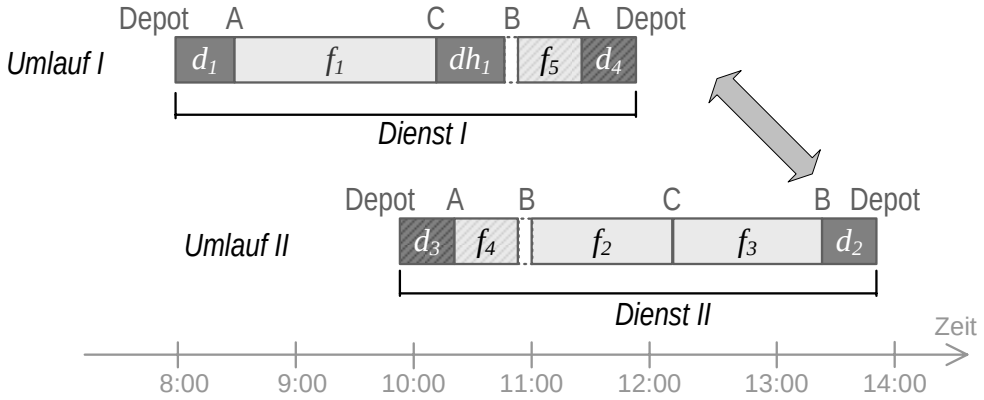

Abbildung 6.2: Einsparung eines Dienstes durch Umgestaltung der Umläufe

Diese Eigenschaft der Mehrdeutigkeit von Umlaufplänen wird in unserem Verfahren ausgenutzt. Anstatt eines einzigen Umlaufplans, betrachten wir ein Bündel von optimalen (bzw. gleichguten) Umlaufplänen während der Dienstplanung. Im Vergleich zu anderen teilintegrierten Ansätzen streben wir nach einer besseren Gesamtlösung nicht auf Kosten eines schlechteren Umlaufplans, sondern wir „bewegen" uns im Suchraum der optimalen (bzw. gleichguten) Umlaufpläne.

Wichtige Fragestellungen sind, wie man ein Bündel solcher Umlaufpläne erstellt und wie ein mehrfaches Lösen des Dienstplanungsproblems vermieden werden kann. Das obige Beispiel ist trivial. Der präsentierte „Zweier-Tausch" ist nur die einfachste Möglichkeit die Umläufe umzuordnen. Die realen Umlaufpläne sind dagegen viel umfangreicher und dichter. Sie erlaubt auch fortgeschrittene mehrfache Vertauschungen, die sehr komplex sein können. In unserem Verfahren tritt dieses Problem allerdings gar nicht auf. Eine spezielle Modellierung des Umlaufplanungsproblems bzw. dessen Lösung bildet alle diese Umlaufpläne implizit ab. Dieses Bündel wird an das gekoppelte Dienstplanungsproblem weitergereicht, was nur einmal gelöst wird.

6.1.2 Flusslösung des TSN-basierten Umlaufplanungsproblems

Das Time-Space-Netzwerk und die darauf basierte Modellierung des Umlaufplanungsproblems (bzw. MD-VCSP) wurden bereits im Unterabschnitt 5.2.1 (bzw. 5.1.1) beschrieben. In solchen Modellen werden die Umläufe durch gerichtete Pfade von dem ersten bis zum letzten Depotknoten abgebildet. Ein entscheidender Unterschied von TSN-basierten Netzwerkmodellen im Vergleich zu herkömmlichen Connection-basierten Netzwerkmodellen besteht in einer Aggregation von Warte- und Leerfahrten. Somit ist die Flussgröße durch solche Kanten nicht auf eins begrenzt,

sondern kann mehrere Fahrzeuge repräsentieren, die diese Kanten „befahren" (jede Flusseinheit repräsentiert ein Fahrzeug). Dies führt dazu, dass die resultierende Flusslösung des TSN-basierten Netzwerkproblems nicht unbedingt aus disjunkten Pfaden (Umläufen) besteht, sondern aus Pfaden, die durchaus gemeinsame Kanten bzw. Knoten beinhalten.

Um aus der Flusslösung eindeutige Umläufe zu bekommen, die den gesuchten Umlaufplan bestimmen, muss sie zunächst in disjunkte Pfade zerlegt (*dekomponiert*) werden. Allerdings ist diese Dekomposition nicht eindeutig. In der Abbildung 6.3a ist ein Abschnitt aus einem Time-Space-Netzwerk mit der dazugehörigen Flusslösung abgebildet. Offensichtlich gibt es zwei Möglichkeiten, um zwei ankommende mit zwei abfahrenden Kanten zu verbinden, wobei beide Möglichkeiten gleichwertig sind. Die Mehrdeutigkeit kann nicht nur in Knoten sondern auch in Kanten auftreten. In der Abbildung 6.3b besteht die Flusslösung der Leerfahrt-Kante zwischen zwei Haltestellen aus drei Flusseinheiten, d.h. sie repräsentiert drei Fahrzeuge, die jeweils eine Leerfahrt ausführen. Im diesem Beispiel existieren 6 Möglichkeiten, um drei oberen mit drei unteren Fahrgastfahrt-Kanten zu verbinden.

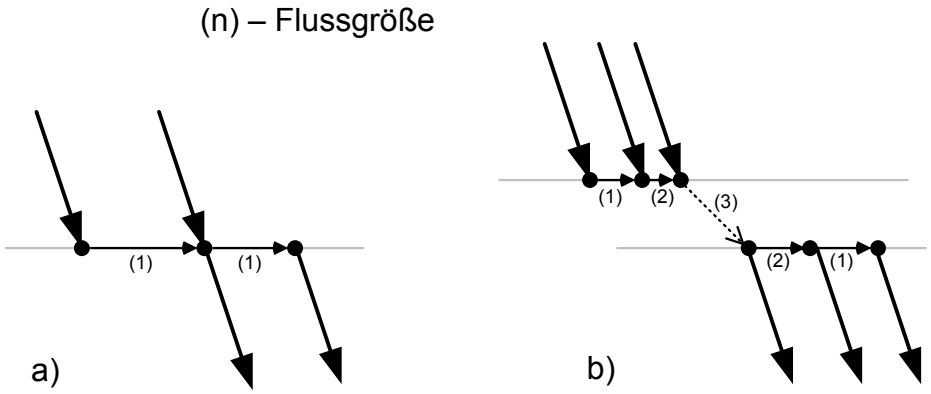

Abbildung 6.3: Flusslösung und Flussdekomposition bei der TSN-basierten Modellierung.

Eine Dekomposition der Flusslösung in disjunkte Pfade bekommt man, indem man in jedem Knoten jeder einfließenden Flusseinheit eine ausfließende Flusseinheit zuordnet. Die Dekompositionsstrategie kann - je nach Anforderungen an die Umläufe bzw. den Umlaufplan - von einfachen lokalen Zuordnungsregeln für jeden Knoten (z.B. *First-In-First-Out*) bis hin zu der komplizierten globalen Partitionierungsregel (siehe [Kliewer, 2005]) variieren.

6.1.3 Adaptive Kopplung von Umlauf- und Dienstplanung

Eine mehrdeutige Flusslösung repräsentiert implizit mehrere Umlaufpläne, die durch unterschiedliche Dekompositionen hergestellt werden können. Ein wichtiger Aspekt dabei ist, dass alle diese Umlaufpläne gleichwertig sind (d.h. sie haben gleiche fahrzeugbezogene Kosten), da sie von der gleichen Flusslösung abgeleitet sind. Diese Eigenschaft der TSN-basierten Modellierung ist der Schlüsselfaktor für die neue Methode.

Wir koppeln Umlauf- und Dienstplanung so aneinander, dass die Dienstplanung nicht erst nach der Umlaufplanung durchgeführt wird, sondern unmittelbar vor der Flussdekomposition, siehe Abbildung 6.4. Somit ist der Umlaufplan während der

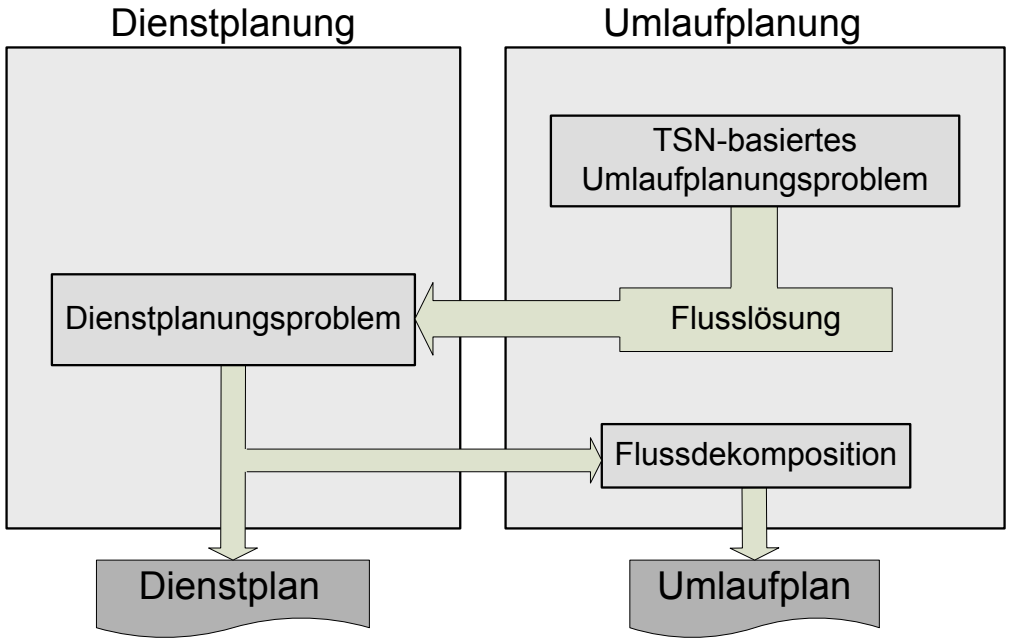

Abbildung 6.4: Adaptive Kopplung von Umlauf- und Dienstplanung

Dienstplanung noch nicht festgelegt. Stattdessen wird die komplette Flusslösung als Grundlage für Dienstgenerierung im Dienstplanungsproblem benutzt. Die Dienstgenerierung erfolgt zunächst für alle alternativen Umlaufpläne. Anschließend wird mit Hilfe des resultierenden Dienstplans ein dazu passender Umlaufplan aus der Flusslösung extrahiert, indem sie „entlang" der errechneten Dienste in disjunkte Pfade zerlegt wird.

Die Einzelheiten zur Dienstplanung auf Basis der Flusslösung werden im nächsten Abschnitt diskutiert.

6.2 Dienstplanungsproblem bei der adaptiven Teilintegration

Das Dienstplanungsproblem bei der adaptiven Teilintegration unterscheidet sich von der klassischen unabhängigen Variante. Das betrifft sowohl das zugrundeliegende Netzwerkmodell als auch die mathematische Formulierung und den Lösungsansatz selbst. Wir bezeichnen diese Variante des Dienstplanungsproblems als *aCSP*.

6.2.1 Netzwerkmodell

Die Flusslösung des Umlaufplanungsproblems kann als ein Netzwerkmodell dargestellt werden. So ein *VSP-Lösungsnetzwerk* hat die Struktur eines Time-Space-Netzwerks. Genauer gesagt ist es das ursprüngliche Planungsnetzwerk für VSP (siehe Unterabschnitt 5.1.1), aber nur mit Kanten, die wirklich in die Lösung ausgewählt wurden, d.h. Kanten mit Flussgrößen größer Null. Die Abbildung 6.5 zeigt graphisch die äquivalente Flusslösung für das Beispiel aus der Abbildung 6.1 und 6.2.

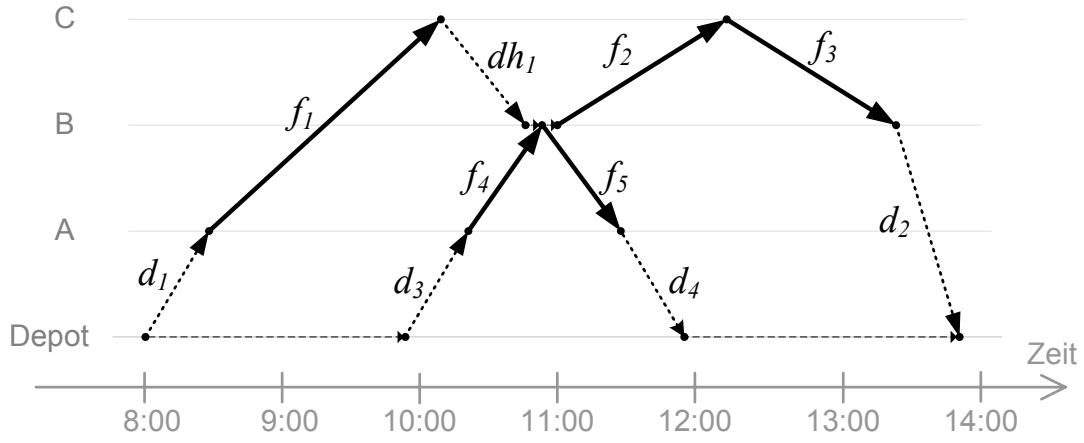

Abbildung 6.5: Äquivalente VSP-Flusslösung für das Beispiel von Abbildungen 6.1 und 6.2

Aus dem Blickwinkel der Dienstplanung repräsentiert jeder Knoten im Planungsnetzwerk einen Ablösepunkt (ähnlich zum Kapitel 5 gilt hier die gleiche Annahme, dass jede Fahrgastfahrt mit einem Ablösepunkt anfängt und endet). Jede Kante außerhalb des Depots entspricht einem Dienstelement und ein gerichteter Pfad zwischen zwei beliebigen Knoten einem Dienststück. Das Dienststück ist gültig, wenn es alle Anforderungen an die Dienststückgültigkeit, wie z.B. minimale und maximale Dienststücklänge, erfüllt und keine Depot-Wartekanten beinhaltet, da ein Aufenthalt im Depot zur Arbeitszeitunterbrechung zählt. Ein Dienst besteht aus

einem oder mehreren solcher Dienststücke (Teilpfaden) und ist genau dann gültig, wenn alle Dienstregeln erfüllt sind. Jeder Dienst wird mit Personalkosten versehen.

Wie schon im Unterabschnitt 5.1.1 beschrieben, wird im Falle mehrerer Depots ein separates Planungsnetzwerk für jedes Depot erstellt. Da in der Flusslösung für jede Fahrgastfahrt genau eine Kante existiert, sind alle Depot-Planungsnetzwerke voneinander unabhängig.

6.2.2 Mathematische Formulierung

Sei K die Menge aller zulässigen Dienste, die durch zulässige Kombinationen mehrerer Dienststücke unter Erfüllung der Dienstregeln erzeugt worden sind, und f_k ihre operativen Kosten $\forall k \in K$. Sei weiterhin E die Menge aller Kanten des gegebenen VSP-Lösungsnetzwerks, die eine Fahrzeugaktivität außerhalb des Depots abbilden (d.h. alle Kanten außer den Depot-Wartekanten und der künstlichen Zirkulationsfluss-Kante). Die Flussgröße durch jede Kante $e \in E$ ist durch die Flusslösung gegeben und beträgt z_e. Sei $K(e)$ die Menge aller Dienste, die die Kante $e \in E$ „überdecken" (d.h. diese Dienste beinhalten ein Dienstelement, das von der durch e repräsentierten Fahrzeugaktivität abgeleitet wird). Für jeden Dienst $k \in K$ definieren wir eine binäre Entscheidungsvariable x_d, die angibt, ob der Dienst d in dem resultierenden Dienstplan enthalten ist. Das adaptive Dienstplanungsproblem kann folgendermaßen formuliert werden:

$$\text{(aCSP):} \qquad \min \quad \sum_{k \in K} f_k x_k \tag{6.1}$$

$$\text{s.t.} \quad \sum_{k \in K(e)} x_k = z_e \qquad \forall e \in E \tag{6.2}$$

$$x_k \in \{0, 1\} \qquad \forall k \in K \tag{6.3}$$

In (6.1) werden die gesamten Dienstkosten minimiert. Die Nebenbedingungen (6.2) stellen sicher, dass jede Kante genau so oft mit Diensten überdeckt wird, wie ihre Flussgröße in der VSP-Flusslösung ist, oder mit anderen Worten, wie viele Fahrzeuge sie repräsentiert. Die Formulierung ist ein *Generalized Set Partitioning Problem*. Ersetzt man das Gleichheitszeichen in (6.2) durch $\geq$, erhält man ein *Generalized Set Covering Problem*, das generell einfacher zu lösen ist.

Im Falle von mehreren Depots besteht das gesamte VSP-Lösungsnetzwerk aus mehreren disjunkten Netzwerkschichten (je eine für jedes Depot). Existieren bei der Dienstplanung keine globalen depotübergreifenden Nebenbedingungen, die den Dienstmix über mehrere Depots beschränken, kann das adaptive Dienstplanungsproblem für jedes Depot (d.h. für jede Schicht des VSP-Lösungsnetzwerks) separat gelöst werden.

6.2.3 Column-Generation-Lösungsansatz

Ähnlich zu dem traditionellen umlaufbasierten Dienstplanungsproblem verwenden wir zur Lösung des aCSP ein Verfahren, das auf einer Kombination aus dem Column-Generation-Ansatz und Lagrange-Relaxation basiert. Das Grundprinzip bzw. der Ablauf von Column-Generation wurde schon mehrmals im Laufe der vorliegenden Arbeit vorgestellt (siehe z.B. Unterabschnitt 5.2.2).

Zur Lösung des eingeschränkten Master-Problems werden die Überdeckungsbedingungen (6.2) und wenn vorhanden die globalen Bedingungen mit Hilfe der Lagrange-Relaxation relaxiert. Zur Generierung neuer Spalten in der Pricing-Phase verwenden wir genauso wie beim Lösen des umlaufbasierten Dienstplanungsproblems eine zweistufige Prozedur, in der zunächst die zulässigen Dienststücke generiert und im zweiten Schritt daraus gültige Dienste mit negativen reduzierten Kosten gebaut werden. Allerdings besteht hier der Unterschied in der Art und Weise, wie die Dienststücke erzeugt werden.

Bei dem umlaufbasierten Dienstplanungsproblem wird die Menge zulässiger Dienststücke durch „Schneiden" der vordefinierten Umläufe in gültige Abschnitte gebildet. Dies ist bei dem adaptiven Dienstplanungsproblem nicht möglich, da die Umläufe nicht festgelegt sind. Stattdessen benutzen wir eine Methode, die in der Pricing-Phase des integrierten Umlauf- und Dienstplanungsproblems zur Dienststück-Generierung eingesetzt wird (siehe Unterabschnitt 5.4.1). Die Menge zulässiger Dienststücke wird dabei durch Finden kürzester Wege zwischen allen Knotenpaaren in einem speziellen *Dienststück-Erzeugungsnetzwerk* bestimmt. Dieses Netzwerk wird von dem VSP-Lösungsnetzwerk direkt abgeleitet, indem nur Kanten übernommen werden, die Bestandteil eines Dienstes sein können. Außerdem werden die Kosten auf Kanten durch entsprechende dienstbezogene reduzierte Kosten ersetzt, sodass die Kosten eines Pfades zwischen zwei Knoten den reduzierten Kosten des dadurch abgebildeten Dienststückes entsprechen.

6.2.4 Ganzzahlige Lösung

Nachdem die Hauptschleife von Column-Generation verlassen wurde, muss eine ganzzahlige Lösung gefunden werden, die den resultierenden Dienstplan bestimmt. Dafür setzen wir ein hybrides Verfahren ein, das aus einer Kombination von Branch-and-Bound-Methode und einer lokalen Suchheuristik auf Basis von Simulated Annealing besteht (siehe Unterabschnitt 5.2.2).

6.3 Nachträgliche Bildung der Umläufe

Nachdem das adaptive Dienstplanungsproblem gelöst und der Dienstplan festgelegt wurde, muss der dazu passende Umlaufplan aus dem implizit betrachteten Bündel der Umlaufpläne ausgewählt werden. Genauer gesagt muss eine Dekompositions-Strategie der VSP-Flusslösung festgelegt werden, die nachträglich einen Umlaufplan erzeugt, der zu dem gefundenen Dienstplan kompatibel ist.

Der resultierende Dienstplan stellt eine Menge von Diensten dar, die jeweils aus einem oder mehreren Dienststücken bestehen. Wie vorher beschrieben repräsentiert ein Dienststück die Arbeitszeit, die ein Fahrer auf einem Fahrzeug aktiv verbringt, d.h. ohne eine gesetzlich vorgeschriebene Pausenunterbrechung. Somit kann ein Umlauf als eine Folge von Dienststücken formuliert werden. Da ein Fahrzeug außerhalb des Depots immer von einem Fahrer besetzt sein muss, wird diese Folge nur im Depot unterbrochen. Es muss also garantiert werden, dass alle Dienstelemente (bzw. die entsprechenden Kanten im Lösungsnetzwerk) eines Dienststückes demselben Umlauf zugeordnet werden. Somit wird die Dekomposition der Flusslösung „entlang" der Dienststücke vorgenommen. Im Detail heißt das, dass in jedem Knoten alle eingehenden und ausgehenden Flusseinheiten so miteinander verknüpft werden, wie die entsprechenden Dienstelemente in den resultierenden Dienststücken verknüpft sind. Wie bereits beschrieben produzieren alle möglichen Dekompositionsstrategien der VSP-Flusslösung gleichwertige Umlaufpläne. Somit ist die Gesamtlösung der adaptiven Umlauf- und Dienstplanung mindestens so gut wie ihre sequenzielle Variante.

In unserem Beispiel auf Seite 131 werden bei dem adaptiven Dienstplanungsproblem beide möglichen Umlaufpläne implizit betrachtet (siehe Abbildung 6.5), d.h. es können Dienste für beide Pläne generiert werden. Offensichtlich wird der optimale Dienstplan aus nur zwei Diensten bestehen, nämlich $D_1 = \{d_1, f_1, dh_1, f_5, d_4\}$ und $D_2 = \{d_3, f_4, f_2, f_3, d_2\}$. Diese beiden Dienste bestimmen nachträglich auch den dazu passenden Umlaufplan, der aus den Umläufen $U_1 = \{d_1, f_1, dh_1, f_5, d_4\}$ und $U_2 = \{d_3, f_4, f_2, f_3, d_2\}$ besteht.

6.4 Entkopplung von der Umlaufplanung

Die vorgestellte teilintegrierte Umlauf- und Dienstplanung kann zu einer besseren Gesamtlösung führen. Der Schlüsselfaktor für die Kopplung der beiden Prozesse ist die Modellierung des Umlaufplanungsproblems als Time-Space-Netzwerk. Dies schränkt aber gleichzeitig die Anwendbarkeit der diskutierten Idee bei Verkehrsbetrieben ein, da nicht nur die Dienstplanung, sondern auch der Umlaufplanungsprozess ggf. umgestellt werden muss.

Aus diesem Grund wurde ein Verfahren entwickelt, das aus einem gegebenen Umlaufplan ein Lösungsnetzwerk ähnlich zu dem im TSN-basierten Umlaufplanungsproblem generiert. Diese Nachbildung hat die gleichen Freiheitsgrade wie das entsprechende originale VSP-Lösungsnetzwerk. Der Vorteil besteht aber darin, dass die Methode, mit der ein Umlaufplan ursprünglich bestimmt wurde, unwesentlich ist. Somit ist die adaptive Dienstplanung von der Umlaufplanung entkoppelt und die existierenden Methoden zur Umlaufbildung müssen nicht geändert werden. Die einzige Voraussetzung, die dabei gelten muss: Der vorgegebene Umlaufplan darf in einem bestimmten Rahmen nachträglich geändert werden. Im Wesentlichen verlangen wir, dass der alternative Umlaufplan die gleichen Kosten haben muss. Es dürfen also keine zusätzlichen Fahrzeugaktivitäten hinzugefügt werden. Das einzige Werkzeug zur Verbesserung des Umlaufplan in die Richtung „besserer Dienstplan-Tauglichkeit" ist die Vertauschung von Umlaufverläufen.

Das nachgebildete Lösungsnetzwerk hat ebenfalls die Struktur eines Time-Space-Netzwerks. Die vorgegebenen Umläufe werden nacheinander im Netzwerk abgebildet, indem für jede Fahrtaktivität ein Start- und ein Endknoten in das Netzwerk und zwar zu den entsprechenden Haltestellen eingefügt werden (innerhalb der Haltestellen sind alle Knoten nach ihrer Zeit eingeordnet). Die beiden Knoten werden mit einer Kante verbunden, die je nach der zugehörigen Fahrtaktivität eine Fahrgast-, Leer- oder Depotfahrt darstellt. Zusätzlich wird der Startknoten durch eine Wartekante mit dem Endknoten der vorangegangenen Fahrtaktivität des Umlaufs verknüpft. Somit entsteht für jeden Umlauf ein gerichteter Pfad in dem Lösungsnetzwerk.

Eine Besonderheit ist, dass zu jedem Zeitpunkt an jeder Haltestelle maximal eine Wartekante im Netzwerk existieren darf. Die „parallelen" Wartekanten werden zusammengefasst und dafür die Flussgröße, d.h. die Anzahl der zu repräsentierenden wartenden Fahrzeuge, erhöht. Das wird folgendermaßen realisiert:

- Beim Einfügen jedes einzelnen Knotens in das Netzwerk wird überprüft, ob an der entsprechenden Haltestelle und zu dem entsprechenden Zeitpunkt schon eine Wartekante existiert. Ist das der Fall, wird sie an der Stelle, wo der neue Knoten eingefügt werden soll, geteilt.
- Beim Einfügen einer Wartekante zwischen zwei Knoten A und B wird überprüft, ob es im Zeitintervall zwischen diesen Knoten schon Wartekanten gibt. Ist das der Fall, wird die Flussgröße der existierenden Kanten erhöht und die eventuell fehlenden Stücke des Weges zwischen A und B mit neuen Wartekanten vervollständigt.

Durch diese Aggregation werden die Umläufe „vermischt", was eine neue Zuordnung ermöglicht.

Das erzeugte Lösungsnetzwerk wird als Eingabe für das adaptive Dienstplanungsproblem benutzt, das wie im Abschnitt 6.2 beschrieben gelöst wird. Anschließend wird das Lösungsnetzwerk anhand der gebildeten Dienste in die ggf. neuen Umläufe zerlegt (siehe Abschnitt 6.3).

Zur Verdeutlichung betrachten wir ein Beispiel. Sei ein Umlaufplan mit drei Umläufen wie in Abbildung 6.6 gegeben.

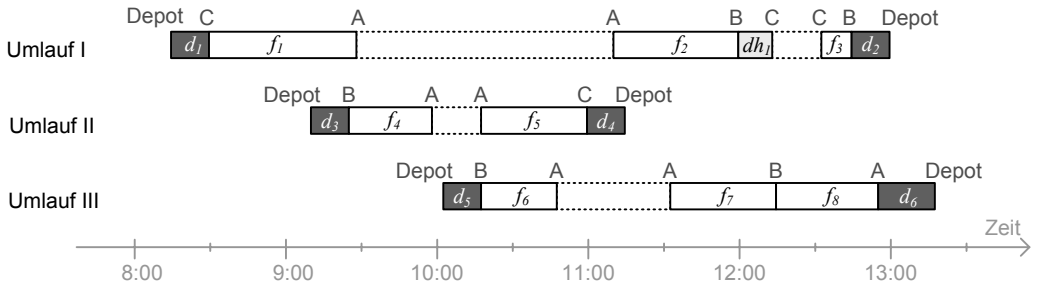

Abbildung 6.6: Ein Beispiel mit einem Depot und drei Umläufen

Abbildung 6.7 veranschaulicht den schrittweisen Aufbau des zugehörigen Lösungsnetzwerks.

Der resultierende Dienstplan kann beispielsweise aus Diensten
$D_1 = \{d_1, f_1, w_1, w_2, f_5, d_4\}$,
$D_2 = \{d_3, f_4, w_2, w_3, w_4, w_5, f_7, f_8, d_6\}$ und
$D_3 = \{d_5, f_6, w_4, f_2, dh_1, w_6, f_3, d_2\}$ bestehen.
Von diesen Diensten lässt sich der kompatible Umlaufplan mit Umläufen
$U_1 = \{d_1, f_1, f_5, d_4\}$,
$U_2 = \{d_3, f_4, f_7, f_8, d_6\}$ und
$U_3 = \{d_5, f_6, f_2, dh_1, f_3, d_2\}$ ableiten.

Wie man sieht, sind die Pfade dank der Aggregation der Wartekanten nicht mehr disjunkt. Bei der Dienstplanung lassen sich auch solche Dienststücke berücksichtigen, die aus Kanten unterschiedlicher Umläufe bestehen, auch wenn diese Umläufe keinen gemeinsamen Schnittpunkt haben, wie z.B. das Dienststück D_2. Dies konnte mit einer einfachen Vertauschungstechnik, z.B. dem Zweier-Tausch für Umläufe II und III aus Abbildung 6.6 nicht erreicht werden.

Die vorgestellte Modellierungstechnik kann zusätzlich erweitert werden, indem man für den neuen Umlaufplan es erlaubt, die Leerfahrten zeitlich zu verschieben. Ist zwischen zwei Fahrgastfahrten eine Leerfahrt notwendig, wobei sie kürzer als die Differenz zwischen der Startzeit der zweiten und der Ankunftszeit der ersten Fahrt ist, dann ist es oft unwesentlich, wann genau in diesem Zeitraum die Leerfahrt erfolgt. Bei der Bildung der Dienste kann diese Entscheidung aber zusätzlich Freiheitsgrade bringen. Würde man die Leerfahrt dh_1 im Beispiel auf der

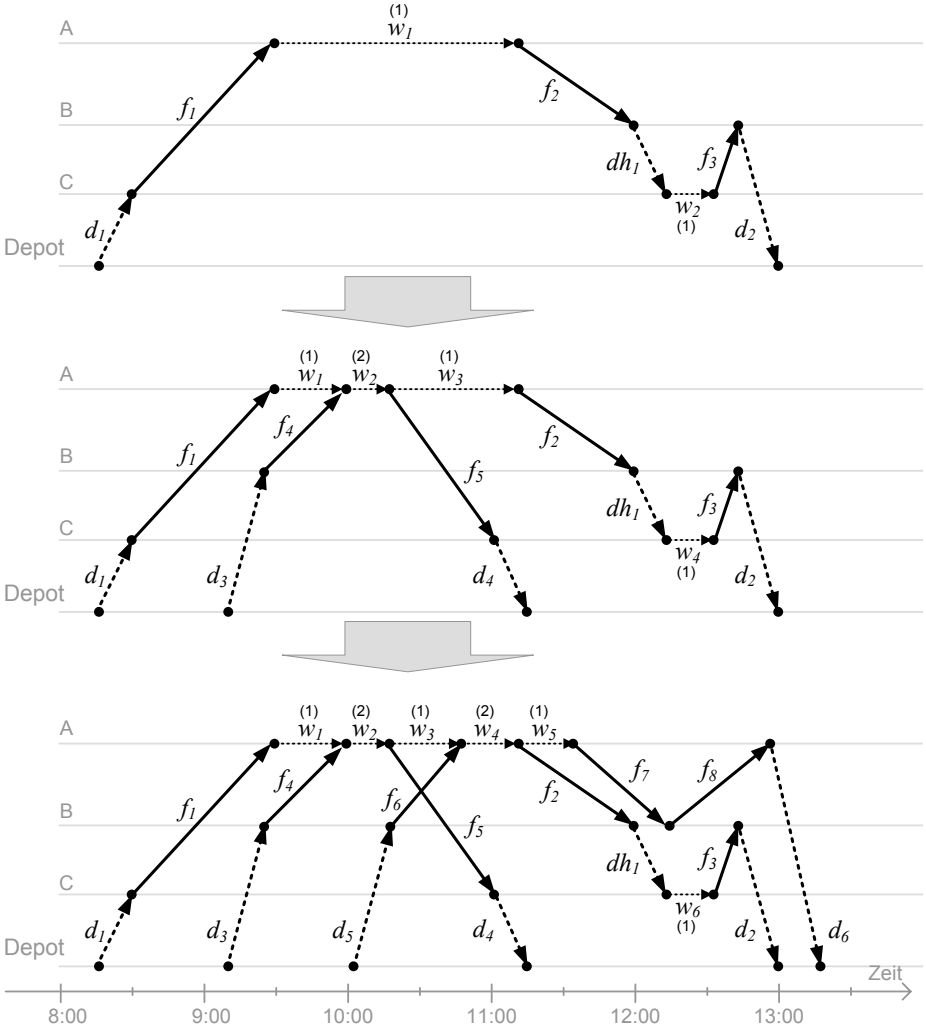

Abbildung 6.7: Schrittweise Nachbildung eines Lösungsnetzwerks

Abbildung 6.7 erst unmittelbar von der Fahrgastfahrt f_3 ausführen (d.h. eine Verschiebung der Leerfahrt dh_1 um ca. 20 Minuten nach vorne), wäre f_7 auch mit f_3 durch diese Leerfahrt verknüpfbar. Dies ermöglicht beispielsweise einen neuen Dienst $\{d_5, f_6, w_4, w_5, f_7, dh'_1, f_3, d_2\}$. Dabei bleiben die Kosten der Umläufe unverändert.

6.5 Adaptive Teilintegration als Unterproblem im Lösungsprozess des MD-VCSP

Im Lösungsansatz für die integrierte Mehrdepot-Umlauf- und Dienstplanung aus dem letzten Kapitel werden die zulässigen und zueinander kompatiblen Umlauf- und Dienstpläne nicht echt simultan, sondern in einem heuristischen Verfahren nacheinander festgelegt (siehe Abschnitt 5.6). Erst wird versucht, einen Umlaufplan zu finden, der zu einer optimalen Gesamtlösung führt. Dabei wird die Kopplung mit der Dienstplanung durch Modifizierung der Kostenfunktion mit Hilfe der Lagrange-Relaxation nur implizit realisiert. Anschließend wird für den resultierenden Umlaufplan das umlaufbasierte Dienstplanungsproblem gelöst.

Anstatt dieser sequenziellen Vorgehensweise bei der Berechnung einer zulässigen Lösung für MD-VCSP können Umlauf- und Dienstplanungsprobleme auch adaptiv teilintegriert gelöst werden. Da das Netzwerkmodell für MD-VCSP auf einem Time-Space-Netzwerk basiert, kann das vorgestellte adaptive Verfahren direkt als Unterproblem zur Berechnung gültiger Dienstpläne (bzw. Umlaufpläne) in dem vollständig integrierten Ansatz aus Kapitel 5 eingesetzt werden. Damit kann eine bessere Gesamtlösung erzielt werden.

6.6 Numerische Ergebnisse

In diesem Abschnitt berichten wir über die durchgeführten Tests zum vorgestellten Ansatz der adaptiven Teilintegration von Umlauf- und Dienstplanung. Zunächst wird der teilintegrierte Ansatz mit der traditionellen umlaufbasierten Vorgehensweise verglichen. Im zweiten Teil wird veranschaulicht, wie die Qualität des im letzten Kapitel diskutierten vollständig integrierten Ansatzes verbessert werden kann, indem in der Phase der Berechnung einer zulässigen Lösung die herkömmliche sequenzielle Umlauf- und Dienstplanung durch den vorgeschlagenen adaptiven Ansatz ersetzt wird.

Die Menge der Testfälle setzt sich aus unterschiedlichen Klassen (mit je 5 Testinstanzen) der künstlich generierten ECOPT-Instanzen von Dennis Huisman (siehe A.2) und vier realen Testfällen aus der Praxis (siehe A.3) zusammen. Ähnlich zu dem Kapitel 5.8 lassen wir fünf Dienstarten zu: Teildienst, Frühdienst, Tagesdienst, Spätdienst und geteilter Dienst (siehe A.1). Außerdem gelten die gleichen Annahmen wie im Kapitel 5 (siehe Seite 70). Die Kostenfunktion ist wie folgt definiert: Fixe Kosten von 1000 Kosteneinheiten für jeden Dienst und jeden Umlauf und variable Kosten von einer Kosteneinheit für jede Minute, die ein Fahrzeug außerhalb des eigenen Depots verbringt.

Sowohl der CSP- als auch der aCSP-Solver wurden in der Programmiersprache C# implementiert und mit .NET Framework der Version 2.0 unter Windows XP kompiliert. Alle in diesem Abschnitt dargestellten Testergebnisse wurden auf einem Dell OptiPlex GX620 Personalcomputer mit einem Pentium IV 3,4 GHz Prozessor und 2 GB RAM erreicht.

6.6.1 Adaptive Teilintegration vs. sequenzielle Planung

Zunächst wird eine herkömmliche umlaufbasierte Dienstplanung (CSP) mit der adaptiv teilintegrierten Variante (aCSP) aus diesem Kapitel verglichen. Dabei wird in beiden Fällen von einem vorgegebenen Umlaufplan ausgegangen. Bei der Umlaufbasierten Dienstplanung werden die existierenden Umläufe an Ablösepunkten in Dienstelemente geteilt und zu möglichen Dienststücken gruppiert. Bei der adaptiven Teilintegration betrachten wir die Variante, die von der Umlaufplanung entkoppelt ist (siehe Abschnitt 6.4). Dies bedeutet, dass aus den vorgegebenen Umläufen zunächst ein Lösungsnetzwerk nachgebildet wird, das als Grundlage für die Konstruktion der Dienststücke verwendet wird.

Das eingeschränkte Master-Problem wird in beiden Verfahren mit Hilfe des Subgradienten-Verfahrens gelöst. Dabei beinhaltet die eingesetzte Variante des Verfahrens alle im Unterabschnitt 5.3.3 vorgeschlagenen Erweiterungen und Verbesserungen. Für das Finden einer ganzzahligen Lösung wird der im Unterabschnitt 5.2.2 beschriebene hybride Ansatz (Branch-and-Bound in Verbindung mit primaler Suchheuristik auf Basis von Simulated-Annealing) eingesetzt.

Für alle durchgeführten Tests gilt:

- Column-Generation terminiert, wenn der Zielfunktionswert sich in den letzten 10 Iterationen um weniger als 2% verbessert hat.

- Die maximale Anzahl von Iterationen ist in Column-Generation auf 30 und im Subgradienten-Verfahren auf 500 begrenzt.

- Die maximale Anzahl neuer Spalten, die dem eingeschränkten Master-Problem in jeder Iteration hinzugefügt werden, ist auf 20000 begrenzt.

- Als Branch-and-Bound-Methode im hybriden IP-Ansatz wird der Algorithmus aus der Optimierungsbibliothek CPLEX (Version 9.1.3 mit Standardeinstellungen) verwendet. Dabei wurde die maximale Lösungszeit auf 45 Minuten bei ECOPT-Instanzen und 180 Minuten bei Praxisinstanzen begrenzt.

Künstlich generierte ECOPT-Instanzen

Für den Vergleich beider Methoden werden zunächst die künstlich erzeugten ECOPT-Instanzen von Dennis Huisman untersucht (siehe A.2). Da die Laufzeiten im Vergleich zu vollständig integrierten Ansätzen relativ klein sind, betrachten wir die 6 größten Klassen und zwar 3 Instanzklassen mit jeweils 200, 320 und 400 Fahrgastfahrten und 2 bzw. 4 Depots pro Instanz. Die Dienstplanungsprobleme werden separat für jedes Depot gelöst, deswegen sind die Probleminstanzen mit 2 Depots meistens schwieriger als die mit 4 Depots zu lösen, da die Problemgröße (Anzahl Fahrgastfahrten bzw. Umläufe) pro Depot hier tendenziell größer ist. Jede der sechs Problemklassen enthält jeweils 5 Testinstanzen.

In Tabelle 6.1 sind die Laufzeiten und die Unterschiede in der Qualität der Lösung (Anzahl der Dienste in den resultierenden Dienstplänen) für beide Lösungsverfahren gegenübergestellt. Die Ergebnisse sind in sechs Abschnitte nach Instanzklassen unterteilt. Neben den einzelnen Ergebnissen zu jeder Testinstanz (Spalten $I_1, \ldots, I_5$) sind in der Tabelle auch die Durchschnittswerte (Spalten $\oslash$) pro Instanzklasse zu finden. Außerdem ist zu jedem Abschnitt die durchschnittliche Größe der vorgegebenen Umlaufpläne wie Anzahl der Umläufe und davon abgeleitete Dienstelemente angegeben. Instanzen, bei denen mit der adaptiven Planung eine bessere Lösung erreicht werden konnte, sind zur besseren Lesbarkeit durch fett markierte Werte bei der Dienstanzahl hervorgehoben. Die Laufzeitangaben, bei denen die maximal zur Verfügung stehende Zeit von 900 Sekunden in Branch-and-Bound ausgeschöpft wurde, sind mit * gekennzeichnet. Das bedeutet, dass bei diesen Instanzen theoretisch noch weiteres Potenzial besteht, eine bessere Lösung zu finden, wenn man den Branch-and-Bound-Algorithmus länger laufen ließe.

Der entscheidende Vorteil der adaptiv teilintegrierten Vorgehensweise ist die implizite Konstruktion und Betrachtung mehrerer alternativen Umlaufpläne als Grundlage für die Erzeugung möglicher Dienste. Somit ergeben sich viel mehr Möglichkeiten für den resultierenden Dienstplan. Auf der einen Seite bedeutet das einen größeren Aufwand beim Lösen der Probleme, aber auf der anderen Seite helfen diese zusätzlichen Freiheitsgrade eine bessere Gesamtlösung für Umlauf- und Dienstplanung zu finden. Dies bestätigen auch die präsentierten Ergebnisse. Der Lösungsprozess für aCSP benötigt zwar mehr Rechenzeit, liefert aber in den meisten Fällen einen besseren Dienstplan. Dabei vergrößert sich der Qualitätsunterschied überproportional mit steigender Problemgröße.

Instanzen aus der Praxis

Als nächstes wird untersucht, ob die gewonnenen Erkenntnisse über die Überlegenheit der adaptiven Teilintegration gegenüber der herkömmlichen rein umlaufbasier-

ten Dienstplanung auch für reale Testinstanzen aus der Praxis bestätigt werden. Dafür untersuchen wir drei der größten Praxisfälle mit bis zu 2633 Fahrgastfahrten und 3 Depots (siehe Anhang A.3). Im oberen Teil der Tabelle 6.2 werden die wichtigsten Größen der getesteten Instanzen R1296, R2047 und R2633 sowie ihrer vorgegebenen Umlaufpläne zusammengefasst (Anzahl Depots, Fahrgastfahrten, Umläufe und davon abgeleitete Dienstelemente).

Neben der benötigten Laufzeit und der Anzahl der resultierenden Dienste sind in der Tabelle auch die Anzahl der zulässigen Dienststücke und zulässiger Dienste für beide Lösungsmethoden abgebildet. Diese Angaben machen noch mal deutlich, dass der Suchraum bei dem adaptiv teilintegrierten Ansatz viel größer als bei der rein umlaufbasierten Variante ist. Besonders ist der Unterschied bei R1296 zu se-

Lösungs-	Laufzeit (sek.)						Anzahl Dienste					
verfahren	I_1	I_2	I_3	I_4	I_5	⌀	I_1	I_2	I_3	I_4	I_5	⌀
200 Fahrten, 4 Depots (UP: durchschnittlich 19 Umläufe und 327 Dienstelemente)												
CSP	5	3	2	3	6	3,8	47	53	50	42	47	47,8
aCSP	11	4	4	5	8	6,4	47	53	**49**	42	47	**47,6**
320 Fahrten, 4 Depots (UP: durchschnittlich 23 Umläufe und 432 Dienstelemente)												
CSP	51	55	28	20	20	34,8	58	48	67	68	49	58,0
aCSP	56	95	34	41	45	54,2	**57**	48	67	68	49	**57,8**
400 Fahrten, 4 Depots (UP: durchschnittlich 34 Umläufe und 542 Dienstelemente)												
CSP	115	34	16	52	59	55,2	82	81	85	67	83	79,6
aCSP	122	56	44	57	76	71,0	**81**	**80**	**84**	67	**82**	**78,8**
200 Fahrten, 2 Depots (UP: durchschnittlich 19 Umläufe und 308 Dienstelemente)												
CSP	8	9	8	13	22	12,0	44	52	44	40	45	45,0
aCSP	13	30	21	25	64	30,6	44	**51**	44	**38**	45	**44,4**
320 Fahrten, 2 Depots (UP: durchschnittlich 24 Umläufe und 406 Dienstelemente)												
CSP	92	980*	63	34	300	294	57	49	56	64	43	53,8
aCSP	194	992*	995*	83	277	508	**56**	**48**	56	**63**	**42**	**53,0**
400 Fahrten, 2 Depots (UP: durchschnittlich 34 Umläufe und 527 Dienstelemente)												
CSP	59	47	92	74	983*	251	77	77	74	66	78	74,4
aCSP	351	457	365	986*	997*	631	**76**	**75**	**73**	**65**	**77**	**73,2**

* Zeitlimit von 900 Sekunden in Branch-and-Bound erreicht

Tabelle 6.1: Umlaufbasierte vs. adaptiv teilintegrierte Dienstplanung für ECOPT-Instanzen

		R1296	R2047	R2633
	Depots	2	2	3
	Fahrgastfahrten	1296	2047	2633
	Umläufe	46	114	126
	Dienstelemente	2051	2545	3075
CSP	mögliche Dienststücke	57.391	15.745	67.002
	mögliche Dienste	7.894.529	1.824.270	15.504.262
	Laufzeit (min.)	136	6	270*
	Dienste	**118**	**420**	**297**
aCSP	mögliche Dienststücke	288.374	32.100	97.006
	mögliche Dienste	188.800.526	8.038.295	31.684.492
	Laufzeit (min.)	268*	62	366*
	Dienste	**111**	**418**	**292**
Diff. Dienste (aCSP - CSP)		**-7**	**-2**	**-5**

* Zeitlimit von 180 Minuten in Branch-and-Bound erreicht

Tabelle 6.2: Umlaufbasierte vs. adaptiv teilintegrierte Dienstplanung für reale Praxisinstanzen

hen. Dort konnten, dank der zusätzlichen Flexibilität, die vorliegenden Umläufe ohne Mehrkosten umzustrukturieren, über 26 Mal mehr zulässige Dienste konstruiert werden. Dieser zusätzliche Freiheitsgrad spiegelt sich auch in der Qualität der gefundenen Lösung wider. Mit aCSP wurde für R1296 ein Dienstplan mit 7 Diensten weniger als bei CSP gefunden, was eine Einsparung von 5,9% gegenüber der herkömmlichen Lösung darstellt. Die letzte Zeile in der Tabelle 6.2 verdeutlicht nochmal die absolute Einsparung, die durch den Einsatz von aCSP gegenüber CSP erzielt wurde.

Somit stellt die in diesem Kapitel vorgestellte alternative Methode zur Behandlung von Umlauf- und Dienstplanungsproblemen eine echte Alternative zum herkömmlichen, streng sequenziellen Ansatz dar. Sie benötigt zwar etwas mehr Rechenzeit, liefert aber in den meisten Fällen eine deutlich bessere Gesamtlösung. Außerdem ist der zusätzliche Mehraufwand deutlich geringer als die Laufzeit, die bei einem vollständig integrierten Ansatz in Anspruch genommen wird. Somit kann die adaptive Teilintegration auch als eine Alternative für MD-VCSP für mittlere und große Probleme angesehen werden.

6.6.2 Adaptive Teilintegration als Unterproblem für MD-VCSP

Im Lösungsprozess für MD-VCSP wird eine zulässige Lösung mit Hilfe eines sequenziellen Ansatzes für modifizierte Umlauf- und Dienstplanungsprobleme bestimmt. Wie die letzten Ergebnisse zeigen, stellt eine adaptiv teilintegrierte Betrachtung der beiden Unterprobleme eine Alternative zu der rein sequenziellen Vorgehensweise dar.

Wir haben diese Varianten für 15 Testinstanzen aus der ECOPT-Bibliothek (siehe A.2) getestet. Die Testfälle sind in drei Gruppen je 5 Instanzen mit 80, 160 bzw. 320 Fahrgastfahrten und 4 Depots zusammengefasst. Wir untersuchten sechs Strategien für die Häufigkeit, mit der eine zulässigen Lösung in MD-VCSP bestimmt wird (siehe Unterabschnitt 5.8.2). Bei den ersten drei $UB^{IP}(50)$, $UB^{IP}(10)$ bzw. $UB^{IP}(2)$ wird die zulässige Lösung nur in der IP-Phase berechnet und zwar alle 50, 10 bzw. 2 Iterationen des Subgradienten-Verfahrens. Bei den anderen drei Strategien $UB^{LP+IP}(50)$, $UB^{LP+IP}(10)$ bzw. $UB^{LP+IP}(2)$ wird zusätzlich zur IP-Phase auch am Ende jeder Column-Generation-Iteration die Bestimmung einer zulässigen Lösung angestoßen.

In Tabelle 6.3 sind die Ergebnisse der durchgeführten Tests zusammengefasst (vgl. Tabelle 5.5). Für jede IP-Strategie und jede Instanzklasse werden die Durchschnitte über die Laufzeiten, die resultierenden IP-Werte und die Anzahl der Dienste in der besten gefundenen Gesamtlösung angegeben. Die Zeilen VCSP_{CSP} präsentieren dabei die Ergebnisse für Lösung MD-VCSP unter Verwendung der herkömmlichen, rein sequenziellen Umlauf- und Dienstplanung als Unterproblem zur Bestimmung einer zulässigen Lösung, während die Zeilen VCSP_{aCSP} dafür die Verwendung des alternativen adaptiven Ansatzes vorstellen.

Die präsentierten Ergebnisse machen deutlich, dass die Lösungsqualität für MD-VCSP durch den Einsatz eines alternativen, adaptiv teilintegrierten Verfahrens zur Bestimmung gültiger Umlauf- und Dienstpläne deutlich verbessert werden kann. Das relative Verbesserungspotenzial steigt dabei mit der Problemgröße. Für die Problemklasse mit 320 Fahrten konnten beispielsweise im Schnitt 2 Dienste pro Dienstplan eingespart werden. Auf der anderen Seite steht einer besseren Lösung auch eine längere Laufzeit gegenüber. Die Wahl einer passenden Lösungsmethode für die Umlauf- und Dienstplanung im MD-VCSP-Lösungsprozess ist, genauso wie die Wahl einer passenden IP-Strategie, ein Kompromiss zwischen der gewünschten Lösungsqualität und der zur Verfügung stehenden Rechenzeit.

	80 Fahrten			160 Fahrten			320 Fahrten		
	Zeit	IP	Dienste	Zeit	IP	Dienste	Zeit	IP	Dienste
IP-Strategie: $UB^{IP}(50)$									
$VCSP_{(CSP)}$	6,5	31185	19,8	36,3	48289	31,2	114	79985	52,6
$VCSP_{(aCSP)}$	6,4	30378	19,0	42,8	46836	30,0	110	77721	50,4
IP-Strategie: $UB^{IP}(10)$									
$VCSP_{(CSP)}$	7,9	30587	19,2	40,0	47640	30,6	128	79460	52,0
$VCSP_{(aCSP)}$	10,2	29720	18,4	47,3	46196	29,4	164	77319	49,8
IP-Strategie: $UB^{IP}(2)$									
$VCSP_{(CSP)}$	12,7	30141	18,8	51,6	46623	29,8	191	78332	51,0
$VCSP_{(aCSP)}$	14,3	29242	18,0	64,5	46011	29,2	283	76415	49,0
IP-Strategie: $UB^{LP+IP}(50)$									
$VCSP_{(CSP)}$	10,5	30453	18,8	31,6	46742	30,0	133	78203	51,0
$VCSP_{(aCSP)}$	11,7	29428	18,2	41,2	45565	28,8	141	77540	50,2
IP-Strategie: $UB^{LP+IP}(10)$									
$VCSP_{(CSP)}$	12,0	30095	18,8	34,4	46752	30,0	143	78032	50,8
$VCSP_{(aCSP)}$	13,1	28999	17,8	44,3	45740	29,0	186	76770	49,2
IP-Strategie: $UB^{LP+IP}(2)$									
$VCSP_{(CSP)}$	16,5	29530	18,2	46,6	46187	29,4	192	77854	50,6
$VCSP_{(aCSP)}$	16,5	28612	17,4	64,8	45170	28,4	295	76079	48,6

Tabelle 6.3: Umlaufbasierte vs. adaptiv teilintegrierte Dienstplanung als Unterproblem im MD-VCSP

6.7 Zusammenfassung

Die am weitesten verbreitete Vorgehensweise bei der Verplanung von Umläufen und Diensten ist streng sequenziell. Zuerst werden Umläufe für Fahrzeuge festgelegt und erst danach die passenden Dienste dazu. Das grenzt bekanntlich den Lösungsraum für das Finden guter Dienste bzw. eines effizienten Dienstplans erheblich ein, da die Fahrzeugrouten fest vorgegeben sind. Im Vergleich zu dieser umlaufbasierten Dienstplanung bietet eine simultane Betrachtung der beiden Planungsprobleme viel mehr planerische Freiheitsgrade, insbesondere bei der Bildung der Dienste. Hinzu kommt, dass die Personalkosten in der Regel die fahrzeugbezogenen Betriebskosten dominieren. Allerdings stößt eine vollständig integrierte Betrachtung aus Gründen der Komplexität schon für Probleme mittlerer Größe an ihre Grenze. Daher bleibt eine sequenzielle Abarbeitung der Umlauf- und Dienstplanungsprobleme für viele Verkehrsbetriebe immer noch die einzige praktikable Alternative.

In diesem Kapitel wurde ein adaptiv teilintegrierter Ansatz diskutiert, der trotz

einer sequenziellen Vorgehensweise eine gewisse Interaktion zwischen der Umlauf- und Dienstplanung erlaubt und dadurch zu einer besseren Gesamtlösung führt. Die Grundidee basiert auf der Beobachtung, dass ein vorgegebener bzw. während der Umlaufplanung errechneter Umlaufplan sehr häufig nicht eindeutig ist, d.h. es existiert oft mindestens ein alternativer Umlaufplan mit gleichen Kosten. Im Prinzip könnte für jede dieser Alternativen ein zugehöriger Dienstplan berechnet werden. Die Kombination mit den besten Gesamtkosten würde die Endlösung bestimmen. Allerdings ist dieser Prozess sehr zeitintensiv und außerdem ist nicht klar, wie die zahlreichen Alternativen von dem gegebenen Umlaufplan abgeleitet werden sollen.

Der entscheidende Schlüsselfaktor unseres Verfahrens ist ein spezielles Lösungsnetzwerk, das dem Dienstplanungsproblem zugrundeliegt. Dieses Netzwerk stellt im Prinzip eine Flusslösung aus einem Umlaufplanungsproblem dar, in der gerichtete Pfade von der Quelle zur Senke Umläufe repräsentieren. Zusammen bestimmen sie den resultierenden Umlaufplan. Da aber das vorgeschlagene Netzwerk eine so genannte Time-Space-Struktur besitzt, sind diese Pfade nicht disjunkt, woraus sich unterschiedliche Zerlegungen des Flusses ergeben. Alle Dekompositionen sind bezüglich ihrer Kosten gleichwertig und stellen mögliche alternative Umlaufpläne dar.

Dieses Bündel impliziter Umlaufpläne wird an die Dienstplanung übergeben. Dort erfolgt die Dienstgenerierung nicht entlang einzelner vorgegebener Umläufe, sondern gleichzeitig entlang aller möglichen Pfade im Lösungsnetzwerk. Somit legt man die Dekomposition der Pfade in Umläufe nicht vorab fest. Dies resultiert in einer viel größeren Menge möglicher Dienste. Die mathematische Formulierung als Generalized Set-Partitioning-Problem sorgt für eine korrekte Überdeckung der fahrzeugbezogenen Flusslösung durch generierte Dienste. Anschließend wird mit Hilfe des resultierenden Dienstplans ein dazu passender Umlaufplan aus der Flusslösung extrahiert, indem sie „entlang" der errechneten Dienste in disjunkte Pfade zerlegt wird. Im Vergleich zu anderen teilintegrierten Ansätzen streben wir nach einer besseren Gesamtlösung nicht auf Kosten eines schlechteren Umlaufplans, sondern wir „bewegen" uns im Suchraum der optimalen (bzw. gleichguten) Umlaufpläne.

Das Lösungsnetzwerk kann entweder direkt aus dem Lösungsverfahren des Umlaufplanungsproblems entnommen werden (Umlauf- und Dienstplanung sind gekoppelt), falls es auf Basis eines Time-Space-Netzwerks modelliert wurde, oder aus dem vorgegebenen Umlaufplan konstruiert werden. Die Nachbildung hat die gleichen Freiheitsgrade wie das entsprechende originale VSP-Lösungsnetzwerk. Der Vorteil ist aber, dass es unwesentlich ist, mit welchem Verfahren der Umlaufplan bestimmt wurde. Somit kann die adaptive Dienstplanung auch entkoppelt von der Umlaufplanung eingesetzt werden. Die einzige Voraussetzung, die dabei gelten muss: Der vorgegebene Umlaufplan darf in einem bestimmten Rahmen nachträglich geändert werden. Im Wesentlichen verlangen wir, dass der alternative Umlaufplan

gleiche fahrzeugbezogene Kosten haben muss. Es dürfen also keine zusätzlichen Fahrzeugaktivitäten hinzugefügt werden. Das einzige Werkzeug zur Verbesserung des Umlaufplan in die Richtung „besserer Dienstplan-Tauglichkeit" ist die Vertauschung von Umlaufverläufen.

Neben einer direkten Anwendbarkeit als eigenständige Methode zur Umlauf- und Dienstplanung kann das vorgestellte adaptiv teilintegrierte Verfahren auch als Werkzeug zur Bestimmung einer zulässigen Lösung in einem vollständig integrierten Ansatz aus Kapitel 5 eingesetzt werden. Die am Ende des Kapitels präsentierten Testergebnisse zeigen, dass die vorgestellte adaptive Teilintegration von Umlauf- und Dienstplanung in beiden Fällen eine echte Alternative zu dem herkömmlichen streng sequenziellen Ansatz darstellt. Sie benötigt zwar etwas mehr Rechenzeit, liefert aber in den meisten Fällen eine deutlich bessere Gesamtlösung. Außerdem ist der zusätzliche Mehraufwand deutlich kleiner als die Laufzeit, die von einem vollständig integrierten Ansatz in Anspruch genommen wird. Somit kann die adaptive Teilintegration auch als eine Alternative für VCSP für mittlere und große Probleme angesehen werden.

Kapitel 7

Fix-and-Optimize-Verfahren zur Lösung großer MD-VCSP

Sowohl die Wichtigkeit einer simultanen Behandlung von Umlauf- und Dienstplanung als auch das mögliche Einsparungspotenzial durch die Integration der beiden Planungsprozesse wurden bereits mehrmals im Laufe dieser Arbeit beschrieben. Auf der anderen Seite wird aus den präsentierten Ergebnissen deutlich, wo die Grenzen dieser Vorgehensweise zur Zeit liegen. Schon für Probleme mittlerer Größe dauert der Lösungsprozess sehr lang (siehe Abschnitt 5.8). Im letzten Kapitel wurde ein adaptiver teilintegrierter Ansatz vorgestellt, der ein Kompromiss zwischen der sequenziellen und simultanen Behandlung der Umlauf- und Dienstplanung ist. Trotz einer Kopplung der beiden Planungsprozesse ist dieses Verfahren eher sequenziell, da die gefundenen fahrzeugbezogenen Kosten nicht verschlechtert werden dürfen und es somit deutlich weniger Freiheitsgrade als der vollständig integrierte Ansatz aus Kapitel 5 gibt.

In diesem Kapitel stellen wir ein neues, approximatives Verfahren vor, das es möglich macht, auch größere Umlauf- und Dienstplanungsprobleme integriert zu lösen (siehe auch [Gintner et al., 2005c]). Es basiert auf dem vollständig integrierten Lösungsansatz für MD-VCSP aus Kapitel 5. Der Unterschied besteht darin, dass die Problemgröße im Vorfeld heuristisch verkleinert wird, indem einige Fahrten noch vor der Optimierung fixiert werden. Dadurch werden einige Entscheidungen im Vorfeld getroffen und die Problemgröße somit reduziert. Das verbleibende verkleinerte Problem wird anschließend mit dem zeitintensiveren vollständig integrierten Ansatz aus Kapitel 5 gelöst. Wir nennen unser Verfahren *Fix-and-Optimize (FaO)*. Eine ähnliche Idee wurde für das Mehrdepot-Umlaufplanungsproblem in [Gintner et al., 2005a] und [Kliewer, 2005] erfolgreich umgesetzt.

7.1 Grundschema des Verfahrens

Reale Fahrpläne weisen in der Praxis oft eine spezielle Struktur auf: Die Fahrgastfahrten sind nicht zufällig verteilt, sondern aufeinander abgestimmt. So existieren oft Fahrten bzw. Sequenzen von Fahrten, die mit großer Wahrscheinlichkeit in der optimalen Gesamtlösung von einem Fahrzeug direkt nacheinander gefahren werden. Wir bezeichnen solche Sequenzen als *stabile Fahrtketten* (analog zu [Gintner et al., 2005a] und [Kliewer, 2005]). Fixiert man solche Fahrten noch vor dem Lösen des integrierten Problems, dann wird der Suchraum verkleinert und das resultierende MD-VCSP kann schneller gelöst werden.

Die erste offene Frage ist, wie man solche stabile Fahrtketten findet. Dazu lösen wir zunächst das Umlaufplanungsproblem und das fahrplanbasierte Dienstplanungsproblem für den vorgegebenen Fahrplan unabhängig voneinander. Somit wird sowohl der optimale Umlaufplan als auch der optimale Dienstplan bestimmt. Da die beiden Planungsprobleme unabhängig voneinander und basierend auf dem Fahrplan gelöst wurden, haben die resultierenden Pläne keinen Anspruch auf gegenseitige Kompatibilität.

In der zweiten Phase werden die beiden Pläne miteinander verglichen und auf identische Sequenzen von Fahrten untersucht. Der Grundgedanke basiert auf der folgenden Überlegung: Gibt es zwei bzw. mehrere Fahrten, die sowohl im optimalen Umlaufplan als auch im optimalen (fahrplanbasierten) Dienstplan nacheinander ausgeführt werden, dann ist die Wahrscheinlichkeit groß, dass sie auch in der gesamtoptimalen Lösung nacheinander ausgeführt werden. Jede dieser Sequenzen wird als stabile Fahrtkette gekennzeichnet.

Ein Überblick über das gesamte Verfahren ist in Abbildung 7.1 dargestellt. Zur Lösung des (Mehrdepot-)Umlaufplanungsproblems wird die im Unterabschnitt 5.2.1 beschriebene Technik eingesetzt. Der Lösungsansatz für das fahrplanbasierte Dienstplanungsproblem wird im nächsten Unterabschnitt diskutiert.

Nachdem die Menge stabiler Fahrtketten identifiziert wurde, werden die betroffenen Fahrtsequenzen fixiert und das modifizierte eingeschränkte Problem vollständig integriert gelöst. Anstatt einer Modifikation des Netzwerkmodells und Lösungsansatzes für MD-VCSP aus Kapitel 5 wird ein neuer Fahrplan T' erzeugt, in dem die Fahrten einer stabilen Fahrtkette zu einer einzigen Fahrgastfahrt zusammengefasst werden. T' hat somit weniger Fahrgastfahrten als das Original T. Der große Vorteil dieser Vorgehensweise ist, dass der existierende integrierte Ansatz zur Lösung von MD-VCSP direkt und ohne Anpassungen für den neuen Fahrplan T' angewendet werden kann. Bei mehreren Depots mit unterschiedlichen Kostenfunktionen ist allerdings eine kleine Anpassung notwendig. Die Fahrtenfixierung für stabile Fahrtketten wird damit indirekt realisiert.

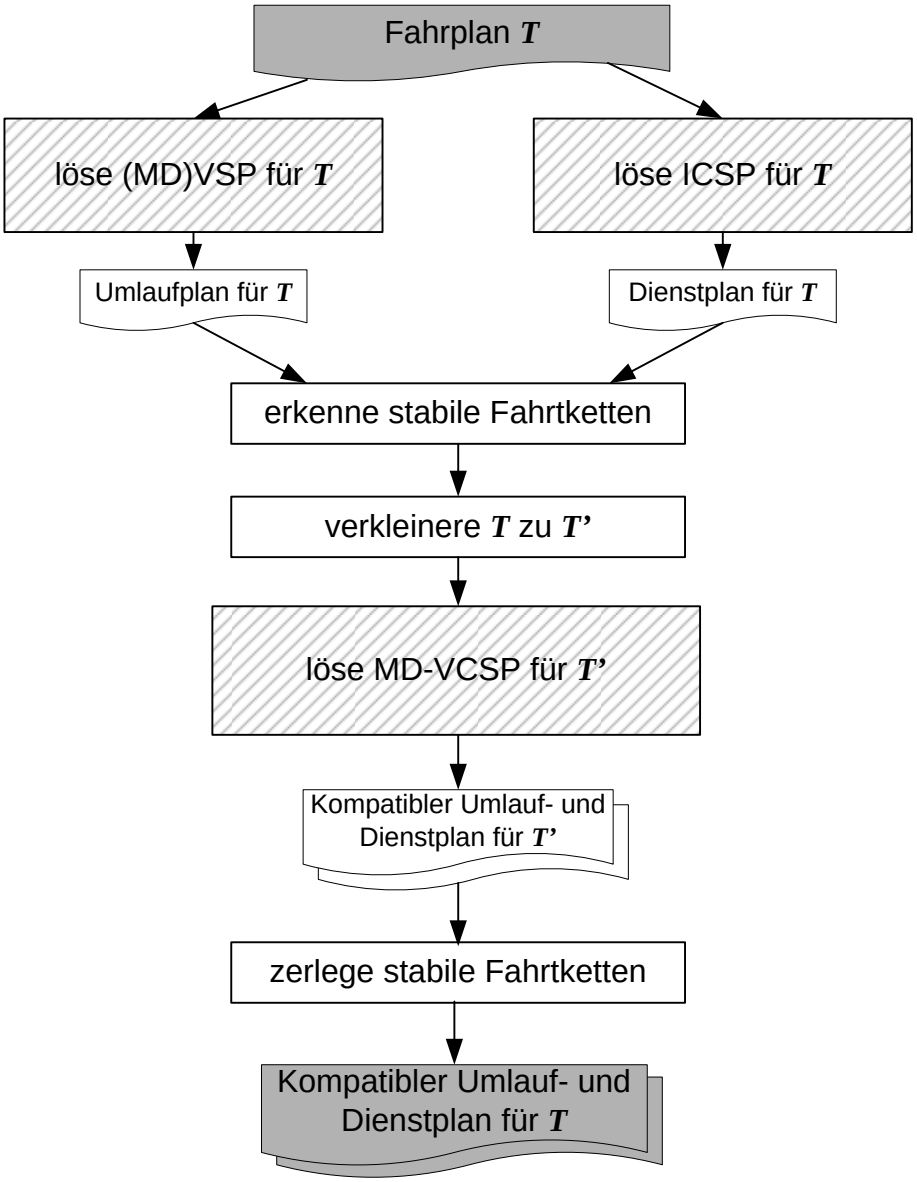

Abbildung 7.1: Grundschema des Fix-and-Optimize-Ansatzes

Bei der Erzeugung künstlicher Fahrten, die eine Fahrtkette abbilden, werden ihre Start- und Endattribute (Start- und Endhaltestelle sowie Abfahrts- und Ankunftszeit) von der ersten bzw. letzten Fahrt der Fahrtkette übernommen. Die Kosten solcher künstlichen Fahrten setzen sich aus den fahrzeugbezogenen Kosten der zugehörigen Fahrtsequenz zusammen. Bei Problemen mit mehreren Depots und/oder mehreren Fahrzeugtypen können diese Kosten für jede Depot-Fahrzeugtyp-Kombination anders ausfallen, wenn für unterschiedliche Depots bzw. Fahrzeugtypen unterschiedliche Kostenfunktionen definiert sind. Dieser Aspekt muss im inte-

grierten Modell betrachtet werden. Deswegen werden im modifizierten Fahrplan T' für jede künstliche Fahrgastfahrt die Kosten für jedes zulässige Depot mitgespeichert. Der Lösungsansatz für MD-VCSP wird so angepasst, dass beim Aufbauen des Netzwerks diese Kosten direkt übernommen werden.

Durch die Art und Weise, wie die stabilen Fahrtketten gebildet werden, wird garantiert, dass die entsprechenden künstlichen Fahrten nicht zu lang sind und von zulässigen Diensten überdeckt werden können. Außerdem muss sichergestellt werden, dass die stabilen Fahrtketten keine Aufenthalte im Depot beinhalten, da jeder Depotaufenthalt eine Dienststückunterbrechung bedeutet und die dazugehörige künstliche Fahrt dann nicht mehr einem Dienstelement entspräche. Deswegen werden stabile Fahrtketten nur für Sequenzen von Fahrten zwischen Depotaufenthalten gebildet. Allerdings kann es im Falle von mehreren Depots passieren, dass es im erzeugten Umlaufplan zwei Fahrten gibt, die durch Warten an der Anschlusshaltestelle verbunden sind, da das zugehörige Depot zu weit für einen Zwischenstopp entfernt ist. Treten die beiden Fahrten auch in dem Dienstplan in einem Dienststück auf, dann bilden sie eine stabile Fahrtkette. Allerdings kann es passieren, dass die dafür erzeugte künstliche Fahrt für ein anderes Depot nicht mehr gültig ist, da das andere Depot viel näher an der Anschlusshaltestelle ist und es günstiger wäre, zwischen den beiden Fahrten ins Depot zu fahren und dort zu warten. Somit würde so eine zusammengesetzte Fahrt einen Depotaufenthalt enthalten und die obige Annahme verletzen. Deswegen wird beim Erzeugen künstlicher Fahrten überprüft, ob die entsprechende stabile Fahrtkette in keinem der zulässigen Depots einen Aufenthalt braucht. Ist das nicht der Fall, dann wird sie zwischen betroffenen Fahrten geteilt und es entstehen zwei (bzw. mehrere) künstliche Fahrten daraus. Somit können wir garantieren, dass jede Fahrt in T' durch ein zulässiges Dienststück überdeckt werden kann.

Beim Lösen des MD-VCSP für T' wird ein Umlauf- und ein dazu kompatibler Dienstplan erzeugt. Die beiden Pläne beinhalten künstliche Fahrten, da für sie der alternative Fahrplan T' zugrundelag. Jede künstliche Fahrt entspricht einer Folge von realen Fahrten. Im letzten Schritt des Verfahrens werden die künstlichen Einträge in den resultierenden Umlauf- und Dienstplänen aufgelöst und es entsteht eine Lösung (ein Umlauf- und dazu kompatibler Dienstplan) für den ursprünglichen Fahrplan T (siehe Abbildung 7.1).

7.2 Das (unabhängige) fahrplanbasierte Dienstplanungsproblem

Wie der Name schon sagt, liegt dem fahrplanbasierten Dienstplanungsproblem nicht der Umlaufplan, sondern der Fahrplan zugrunde. Das bedeutet, dass die Fahrzeugumläufe noch nicht festgelegt sind und sich somit viel mehr planerische Freiheitsgrade bei der Bildung von Dienststücken bzw. Diensten ergeben. Die Dienstelemente werden direkt von den Fahrgastfahrten des Fahrplans abgeleitet. Die Möglichkeit zwei Dienstelemente zu einem Dienststück aneinander zu reihen ist nur durch zeitliche und räumliche Gegebenheiten begrenzt.

Das Ziel des fahrplanbasierten Dienstplanungsproblems (ICSP) ist gleich dem des umlaufbasierten Dienstplanungsproblems (CSP), nämlich eine kostenminimale Menge von Diensten zu finden, sodass jeder Dienst und der Dienstplan insgesamt gültig sind und jedes Dienstelement in genau einem Dienst enthalten ist (siehe Unterabschnitt 2.3.2). Der Unterschied zum CSP besteht jedoch in der Menge der zu überdeckenden Dienstelemente. Während beim CSP die Dienstelemente durch das „Schneiden" der Umläufe an Ablösepunkten erzeugt werden und somit auch Leer- und Depotfahrten bzw. Warteaktivitäten abbilden können, werden sie bei dem ICSP direkt von den Fahrplanfahrten abgeleitet. Eine weitere Eigenschaft der fahrplanbasierten Dienstplanung besteht darin, dass die Menge gültiger Dienste viel größer ist, da bei ihrer Konstruktion viel mehr Freiheitsgrade zur Verfügung stehen.

Unser Lösungsverfahren für die fahrplanbasierte Dienstplanung ist dem im Unterabschnitt 5.2.2 präsentierten Ansatz für herkömmliche umlaufbasierte Dienstplanung ähnlich und basiert auf einem Column-Generation-Verfahren in Kombination mit Lagrange-Relaxation. Die mathematische Formulierung ist ein Set-Partitioning-Problem (bzw. Set-Covering-Problem).

Zur Lösung des eingeschränkten Master-Problems werden die Überdeckungsbedingungen und wenn vorhanden die globalen Bedingungen durch Lagrange-Relaxation eliminiert. Die verbleibende Lagrange-Funktion ist ein triviales Auswahlproblem ohne weitere Nebenbedingungen. Zur Lösung des Lagrange-Dual-Problems setzen wir Subgradienten-Verfahren mit einige im Unterabschnitt 5.3.3 diskutierten Modifikationen und Erweiterungen ein.

Die (sub)optimalen Lagrange-Multiplikatoren aus dem Master-Problem werden dazu benutzt, im Pricing-Schritt neue Dienste mit negativen reduzierten Kosten zu generieren. Dazu verwenden wir genauso wie bei dem umlaufbasierten CSP sowie MD-VCSP eine zweistufige Prozedur, in der zunächst eine Teilmenge zulässiger Dienststücke generiert und im zweiten Schritt daraus gültige Dienste konstruiert

werden.

Die Menge zulässiger Dienststücke wird durch Finden kürzester Wege zwischen allen Knotenpaaren in einem speziellen *Dienststück-Erzeugungsnetzwerk* bestimmt. In diesem Netzwerk wird jedes Dienstelement durch zwei Knoten und eine Kante dazwischen repräsentiert. Sind zwei Dienstelemente i und j zueinander kompatibel, d.h. j kann direkt nach i innerhalb eines Dienststückes ausgeführt werden, dann wird der Endknoten von i mit dem Startknoten von j durch eine Kante verbunden (das Netzwerk ist somit connection-basiert). Die Kosten auf Kanten sind so gesetzt, dass die Kosten eines Pfades zwischen zwei Knoten den reduzierten Kosten des dadurch abgebildeten Dienststückes entsprechen.

Im zweiten Schritt der Pricing-Phase werden aus der Menge der gefundenen Dienststücke gültige Dienste mit negativen reduzierten Kosten erstellt. Dies kann je nach Dienstart entweder durch das Aufzählen gültiger Kombinationen aus Dienststücken (siehe Unterabschnitt 5.4.2) oder durch das Lösen eines (ressourcen-)beschränkten Kürzeste-Wege-Problems auf einem speziellen Dienst-Erzeugungsnetzwerk (siehe Unterabschnitt 5.4.3) gelöst werden.

Das fahrplanbasierte Dienstplanungsproblem ist „depotneutral", d.h. das Depot selbst wird bei der Planung der Dienste nicht betrachtet. Deswegen ist die Vorgehensweise gleich, ob der vorgegebene Fahrplan ein oder mehrere Depots berücksichtigt. Dies kann dazu führen, dass für erzeugte Dienste keine gültigen kompatiblen Umläufe existieren. Einige dieser Fälle lassen sich aber durch eine Anpassung des Netzwerkmodells ausschließen, d.h. Dienste, zu denen es offensichtlich keinen Umlauf gibt, werden nicht erzeugt. Ein Beispiel dafür ist das Entfernen einer Verbindungskante zwischen zwei Dienstelementen in dem Dienststück-Erzeugungsnetzwerk, wenn zwischen den dazugehörigen Fahrgastfahrten eine Fahrt über ein Depot für alle Depots günstiger als das Warten an der Anschlusshaltestelle ist. Somit wird verhindert, dass so ein Dienststück in der Pricing-Phase generiert wird. Die Lösung des fahrplanbasierten Dienstplanungsproblems kann dadurch bzgl. der Kosten etwas schlechter werden, allerdings entspricht sie mehr der Realität und stellt eine bessere untere Schranke dar.

Das Column-Generation-Verfahren terminiert, sobald keine neuen Dienste mit negativen reduzierten Kosten in der Pricing-Phase gefunden werden konnten oder ein anderes Abbruchkriterium erfüllt ist. Danach muss eine ganzzahlige Lösung gefunden werden. Sie bestimmt die Dienste, aus denen der resultierende Dienstplan gebildet wird. Dafür setzten wir ein hybrides Verfahren ein, welches eine Kombination aus der Branch-And-Bound-Methode (aus der CPLEX-Bibliothek) und einer lokalen Suchheuristik auf Basis von Simulated Annealing ist (siehe Unterabschnitt 5.2.2).

Bei dem Branch-And-Bound-Verfahren wenden wir das so genannte *Follow-On-*

Branching an. Diese Verzweigungsstrategie wurde in [Ryan and Foster, 1981] für Set-Partitioning-Probleme vorgeschlagen. Die Idee ist anstatt auf Variablen auf Verbindungen zu verzweigen. Dabei werden zwei Fahrten in einem Zweig direkt hintereinander innerhalb eines Dienstes bedient, während sie in dem anderen Zweig nicht in einem Dienst sein dürfen.

7.3 Erweiterte Fahrtenfixierung

Es ist offensichtlich, dass je mehr stabile Ketten es gibt bzw. gefunden werden, umso größer das Ausmaß der Problemreduktion ist und umso einfacher das eingeschränkte integrierte Problem zu lösen sein wird. Auf der anderen Seite wird der Suchraum für zulässige Dienste durch das Fixieren bzw. Zusammenlegen von Fahrgastfahrten verkleinert. Somit kann eine große Problemreduktion sich negativ auf die Qualität der approximativen Gesamtlösung auswirken.

In vielen Fällen steht die Optimalität der Gesamtlösung allerdings nicht im Vordergrund. Viel wichtiger ist die Möglichkeit, gegebene Umlauf- und Dienstplanungsprobleme in akzeptabler Zeit bzw. überhaupt integriert lösen zu können. Außerdem ist die Qualität einer approximativen Lösung bei der integrierten Betrachtung der beiden Planungsaufgaben üblicherweise immer noch viel besser, als das Ergebnis einer traditionellen strikt sequenziellen Vorgehensweise. Aus diesem Grund kann es besonders für große Probleme sinnvoll sein, so viele stabile Ketten wie möglich zu finden.

Die Grundlage für die Bestimmung stabiler Fahrtketten ist der Umlauf- und der Dienstplan, die separat und völlig unabhängig voneinander bestimmt werden (siehe Abbildung 7.1). Somit ist eine konkrete Lösung der unabhängigen Dienst- und Umlaufplanung dafür verantwortlich, wie viele gleiche Fahrtsequenzen die beiden Pläne haben. Allerdings wissen wir bereits, dass es bei der Umlaufplanung oft mehrere gleichwertige Lösungen geben kann (siehe Unterabschnitt 6.1.1). Würde man beispielsweise bei der Dekomposition der Flusslösung in dem Umlaufplanungsproblem die Informationen aus dem zuvor bestimmten Dienstplan verwenden, so könnte man die Dekompositionsstrategie auswählen, die im späteren Schritt zu mehr stabilen Ketten führt. Alternativ dazu ist auch die umgekehrte Variante möglich, wo bei der fahrplanbasierten Dienstplanung die Kenntnis über die Struktur der zuerst erzeugten Umläufe berücksichtigt wird.

Bei der erweiterten Fixierung verfolgen wir die zweite Variante. Zunächst wird das Umlaufplanungsproblem gelöst. Anschließend modifizieren wir die Kostenfunktion für die Dienstplanung wie folgt: Werden zwei Fahrgastfahrten i und j in dem vorliegenden Umlaufplan von einem Fahrzeug direkt nacheinander ausgeführt, wer-

den die Kosten c_{ij} auf der entsprechenden Verbindungskante zwischen Fahren i und j im Dienststück-Erzeugungsnetzwerk um einen Bonuswert Δ verringert. Somit wird diese Kante bei der Bildung der Dienststücke attraktiver. Bei einem nur sehr geringen Δ wird in erster Linie der optimale Dienstplan gesucht, wobei bei mehreren Entscheidungsalternativen eher diejenige gewählt wird, die die begünstigten Verbindungen beinhaltet. Steht allerdings eine möglichst große Problemreduktion im Vordergrund, kann der Bonuswert Δ höher gesetzt werden, was aber zu einer Abweichung vom Optimum bzgl. den realen Kosten führen kann.

Die vorgestellte Erweiterung der Fahrtenfixierung erlaubt eine flexible Steuerung der Problemreduktion, die je nach Zielen bzw. Anforderungen in einem konkreten Fall angepasst werden kann. Die passende Wahl von Δ ist im Allgemeinen ein Kompromiss zwischen der Lösungszeit des gesamten Fix-and-Optimize-Verfahrens und der Lösungsqualität.

7.4 Numerische Ergebnisse

In diesem Abschnitt werden die numerischen Ergebnisse zum vorgestellten Fix-and-Optimize-Verfahren für integrierte Umlauf- und Dienstplanungsprobleme diskutiert. Es wird untersucht, wie die Lösungszeit und die Qualität der Lösung sich durch den Einsatz des FaO-Verfahrens im Vergleich zum reinen integrierten Lösungsansatz für die Umlauf- und Dienstplanung aus Kapitel 5 ändern. Dabei betrachten wird sowohl die Basisvariante des vorgeschlagenen Lösungsverfahrens als auch seine modifizierte Version mit der erweiterten Fahrtenfixierung.

Die Menge der Testfälle besteht aus drei Klassen mit je 5 Testinstanzen, die sich aus den künstlich generierten ECOPT-Instanzen von Dennis Huisman zusammensetzen (siehe A.2). Dabei wurden Probleminstanzen mit vier Depots ausgewählt, die je nach Klasse 80, 160 bzw. 320 Fahrgastfahrten beinhalten. Ähnlich zu den Tests in den vorangegangenen Kapiteln lassen wir fünf Dienstarten zu: Teildienst, Frühdienst, Tagesdienst, Spätdienst und geteilter Dienst. Die genaue Spezifikation und Eigenschaften dieser Dienstarten sind im Anhang A.1 dieser Arbeit zu finden. Außerdem gelten die gleichen Annahmen wie in Kapitel 5 (siehe Seite 70). Die Kostenfunktion ist wie folgt definiert: fixe Kosten von 1000 Kosteneinheiten für jeden Dienst und jeden Umlauf und variable Kosten von einer Kosteneinheit für jede Minute, die ein Fahrzeug außerhalb des eigenen Depots verbringt. Beim Lösen der (unabhängigen) fahrplanbasierten Dienstplanung als Unterproblem im Verfahren mit erweiterter Fahrtenfixierung wird zu den fixen Dienstkosten ein variabler Bonuswert hinzuaddiert. Dieser Wert ist zu der Anzahl der Fahrtpaare proportional, die sowohl in dem entsprechenden Dienst als auch in dem bereits im Vorfeld generierten Umlaufplan direkt nacheinander ausgeführt werden. Dadurch wird eine

höhere Übereinstimmung von unabhängigen Dienst- und Umlaufplänen und somit eine höhere Reduktion des modifizierten Fahrplans angestrebt.

Das Mehrdepot-Umlaufplanungsproblem wird mit dem MDVSP-Solver gelöst, der am Lehrstuhl DS&OR Lab Universität Paderborn entwickelt wurde (siehe [Kliewer, 2005, Kliewer et al., 2006]). Dabei wird zur Lösung der MIP-Formulierung die Optimierungsbibliothek CPLEX (Version 9.1.3 mit Standardeinstellungen) eingesetzt.

Zur Lösung des (unabhängigen) fahrplanbasierten Dienstplanungsproblems wurde ein ICSP-Solver eingesetzt, dem der im Abschnitt 7.2 beschriebene Lösungsansatz zugrunde liegt. Er wurde in der Programmiersprache C# implementiert und mit .NET Framework der Version 2.0 unter Windows XP kompiliert.

Die resultierenden Umlauf- und Dienstpläne werden zur Identifizierung stabiler Fahrtsequenzen verwendet. Der ursprüngliche Fahrplan wird modifiziert, indem Fahrten jeder stabilen Fahrtkette jeweils zu einer künstlichen Fahrgastfahrt zusammengefasst werden. Für den resultierenden reduzierten Fahrplan wird das integrierte Umlauf- und Dienstplanungsproblem mit Hilfe des Verfahrens aus Kapitel 5 gelöst. Dabei wird folgende Variante des integrierten Verfahrens eingesetzt:

- zum Lösen des Master-Problems wird die Variante **LR1+S** verwendet, d.h. Lagrange-Relaxation I und das Subgradienten-Verfahren (siehe Unterabschnitt 5.8.1),
- bei der Berechnung der zulässigen Lösung wird die Strategie $\mathbf{UB^{LP+IP}(2)}$ angewandt, d.h. sowohl in der LP-Phase (am Ende jeder Column-Generation-Iteration) als auch in der IP-Phase (jede zweite Iteration des Subgradienten Verfahrens) wird zu dem aktuellen Umlaufplan ein kompatibler Dienstplan berechnet (siehe Unterabschnitt 5.8.2).

Alle in diesem Unterabschnitt dargestellten Berechnungen wurden auf einem Dell OptiPlex GX620 Personalcomputer mit einem Pentium IV 3,4 GHz Prozessor und 2 GB RAM ausgeführt.

In Tabelle 7.1 sind die Ergebnisse der Vergleichsläufe zusammengefasst. Die Tabelle ist horizontal in drei Blöcke unterteilt, ein Block für jede Instanzklasse. In jedem Block sind die Kennziffern für fünf Lösungsverfahren angegeben. Dabei steht VCSP für eine direkte Anwendung des integrierten Verfahrens aus Kapitel 5 ohne Fahrplanreduktion (als Referenz), VCSP/FaO für das hier vorgeschlagene Fix-and-Optimize-Verfahren in seiner Basisversion und VCSP/$\text{FaO}_{\Delta=n}$ für die erweiterte Fahrtenfixierung während des FaO-Verfahrens, wobei hier drei unterschiedliche Bonuswerte Δ untersucht werden (siehe Abschnitt 7.3). Für die gegenübergestellten Verfahren werden folgende Kennzahlen verglichen:

Lösungs-	reduz. Fahrplan		Laufzeit (Min.)				Lösung	
verfahren	Fahrten	Reduktion	t_{MDVSP}	t_{ICSP}	t_{VCSP}	t_{gesamt}	Fzg.	Dienste
80 Fahrten, 4 Depots								
VCSP	80	0,00 %	—	—	16,5	16,5	8,6	18,2
VCSP/FaO	63,4	20,75 %	0,1	3,5	10,6	14,2	8,6	18,8
VCSP/FaO$_{\Delta=1}$	60,6	24,25 %	0,1	3,2	9,7	13,0	8,6	18,8
VCSP/FaO$_{\Delta=5}$	60,2	24,75 %	0,1	3,6	9,4	13,1	8,6	19,0
VCSP/FaO$_{\Delta=10}$	59,8	25,25 %	0,1	4,0	9,5	13,6	8,6	19,0
160 Fahrten, 4 Depots								
VCSP	160	0,00 %	—	—	46,6	46,6	13,4	29,4
VCSP/FaO	115,6	27,75 %	0,1	13,1	26,2	39,4	13,4	29,8
VCSP/FaO$_{\Delta=1}$	111,2	30,50 %	0,1	12,6	22,5	35,2	13,4	29,8
VCSP/FaO$_{\Delta=5}$	110,6	30,88 %	0,1	15,4	21,7	37,2	13,4	29,8
VCSP/FaO$_{\Delta=10}$	110,0	31,25 %	0,1	13,3	20,3	33,7	13,4	30,2
320 Fahrten, 4 Depots								
VCSP	320	0,00 %	—	—	191,9	191,9	23,2	50,6
VCSP/FaO	219,8	31,31 %	0,2	24,7	110,5	135,4	23,2	50,4
VCSP/FaO$_{\Delta=1}$	212,2	33,69 %	0,2	32,9	105,3	138,4	23,2	50,8
VCSP/FaO$_{\Delta=5}$	211,8	33,81 %	0,2	22,1	108,0	130,3	23,2	51,4
VCSP/FaO$_{\Delta=10}$	211,0	34,06 %	0,2	26,8	103,7	130,7	23,2	51,2

Tabelle 7.1: VCSP ohne und mit dem Fix-and-Optimize-Verfahren.

- Größe des reduzierten Fahrplans (Anzahl der Fahrten und das prozentuale Ausmaß der Reduktion),
- beanspruchte Laufzeit in Minuten (unterteilt in Laufzeit für das Lösen der beiden Unterprobleme MDVSP und ICSP sowie des integrierten Problems für den ursprünglichen bzw. reduzierten Fahrplan und schließlich die Gesamtlaufzeit),
- Qualität der Lösung (Anzahl der Fahrzeuge und Dienste in den resultierenden Umlauf- und Dienstplänen).

Wie der Tabelle zu entnehmen ist, weisen die Pläne der unabhängig gelösten Umlauf- und Dienstplanungsprobleme tatsächlich eine hohe Übereinstimmung auf. Durch das Fixieren von identifizierten „stabilen" Fahrtsequenzen konnte die Größe des Fahrplans bis zu 34% im Vergleich zum Original reduziert werde, wobei das Ausmaß der Reduktion mit der Problemgröße steigt. Diese starke Problemverkleinerung führte zu einer überproportionalen Beschleunigung beim Lösen des integrierten Umlauf- und Dienstplanungsproblems für den modifizierten Fahrplan. So benötigte man für den bis auf 211 Fahrten reduzierten Fahrplan nur 103 Minuten gegenüber der 190 Minuten bei dem Original-Fahrplan mit 320 Fahrten.

Allerdings wird ein Teil der ersparten Laufzeit durch den Zusatzaufwand fürs Lösen der unabhängigen Unterprobleme wieder wettgemacht. Während die Lösungszeit für MDVCSP fast zu vernachlässigen ist, trägt sie beim Lösen des ICSP etwa mit einem Viertel zu der Gesamtlaufzeit bei. Nichtsdestotrotz braucht das vorgeschlagene Fix-And-Optimize-Verfahren insgesamt immer noch bis zu 30% weniger Lösungszeit im Vergleich zum direkten VCSP-Ansatz, um integrierte Umlauf- und Dienstplanungsprobleme zu lösen.

Neben der Laufzeit ist auch die Lösungsqualität des neuen Verfahrens zu untersuchen. Wie die Testläufe zeigen, wird die Lösungsqualität beim Einsatz des FaO-Verfahrens in seiner Basisversion nicht signifikant schlechter als die, die mit dem direkten VCSP-Ansatz erzielt wurde. Bei der größten zu untersuchenden Problemklasse konnte die Lösung durch den Ansatz des neuen Verfahrens sogar leicht verbessert werden. Diese Tatsache ist damit zu erklären, dass der VCSP-Ansatz kein exaktes Verfahren ist, sondern zahlreiche Techniken beinhaltet, um die Lösungszeit zu beschleunigen, z.B. Zeit- und Iterationslimits sowie heuristische Ansätze zum Lösen des CSP als Unterproblem. Zwar werden diese Limits an die Problemgröße angepasst, dennoch gilt im Allgemeinen, dass die erzielte Lösung mit steigender Problemgröße an Qualität verlieren kann. Da beim Anwenden der Fix-and-Optimize-Vorgehensweise der mit VCSP-Ansatz zu lösende, modifizierte Fahrplan viel kleiner ist, kann er mit etwas besserer Genauigkeit gelöst werden.

Eine Erweiterung der Basisversion des neuen Fix-and-Optimize-Verfahrens ist die erweiterte Fahrtenfixierung. Dabei werden beim Lösen des ICSP solche Sequenzen zusätzlich belohnt, die im bereits errechneten Umlaufplan vorhanden sind. Diese Variante führt zwar auf einer Seite zu mehr stabilen Fahrtketten bzw. zu einer weiteren Reduktion des modifizierten Fahrplans, auf der anderen Seite leidet aber auch die Qualität der Lösung darunter. Die Laufzeiteinsparung steht in keinem Verhältnis zu dem in Kauf zu nehmenden Verlust der Lösungsqualität. Daher betrachten wir im Weiteren der Arbeit nur die Basisversion des FaO-Verfahrens.

7.5 Zusammenfassung

In diesem Kapitel wurde ein neues approximatives Verfahren zum Lösen großer integrierter Umlauf- und Dienstplanungsprobleme vorgestellt. Es basiert auf der Idee, das Problem zunächst heuristisch zu reduzieren und erst dann mit einem integrierten Ansatz für Umlauf- und Dienstplanung zu lösen.

Das vorgestellte Fix-And-Optimize-Verfahren nutzt die spezielle zeitliche und örtliche Struktur von Fahrplänen aus. Die Fahrgastfahrten realer Fahrpläne aus der Praxis sind nicht zufällig verteilt, sondern aufeinander abgestimmt. So existie-

ren oft Fahrten bzw. Sequenzen von Fahrten, die mit großer Wahrscheinlichkeit in der optimalen Gesamtlösung für MD-VCSP direkt nacheinander ausgeführt werden. Fixiert man solche stabilen Fahrtketten noch vor dem Lösen des integrierten Problems, wird der Suchraum verkleinert und das resultierende MD-VCSP kann schneller gelöst werden.

Um stabile Fahrtketten zu identifizieren, werden zunächst die beiden Planungsprobleme Umlauf- und Dienstplanung separat und unabhängig voneinander gelöst. Dabei entstehen ein optimaler Umlaufplan und ein optimaler Dienstplan, die zueinander nicht unbedingt kompatibel sein müssen. Im zweiten Schritt werden die resultierenden Pläne miteinander verglichen und solche Sequenzen von Fahrten identifiziert, die in beiden Plänen vorhanden sind.

Der ursprüngliche Fahrplan wird modifiziert, indem jede gefundene Fahrtkette zu einer künstlichen Fahrgastfahrt zusammengefasst wird. Auf dem reduzierten Fahrplan wird das integrierte Umlauf- und Dienstplanungsproblem gelöst. Anschließend werden die resultierenden Umlauf- und Dienstpläne zurücktransformiert, indem die künstlichen Fahrten wieder durch die Sequenz der ursprünglichen Fahrgastfahrten ersetzt werden.

Wie die durchgeführten Tests zeigen, konnten die ursprünglichen Fahrpläne durch die vorgeschlagene Reduktionsstrategie bis zu 34% verkleinert werden. Dabei konnten die resultierenden reduzierten Fahrpläne bis zu 50% schneller als die Originale integriert gelöst werden. Allerdings wurde ein Teil der eingesparten Laufzeit durch den zusätzlichen Aufwand für das Lösen der beiden unabhängigen Unterprobleme (insbesondere des fahrplanbasierten Dienstplanungsproblems) wieder wettgemacht. Unter dem Strich war das vorgeschlagene FaO-Verfahren ca. 30% schneller als ein direkter VCSP-Ansatz, wobei die Lösungsqualität bei kleineren und mittleren Instanzen nicht signifikant schlechter und bei der größten Problemklasse sogar etwas besser wurde.

Die vorgeschlagene Erweiterung des Ansatzes durch eine zusätzliche Belohnung solcher Fahrtverbindungen, die bereits in einem Umlaufplan vorhanden sind, brachte keine weitere Verbesserung. Zwar konnte damit das Ausmaß der Reduktion weiter gesteigert werden, aber gleichzeitig führte diese Strategie zu einer inakzeptablen Verschlechterung der Lösungsqualität.

Die durchgeführten Tests zeigen, dass das vorgestellte mehrstufige FaO-Verfahren eine sinnvolle Erweiterung des im Kapitel 5 entwickelten Lösungsansatzes für die integrierten Umlauf- und Dienstplanungsprobleme darstellt. Insbesondere bei großen Probleminstanzen zeigt es hervorragende Ergebnisse sowohl im Hinblick auf die Laufzeit als auch auf die Lösungsqualität.

Kapitel 8

Nummerische Ergebnisse und Vergleich der Lösungsansätze

In diesem Kapitel werden die durchgeführten Tests zu den im Rahmen dieser Arbeit vorgestellten Lösungsverfahren zusammengefasst. Außerdem wird untersucht, wie gut sie miteinander kombinierbar sind. Laufzeiten und Lösungsqualität aller Lösungsverfahren und ihrer Kombinationen werden miteinander und mit den aus der Literatur bekannten Ergebnissen verglichen.

Wir untersuchen folgende Lösungsverfahren zur Lösung der Umlauf- und Dienstplanungsprobleme:

CSP: rein sequenzielle Umlauf- und Dienstplanung (siehe Unterabschnitt 5.2.2)

aCSP: adaptive Teilintegration von Umlauf- und Dienstplanung (siehe Kapitel 6)

VCSP: vollständig integrierte Umlauf- und Dienstplanung (siehe Kapitel 5)

FaO: Fix-and-Optimize-Verfahren (siehe Kapitel 7)

VCSP+aCSP: Kombination aus VCSP und aCSP. Dabei wird aCSP zur Lösung der Unterprobleme im VCSP eingesetzt (siehe Abschnitt 6.5)

FaO+aCSP: Kombination aus FaO und aCSP. Dabei wird aCSP zur Lösung der Unterprobleme im VCSP für das durch Fahrtenfixierung reduzierte Problem eingesetzt.

Bei jedem Verfahren wird jeweils die Variante benutzt, die hinsichtlich der verwendeten Formulierung, Lösungstechnik und Parameterwahl bei den entsprechenden Untersuchungen in Abschnitten 5.8, 6.6 und 7.4 am besten abgeschnitten hat.

Sowohl für die Ansätze VCSP, FaO als auch ihre Kombinationen mit aCSP untersuchen wir jeweils zwei Varianten der Verfahren, die unterschiedliche Zielsetzungen haben. Bei der ersten Variante steht die Schnelligkeit (S) und bei der zweiten die Lösungsqualität (Q) im Fokus. Neben unterschiedlichen Zeitlimits für das Lösen der Unterprobleme unterscheiden sich die beiden Varianten in der Strategie zur Berechnung einer ganzzahligen Lösung in der IP-Phase des VCSP-Ansatzes. Während bei der ersten Variante dieser Vorgang nach allen 50 Iterationen des Subgradienten-Verfahrens eingesetzt wird (Strategie $\mathrm{UB}^{\mathrm{LP+IP}}(50)$, siehe Unterabschnitt 5.8.2), beträgt diese Frequenz bei der zweiten Variante alle 2 Iterationen (Strategie $\mathrm{UB}^{\mathrm{LP+IP}}(2)$, siehe Unterabschnitt 5.8.2).

Alle entwickelten Lösungsverfahren werden auf einer Menge künstlich generierter und realer Testinstanzen getestet. Eine detaillierte Beschreibung dieser Instanzen befindet sich in Anhang A. Alle dargestellten Testergebnisse wurden auf einem Dell OptiPlex GX620 PC mit einem Pentium IV 3,4 GHz/2GB erzielt.

Künstlich generierte Instanzen

Tabellen 8.1 bzw. 8.2 fassen die Testläufe für ECOPT-Instanzen mit 2 bzw. 4 Depots zusammen. Für jede Instanzklasse werden die durchschnittliche Anzahl der Umläufe und Dienste (U+D) und die durchschnittliche Laufzeit in Minuten über 10 Instanzen angegeben. Jede Zeile repräsentiert eines der oben beschriebenen Lösungsverfahren. Die beiden letzten Spalten sind Referenzergebnisse aus der Literatur. Dabei steht HFW für die Ergebnisse, die in [Huisman, 2004] und [Huisman et al., 2005] veröffentlicht wurden. Wir verzichten hier auf die Laufzeitangaben, da sie aufgrund eines leistungsschwächeren Computers mit unseren Zeitergebnissen nicht vergleichbar sind. In der Zeile BLW werden die in [Borndörfer et al., 2004] veröffentlichten Ergebnisse dargestellt. Die Autoren benutzten für die Tests einen vergleichbaren Computer (Dell Precision 650 PC mit einem Intel Dual Xeon 3.0 GHz/4GB).

Aufgrund der präsentierten Testläufen lassen sich folgende Aussagen über die unterschiedlichen Lösungsverfahren treffen:

- aCSP liefert eine bis zu 1,2% bessere Lösung als CSP, braucht aber bis zum Vierfachen mehr Laufzeit. Allerdings ist die verbrauchte Laufzeit immer noch viel kleiner als die Zeit zur Lösung des integrierten Problems.

- $\mathrm{VCSP}^{(S)}$ produziert am schnellsten eine gute integrierte Lösung mit bis zu 15% weniger Umläufe und Dienste als bei CSP.

- $\mathrm{VCSP}^{(Q)}$ verbessert zwar die Lösung von $\mathrm{VCSP}^{(S)}$, braucht aber deutlich mehr Laufzeit. Außerdem ist dieses Verfahren sowohl in Hinsicht auf die Lösungsqualität als auch bezüglich der Laufzeit $\mathrm{VCSP}^{(S)}$+aCSP unterlegen.

- FaO$^{(S)}$ kann nur bei großen Instanzen mit VCSP$^{(S)}$ konkurrieren.
- FaO$^{(Q)}$ wird von mindestens einem anderen Verfahren dominiert.
- VCSP$^{(S)}$+aCSP ist zwar langsamer als VCSP$^{(S)}$, liefert dafür aber eine deutlich bessere Lösung.
- VCSP$^{(Q)}$+aCSP ist das Verfahren mit der besten Lösungsqualität, aber auch mit der längsten Laufzeit.
- FaO$^{(S)}$+aCSP und FaO+aCSP$^{(Q)}$ werden von anderen Verfahren dominiert.

Je nach Anforderungen an die Laufzeit und Lösungsqualität stehen dem Planer somit unterschiedliche Verfahren zur Verfügung. Wenn man sie von dem schnellsten zu dem präzisesten einordnet, ergibt sich folgende Reihenfolge zur Auswahl der richtigen Methode:
CSP $\Rightarrow$ aCSP $\Rightarrow$ VCSP$^{(S)}$(evtl. FaO$^{(S)}$) $\Rightarrow$ VCSP$^{(S)}$+aCSP $\Rightarrow$ VCSP$^{(Q)}$+aCSP

Bei den Instanzen mit 2 Depots werden die Referenzergebnisse HFW und BLW schon von dem einfachsten VCSP$^{(S)}$-Verfahren sowohl hinsichtlich der Lösungsqualität als auch hinsichtlich der Laufzeit übertroffen. Bei den großen Instanzen mit 4 Depots und 320 bzw. 400 Fahrgastfahrten schafft das erst das kombinierte Verfahren VCSP$^{(Q)}$+aCSP.

Reale Instanzen aus der Praxis
Tabelle 8.3 fast die durchgeführten Testläufe für die realen Instanzen zusammen. Der Aufbau der Tabelle ist ähnlich zu 8.1 bzw. 8.2, außer der zwei Zeilen mit Referenzergebnissen. Außerdem konnten die drei größten Instanzen nur mit dem CSP- und aCSP-Ansatz gelöst werden.

Bei diesen Instanzen ist das Verhältnis zwischen den unterschiedlichen Verfahren etwas anders als bei den künstlich generierten Instanzen, wobei dies höchstwahrscheinlich auf die Größe der realen Instanzen zurückzuführen ist. Bei R386 und R653 ist die FaO-Heuristik dem einfachen VCSP-Verfahren klar überlegen. Bei R653 konnte die beste Lösung sogar mit der Kombination aller drei Lösungstechniken FaO$^{(Q)}$+aCSP erreicht werden.

Wenn man hier die zur Verfügung stehenden Verfahren von dem schnellsten zu dem präzisesten einordnet, ergibt sich folgende Reihenfolge zur Auswahl der richtigen Methode:
CSP $\Rightarrow$ aCSP $\Rightarrow$ FaO $\Rightarrow$ VCSP+aCSP (bzw. FaO+aCSP)

Wie die Ergebnisse zeigen, stellt die FaO-Heuristik mit oder ohne Kombination mit aCSP bei großen Instanzen eine gute Alternative zu dem einfachen VCSP-Ansatz dar.

	80		100		160		200		320		400	
	U+D	Zeit	U+D	Zeit	U+D	Zeit	U+D	Zeit	U+D	Zeit	U+D	Zeit
CSP	34,1	0,03	41,5	0,04	51,8	0,2	62,9	0,2	87,1	2,8	107,1	3
aCSP	33,7	0,06	41,4	0,08	51,2	0,5	62,3	0,6	86,4	8,2	106,3	12
VCSP$^{(S)}$	29,7	3,8	35,3	4,6	47,7	16,1	58,2	22,0	84,5	86	103,1	119
VCSP$^{(Q)}$	29,4	8,1	34,7	9,2	47,4	36,6	57,6	41,2	83,9	162	102,4	191
FaO$^{(S)}$	30,0	5,1	35,7	6,3	48,0	11,6	58,5	29,1	84,5	73	102,9	120
FaO$^{(Q)}$	29,7	7,9	35,0	10,6	47,6	27,8	58,2	49,6	84,2	142	102,7	202
VCSP$^{(S)}$+aCSP	28,9	5,8	34,4	5,6	46,6	37,3	56,7	50,1	82,5	183	101,1	209
VCSP$^{(Q)}$+aCSP	28,7	18,1	33,9	19,8	46,2	121	56,5	147	82,2	280	100,9	382
FaO$^{(S)}$+aCSP	29,5	5,8	34,7	8,3	47,1	31,5	57,0	50,3	82,8	155	101,2	223
FaO$^{(Q)}$+aCSP	29,4	15,2	34,2	21,5	46,7	72,1	56,9	134	82,7	212	101,1	320
Referenz HFW	29,8	–	35,6	–	48,3	–	59,1	–	85,6	–	106,1	–
Referenz BLW	30,6	5	36,3	8	48,9	17	59,2	31	84,7	118	103,1	199

Tabelle 8.1: Vergleich der Lösungsansätze für ECOPT-Instanzen mit 2 Depots

	80		100		160		200		320		400	
	U+D	Zeit	U+D	Zeit	U+D	Zeit	U+D	Zeit	U+D	Zeit	U+D	Zeit
CSP	36,6	0,03	42,9	0,04	54,7	0,1	65	0,2	89,7	1,2	111,2	4,8
aCSP	36,5	0,04	42,8	0,05	53,9	0,3	64,8	0,4	89,1	5,9	110,4	6,7
VCSP$^{(S)}$	29,4	5,8	35,2	8,6	47,7	21,7	58,6	41,4	84,8	132	103,7	203
VCSP$^{(Q)}$	29,1	10,0	34,9	14,5	47,3	31,2	57,8	58,8	83,7	194	102,8	295
FaO$^{(S)}$	30,2	6,2	35,9	9,4	48,1	21,6	58,8	37,3	84,7	113	103,5	194
FaO$^{(Q)}$	30,0	11,3	35,4	14,9	47,7	29,6	58,5	49,7	84,3	138	103,3	239
VCSP$^{(S)}$+aCSP	29,1	6,1	34,5	9,2	46,9	25,8	57,8	47,1	83,3	172	102,0	240
VCSP$^{(Q)}$+aCSP	28,6	11,6	34,2	15,9	46,6	39,3	57,2	70,2	82,4	324	101,0	394
FaO$^{(S)}$+aCSP	29,6	6,9	35,2	11,1	47,5	25,6	58	38,4	83,9	136	102,2	226
FaO$^{(Q)}$+aCSP	29,4	11,5	34,8	16,8	47,2	32,3	57,8	62,8	83,3	289	101,5	328
Referenz HFW	29,6	–	36,2	–	49,5	–	60,4	–	–	–	–	–
Referenz BLW	29,6	13	35,7	21	47,7	44	59,0	106	82,8	328	102,0	720

Tabelle 8.2: Vergleich der Lösungsansätze für ECOPT-Instanzen mit 4 Depots

	R194		R386		R653		R1296		R2047		R2633	
	U+D	Zeit	U+D	Zeit	U+D	Zeit	U+D	Zeit	U+D	Zeit	U+D	Zeit
CSP	52	0,2	92	1,4	191	94	164	136	534	6	423	270
aCSP	52	0,5	91	4,7	187	186	157	268	532	62	418	366
VCSP$^{(S)}$	48	47	88	397	183	756	–	–	–	–	–	–
VCSP$^{(Q)}$	47	61	86	513	180	890	–	–	–	–	–	–
FaO$^{(S)}$	50	38	87	324	182	593	–	–	–	–	–	–
FaO$^{(Q)}$	49	47	86	438	179	782	–	–	–	–	–	–
VCSP$^{(S)}$+aCSP	47	53	85	492	179	835	–	–	–	–	–	–
VCSP$^{(Q)}$+aCSP	46	75	84	636	179	1038	–	–	–	–	–	–
FaO$^{(S)}$+aCSP	48	40	86	404	180	695	–	–	–	–	–	–
FaO$^{(Q)}$+aCSP	48	63	86	586	178	906	–	–	–	–	–	–

Tabelle 8.3: Vergleich der Lösungsansätze für reale Instanzen aus der Praxis

Kapitel 9

Zusammenfassung und Ausblick

Zur Zeit ist die Nachfrage nach Ansätzen für einen effizienten Ressourceneinsatz seitens der Verkehrsunternehmen so groß wie nie. Dies ist einerseits auf den immer weiter steigenden Wettbewerbsdruck und andererseits auf die immer tiefer greifende Kürzung von Subventionen im ÖPNV zurückzuführen. Eine Bestandsgarantie für einzelne Verkehrsunternehmen kann es nicht geben. Um im freien Markt überleben zu können, müssen die privaten Verkehrsunternehmen ihre Kosten senken, die Produktivität steigern und das Angebot verbessern. Auf der anderen Seite erzwingt der Wettbewerb eine hohe Flexibilität und Reaktionsgeschwindigkeit, sich auf den sich ständig verändernden Rahmenbedingungen schnell einzustellen. Die angebotenen Leistungen müssen ständig überprüft und gegebenenfalls schnell angepasst werden. Die Möglichkeit, verschiedene Szenarien zu simulieren und Sensitivitätsanalysen durchzuführen, gewinnt immer mehr an Bedeutung.

Dabei spielen computergestützte Planungswerkzeuge auf Basis von Methoden aus dem Bereich des Operations Research eine entscheidende Rolle. In vielen modernen ÖPNV-Betrieben sind sie heutzutage nicht mehr wegzudenken. Mit der Weiterentwicklung von Computertechnik und mathematischer Optimierung werden solche Werkzeuge immer leistungsfähiger und bieten Verkehrsbetrieben immer mehr Unterstützung in unterschiedlichen Phasen im umfangreichen Planungsprozess. Dabei rückt ein ökonomischer und möglichst effizienter Ressourceneinsatz immer mehr in den Fokus, da die beiden Hauptressourcen eines Verkehrsbetriebs, nämlich Fahrzeuge und Personal, die größten Kostenfaktoren darstellen.

Der Einsatz von Optimierungsmethoden bei der Umlauf- und Dienstplanung wird schon seit längerer Zeit erforscht. Es existiert eine Reihe leistungsstarker Verfahren, die in kommerziellen Planungswerkzeugen integriert sind und eine hervorragende Hilfe bei der effizienten Planung von Fahrzeugen und Fahrern leisten. Allerdings verfolgen sie, ähnlich zu der manuellen Planung, eine streng sequenzielle Abarbeitung der beiden Planungsschritte: Zuerst werden die Umläufe für

Fahrzeuge geplant und darauf basierend Dienste für die Fahrer. Auf der anderen Seite ist bekannt, dass eine simultane Betrachtung der beiden Planungsschritte einen weiteren Schritt in Richtung effizienter Ressourceneinsatzplanung geht und zusätzliches Einsparpotenzial darstellt. Ferner spielt die Integration von Umlauf- und Dienstplanung bei den Betrieben im Nachbarsort- und Regionalverkehr eine wichtige Rolle, da dort die herkömmliche sequenzielle Vorgehensweise wegen der besonderen Netzstruktur nur sehr begrenzt einsetzbar ist.

Die Zielsetzung der vorliegenden Arbeit bestand in der Entwicklung und Erprobung mehrerer Lösungsverfahren zum Lösen integrierter Umlauf- und Dienstplanungsprobleme mit mehreren Depots und einem unterschiedlichen Grad der Integration. Weiterhin sollten die Einsatzgebiete aller Verfahren analysiert und gegeneinander abgegrenzt werden.

Nach einer Einführung in die aktuelle Situation im Nahverkehr in Kapitel 1 wurde in Kapitel 2 der operative Planungsprozess im ÖPNV diskutiert. Insbesondere wurde dabei auf die beiden Aufgaben der Ressourceneinsatzplanung, die Umlauf- und Dienstplanung, sowohl einzeln als auch in einem integrierten Kontext eingegangen.

Zum besseren Verständnis der im Rahmen der Arbeit diskutierten und verwendeten Modelle und Techniken wurde in Kapitel 3 die verwendeten Techniken des Operations Research beschrieben. Anschließend wurden in Kapitel 4 der Stand der Forschung im Bereich der Umlauf- und Dienstplanung sowohl bei sequenzieller als auch bei gleichzeitiger Betrachtung ausführlich diskutiert. Dabei wurden die existierenden integrierten Modelle je nach Grad der Integration klassifiziert.

In Kapitel 5 wurde einer der in dieser Arbeit zentralen Ansätze zur Lösung integrierter Umlauf- und Dienstplanung mit mehreren Depots vorgestellt. Wir präsentierten eine neue Modellierung und Formulierung des integrierten Umlauf- und Dienstplanungsproblems mit mehreren Depots. Das zugrundeliegende Netzwerkmodell wurde zum ersten Mal mit einer neuartigen Modellierungstechnik als Time-Space-Netzwerk formuliert, das dank seiner Struktur zu einer erheblichen Reduktion der Netzwerkgröße führt. Demzufolge besitzt die davon abgeleitete mathematische Formulierung des Problems wesentlich weniger Entscheidungsvariablen als die vergleichbaren Formulierungen aus der Literatur. Weiterhin konnten netzwerkbasierte Unterprobleme viel schneller und effizienter bewältigt werden.

Der Lösungsansatz basiert auf einer Kombination eines Column-Generation-Verfahrens und Lagrange-Relaxation. Dabei untersuchten wir unterschiedliche Varianten von Lagrange-Relaxationen. Zur Lösung des resultierenden Lagrange-Dual-Problems wurden zwei Lösungsalgorithmen untersucht. Der erste Ansatz war das weit verbreitete Subgradienten-Verfahren. Wir untersuchten die Eignung unterschiedlicher in der Literatur vorgestellten Modifikationen des Verfahrens sowie ei-

gene Ideen für die gegebene Problemstellung. Der zweite untersuchte Algorithmus war der Volume-Algorithmus. Allerdings zeigten die durchgeführten Tests, dass der Volume-Algorithmus sich weder bezüglich der Lösungsqualität noch Lösungszeit gegen die fortgeschrittene Version des Subgradienten-Verfahrens durchsetzen konnte.

Bei der Lösung des Pricing-Problems im Column-Generation-Verfahren wurde eine Methode präsentiert, die dank einer speziellen Datenstruktur zur Verwaltung gültiger Dienste die Lösungszeit für diese Phase drastisch reduzieren konnte. Auch die Anzahl der Column-Generation-Iterationen konnte dank eines intelligenten Spaltenmanagements enorm verringert werden. Weiterhin wurden zahlreiche Strategien zur Berechnung einer ganzzahligen Lösung untersucht, wodurch eine gute Qualität der integrierten Lösung erreicht werden konnte.

Da eine integrierte Betrachtung von Umlauf- und Dienstplanung aus Gründen der Komplexität schon für Probleme mittlerer Größe an ihre Grenze stößt, bleibt eine sequenzielle Abarbeitung der beiden Planungsphasen für viele Verkehrsbetriebe immer noch die einzige praktikable Alternative. Deshalb wurde als zweiter Schwerpunkt dieser Arbeit ein teilintegriertes Verfahren entwickelt, das in Kapitel 6 präsentiert wurde. Es stellt einen Kompromiss zwischen der sequenziellen und simultanen Planung dar. Trotz einer sequenziellen Vorgehensweise ermöglicht dieser Ansatz eine gewisse Interaktion zwischen beiden Planungsphasen. Die derartige Kopplung hilft, bei in etwa vergleichbarer Laufzeit deutlich bessere Gesamtlösungen als bei der rein sequenziellen Vorgehensweise zu finden. Somit bieten wir mit diesem Verfahren eine anspruchsvolle Alternative auch für große Probleminstanzen.

Schließlich wurde ein approximativer Ansatz zur Lösung integrierter Umlauf- und Dienstplanungsprobleme entwickelt (siehe Kapitel 7). Die Grundidee besteht darin, das Problem zunächst zu reduzieren, indem einige Entscheidungen aufgrund gewisser Erwartungen im Vorfeld heuristisch getroffen werden, und erst dann mit einem integrierten Ansatz für Umlauf- und Dienstplanung zu lösen. Diese Vorgehensweise macht es möglich, auch größere Probleme integriert zu lösen. In diesem Zusammenhang präsentierten wir eine Technik zur Erkennung von Fahrtfolgen, die mit großer Wahrscheinlichkeit in einer optimalen bzw. guten Gesamtlösung vorkommen.

Alle entwickelten Verfahren wurden ausführlich getestet, zur Eignung für unterschiedliche Problemgrößen untersucht und einander gegenüber gestellt. Eine Zusammenfassung dieser Tests wurde in Kapitel 8 beschrieben. Dort wird auch eine Empfehlung gegeben, welche Verfahren für welche Problemgrößen bzw. Planungssituationen vorzuziehen sind.

Zusammenfassend lässt sich sagen, dass mit den vorgestellten Ergebnissen die Ziele dieser Arbeit erreicht wurden: Es wurden mehrere Lösungsverfahren zum Lö-

sen integrierter Umlauf- und Dienstplanungsprobleme mit mehreren Depots und einem unterschiedlichen Grad der Integration entwickelt und erprobt. Die Einsatzgebiete aller Verfahren wurden analysiert und gegeneinander abgegrenzt. Das vorgeschlagene integrierte Verfahren konnte die zur Verfügung stehenden Probleminstanzen aus der Literatur hervorragend meistern und übertraf die bis jetzt veröffentlichen Ergebnisse. Das adaptiv teilintegrierte Verfahren bietet dem Planer eine gute Alternative zu der herkömmlichen rein sequenziellen Vorgehensweise bei der Verplanung von Umläufen und Diensten auch bei großen Problemfällen. Der vorgeschlagene heuristische Ansatz bietet insbesondere bei den größten mit dem integrierten Ansatz zu lösenden Instanzen eine Verbesserung der Laufzeit und der Qualität der Lösung. Schließlich können alle Verfahren miteinander kombiniert werden. Wie die Ergebnisse der durchgeführten Tests zeigen, bieten solche Kombinationen in vielen Fällen zusätzliche Synergieeffekte und führen zu besseren Ergebnissen als die einzelnen Verfahren.

Die drei Hauptverfahren wurden in einem einheitlichen Framework prototypisch umgesetzt. Sie können miteinander kombiniert werden und stellen somit ein leistungsstarkes Werkzeug dar, das für unterschiedliche Problemgrößen und Planungssituationen zahlreiche Alternativen bietet. Eine geeignete Modularisierung und objektorientierte Struktur des Frameworks erlauben eine einfache Erweiterung der Methoden und ihre Weiterverwendung. Sowohl die entwickelten Methoden als auch die gewonnenen Erkenntnisse dienen bereits als Grundlage für weiterführende Forschungsaktivitäten in diesem Bereich am Lehrstuhl Decision Support & Operations Research Lab der Universität Paderborn. Einige Komponenten des entwickelten Prototyps wurden bereits für Praxisprojekte eingesetzt. Weithin findet im Moment im Rahmen einer Kooperation mit einem Industriepartner eine Integration der entwickelten Algorithmen in ein kommerzielles Planungstool statt.

Die vorliegende Arbeit eröffnet eine Basis für weitere Forschungs- und Entwicklungsaktivitäten im Bereich effizienter Planung im ÖPNV. So können die entwickelten Verfahren durch Weiterentwicklung der einzelnen Algorithmen verbessert und insbesondere für große Problemfälle anwendbar gemacht werden. Forschungsbedarf besteht weiterhin in der Erweiterung der integrierte Umlauf- und Dienstplanung um weitere praxisrelevante Aspekte. Einige Beispiele dafür sind die Berücksichtigung von flexiblen Abfahrtszeiten, Anforderungen an die Linienzusammensetzung, Berücksichtigung von Fahrerwünschen, Einbeziehung von komplizierten Dienstregeln. Weiteres Interesse seitens der Verkehrsbetriebe besteht in einer robusten Planung, bei der die resultierenden Umläufe und Dienste möglichst stabil gegen eventuelle Störungen wie beispielsweise Verspätungen sein sollen. Schließlich besteht die Möglichkeit, auch weitere Planungsschritte in einem großen Modell zu integrieren.

Anhang A

Testinstanzen

Dieser Anhang bietet einen Überblick über Dienstarten und Testinstanzen, die im Rahmen dieser Arbeitet verwendeten werden.

A.1 Dienstarten

Bei allen Testinstanzen, die im Rahmen dieser Arbeit für die Validierung der untersuchten Ansätze verwendet wurden, erlaubten wir fünf grundlegende Dienstarten, die die Zulässigkeit der generierten Dienste bestimmten. Diese Dienstarten wurden in [Huisman, 2004] definiert und sowohl dort als auch in [Borndörfer et al., 2004] eingesetzt. Ihre grundlegenden Eigenschaften sind in Tabelle A.1 zusammengefasst (vgl. [Huisman, 2004, Seite 90]). Bei der Beschreibung verwenden wir die in Unterabschnitt 2.3.3 eingeführte Terminologie.

Dienstart	Teil-dienst	Früh-dienst	Tages-dienst	Spät-dienst	geteilter Dienst
frühester Dienstbeginn			8:00	13:15	
spätestes Dienstende		16:30	18:14	19:30	
Anzahl der Dienststücke	1	2	2	2	2
Min. Dienststücklänge	0:30	0:30	0:30	0:30	0:30
Max. Dienststücklänge	5:00	5:00	5:00	5:00	5:00
Min. Pausenzeit		0:45	0:45	0:45	0:45
Max. Dienstlänge		9:45	9:45	9:45	12:00
Max. Arbeitszeit		9:00	9:00	9:00	9:00

Tabelle A.1: Eigenschaften der verwendeten Dienstarten

Während ein Teildienst nur aus einem Dienststück besteht, sind bei den übrigen

vier Dienstarten - Frühdienst, Tagesdienst, Spätdienst und geteilter Dienst - genau zwei Dienststücke pro Dienst erlaubt. Bei drei Dienstarten ist ein frühester Dienstbeginn bzw. ein späteste Dienstende definiert. Die Dienststücklänge entspricht bei allen Testinstanzen der ununterbrochenen Lenkzeit. Sie muss bei allen Dienstarten in einem Intervall zwischen 30 Minuten und 5 Stunden liegen. Zwischen den Dienststücken (außer bei Teildiensten) muss eine gesetzlich vorgeschriebene Pause von mind. 45 Minuten liegen. Die maximale Arbeitszeit beträgt bei allen Dienstarten 9 Stunden, die maximale Dienstlänge 9:45 Stunden bei normalen Diensten und bis zu 12 Stunden bei geteilten Diensten. Die maximale Arbeitszeit und Dienstlänge bei Teildiensten sind durch die Begrenzung der Dienststücklänge immer erfüllt.

Jeder Dienst beginnt mit einer Vorbereitungszeit and endet mit einer Abschlusszeit (oder Nachbereitungszeit). Sie hängt von der Start- bzw. Endhaltestelle des Dienstes ab und beträgt 10 Minuten für die Vorbereitungszeit bzw. 5 Minuten für die Abschlusszeit, falls der Dienst im Depot beginnt bzw. endet. Liegt die Start- bzw. Endposition des Dienstes an einem anderen Ablösepunkt (außerhalb des Depots), betragen die beiden Zeitpauschalen jeweils 15 Minuten. In diesem Fall wird außerdem die Transferzeit vom Depot zu der Starthaltestelle bzw. von der Endhaltestelle zum Depot hinzugerechnet.

A.2 Künstlich erzeugte ECOPT-Instanzen

Zum Testen der entwickelten Lösungsansätze verwenden wir unterschiedliche Testinstanzen. Eine Menge davon sind die ECOPT[1]-Instanzen. Es handelt sich um künstlich generierte Instanzen, die von Dennis Huisman im Rahmen seiner Promotionsarbeit speziell für Umlauf- und Dienstplanungsprobleme mit mehreren Depots erzeugt worden sind (siehe [Huisman, 2004]). Eine detaillierte Beschreibung der Struktur der Instanzen sowie die verwendeten Verteilungen und Annahmen sind in [Huisman, 2004] zu finden. Wir entschieden uns für diese Bibliothek, weil sie frei zugänglich ist (siehe [Huisman, 2005]) und außerdem zum Testen integrierter Ansätze für Umlauf- und Dienstplanung mit mehreren Depots bereits verwendet wurde (siehe [Huisman, 2004], [Borndörfer et al., 2004] und [Mesquita et al., 2005]). Somit können wir unsere Ergebnisse mit den bereits veröffentlichten direkt vergleichen.

Die Testbibliothek besteht aus 120 Instanzen vom Typ A, die je nach Anzahl der Depots und Fahrgastfahrten in 12 Problemklassen mit je 10 Instanzen pro Klasse wie folgt gruppiert sind:

- 6 Problemklassen mit 2 Depots und jeweils 80, 100, 160, 200, 320 und 400

[1] Erasmus Center for Optimization in Public Transport, Rotterdam

Fahrgastfahrten pro Instanz und

- 6 Problemklassen mit 4 Depots und jeweils 80, 100, 160, 200, 320 und 400 Fahrgastfahrten pro Instanz.

Die kleineren Instanzen mit 80, 100 und 160 Fahrgastfahrten repräsentieren einen Fahrplan mit vier Buslinien und vier Endhaltestellen, während die mittelgroßen Instanzen mit 200, 320 und 400 Fahrgastfahrten einen Fahrplan mit fünf Buslinien und fünf Endhaltestellen darstellen. Alle Endhaltestellen können als Ablösepunkte benutzt werden. Jede Fahrgastfahrt darf von allen zwei bzw. vier Depots bedient werden.

A.3 Reale Instanzen aus der Praxis

Neben den künstlich generierten Instanzen benutzen wir auch reale Instanzen aus der Praxis, um die entwickelten Lösungsverfahren zu testen. Es handelt sich um 6 Instanzen unterschiedlicher Größe, die uns von unserem Kooperationspartner zur Verfügung gestellt worden sind. Die wichtigsten Kenngrößen dieser Instanzen sind in Tabelle A.2 abgebildet. Dort ist neben der Anzahl der Fahrgastfahrten und Depots auch die minimale Anzahl der Fahrzeuge angegeben, die sich aus der Lösung des Umlaufplanungsproblems für diese Instanzen ergibt.

	R194	R386	R653	R1296	R2047	R2633
Depots	4	4	4	2	2	3
Fahrgastfahrten	194	386	653	1296	2047	2633
Fahrzeuge	19	32	67	46	114	126

Tabelle A.2: Eigenschaften der realen Testinstanzen aus der Praxis

Literaturverzeichnis

[Assad et al., 1983] Assad, A., Ball, M., Bodin, L., and Golden, B. (1983). Routing and scheduling of vehicles and crews. *Computers & Operations Research*, 10(2):63–211.

[Bahiense et al., 2002] Bahiense, L., Maculan, N., and Sagastizábal, C. (2002). The volume algorithm revisited: Relation with bundle methods. *Mathematical Programming*, 94:41–69.

[Ball et al., 1983] Ball, M., Bodin, L., and Dial, R. (1983). A matching based heuristic for scheduling mass transit crews and vehicles. *Transportation Science*, 17(1):4–31.

[Banihashemi and Haghani, 2001] Banihashemi, M. and Haghani, A. (2001). A new model for the mass transit crew scheduling problem. In Voß, S. and Daduna, J. R., editors, *Computer-Aided Scheduling of Public Transport*, Lecture Notes in Economics and Mathematical Systems, pages 1–16. Springer Verlag, Berlin.

[Barahona and Anbil, 2000] Barahona, F. and Anbil, R. (2000). The volume algorithm: Producing primal solutions with a subgradient method. *Mathematical Programming*, 87:385–399.

[Barahona and Anbil, 2002] Barahona, F. and Anbil, R. (2002). On some difficult linear programs coming from set partitioning. *Discrete Applied Mathematics*, 118:3–11.

[Barnhart et al., 2003] Barnhart, C., Cohn, A. M., Johnson, E. L., Klabjan, D., Nemhauser, G. L., and Vance, P. H. (2003). Airline crew scheduling. In Hall, R. W., editor, *Handbook of Transportation Science*, pages 517–560. Kluwer Academic Publishers, Norwell, MA, 2nd edition.

[Barnhart et al., 1998] Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. (1998). Branch-and-Price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329.

[Beasley, 1993] Beasley, J. E. (1993). Lagrangean relaxation. In Reeves, C. R., editor, *Modern Heuristic Techniques for Combinatorial Problems*, pages 243–303. Blackwell Scientific Publications.

[Beasley and Cao, 1998] Beasley, J. E. and Cao, B. (1998). A dynamic programming based algorithm for the crew scheduling problem. *Computers & Operations Research*, 25:567–582.

[Beasley and Chu, 1996] Beasley, J. E. and Chu, P. C. (1996). A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94:392–404.

[Ben Amor et al., 2004] Ben Amor, H., Desrosiers, J., and Frangioni, A. (2004). Stabilization in column generation. Les Cahiers du Gerad G-2004-75, Gerad - Universitè de Montreal.

[Bertossi et al., 1987] Bertossi, A. A., Carraresi, P., and Gallo, G. (1987). On some matching problems arising in vehicle scheduling models. *Networks*, 17:271–281.

[Bertsekas and Castañon, 1992] Bertsekas, D. P. and Castañon, D. A. (1992). A forward/reverse auction algorithm for assymmetric assignment problems. *Computational Optimization and Applications*, 1:277–297.

[Bianco et al., 1994] Bianco, L., Mingozzi, A., and Ricciardelli, S. (1994). A set partitioning approach to the multiple-depot vehicle scheduling problem. *Optimization Methods and Software*, 3:163–194.

[BMVBS, 2000] BMVBS (2000). Eckpunkte für einen leistungsfähigen und attraktiven öffentlichen personennahverkehr. Online-Publikation des Bundesministerium für Verkehr, Bau und Stadtentwicklung, http://www.bmvbs.de/-,1491.1805/Eckpunkte-fuer-einen-leistungs.htm.

[Bodin and Golden, 1981] Bodin, L. and Golden, B. (1981). Classification in vehicle routing and scheduling. *Networks*, 11:97–108.

[Bodin et al., 1983] Bodin, L., Golden, B., Assad, A., and Ball, M. (1983). Routing and scheduling of vehicles and crews: The state of the art. *Computers and Operations Research*, 10:63–211.

[Borndörfer et al., 2001] Borndörfer, R., Grötschel, M., and Löbel, A. (2001). Scheduling duties by adaptive column generation. Technical Report ZIB-Report 01-02, ZIB - Zuse Institute Berlin, Berlin.

[Borndörfer et al., 2003] Borndörfer, R., Grötschel, M., and Löbel, A. (2003). Duty scheduling in public transit. In Jäger, W., editor, *Mathematics - Key Technology for the Future*, pages 653–674. Springer Verlag.

[Borndörfer et al., 2002] Borndörfer, R., Löbel, A., and Weider, S. (2002). Integrierte Umlauf- und Diensplanung im nahverkehr. Technical Report ZIB-Report 02-10, ZIB - Zuse Institute Berlin, Berlin.

[Borndörfer et al., 2004] Borndörfer, R., Löbel, A., and Weider, S. (2004). A bundle method for integrated multi-depot vehicle and duty scheduling in public transit. Technical Report ZIB-Report 04-14, ZIB - Zuse Institute Berlin, Berlin.

[Borndörfer et al., 2005] Borndörfer, R., Schelten, U., Schlechte, T., and Weider, S. (2005). A column generation approach to airline crew scheduling. Technical Report ZIB-Report 05-37, ZIB - Zuse Institute Berlin, Berlin.

[Branco et al., 1995] Branco, I. M., Costa, A., and Paixão, J. M. P. (1995). Vehicle scheduling problem with mutiple type of vehicles and a single depot. In Daduna, J. R., Branco, I. M., and Paixão, J. M. P., editors, *Computer-Aided Transit Scheduling*, pages 115–129. Springer Verlag, Berlin.

[Camerini et al., 1975] Camerini, P. M., Fratta, L., and Maffioli, F. (1975). On improving relaxation methods by modified gradient techniques. *Mathematical Programming Study*, 3:26–34.

[Caprara et al., 1999] Caprara, A., Fischetti, M., and Toth, P. (1999). A heuristic method for the set covering problem. *Operations Research*, 47(5):730–743.

[Caprara et al., 2000] Caprara, A., Toth, P., and Fischetti, M. (2000). Algorithms for the set covering problem. *Annals of Operations Research*, 89:353–371.

[Carpaneto et al., 1989] Carpaneto, G., Dell'Amico, M., Fischetti, M., and Toth, P. (1989). A branch and bound algorithm for the multiple depot vehicle scheduling problem. *Networks*, 19:531–548.

[Carraresi and Gallo, 1984] Carraresi, P. and Gallo, G. (1984). Network models for vehicle and crew scheduling. *European Journal of Operational Research*, 16:139–151.

[Carraresi et al., 1995] Carraresi, P., Girardi, L., and Nonato, M. (1995). Network models, lagrangean relaxation and subgradients bundle approach in crew scheduling problems. In Daduna, J. R., Branco, I. M., and Paixão, J. M. P., editors, *Computer-Aided Transit Scheduling*, pages 188–212. Springer Verlag, Berlin.

[Cavique et al., 1999] Cavique, L., Rego, C., and Themido, I. (1999). Subgraph ejection chains and tabu search for the crew scheduling problem. *European Journal of Operational Research*, 50:608–616.

[Ceria et al., 1998] Ceria, S., Nobili, P., and Sassano, A. (1998). A lagrangian-based heuristic for large-scale set covering problems. *Mathematical Programming*, 81:215–228.

[Clement and Wren, 1995] Clement, R. and Wren, A. (1995). Greedy genetic algorithms, optimising mutations and bus driver scheduling. In Daduna, J. R., Branco, I. M., and Paixão, J. M. P., editors, *Computer-Aided Transit Scheduling*, pages 213–235. Springer Verlag, Berlin.

[Cohn and Barnhart, 2003] Cohn, A. and Barnhart, C. (2003). Improving crew scheduling by incorporating key maintenance routing decision. *Operations Research*, 51(3):387–396.

[Cordeau et al., 2001] Cordeau, J.-F., Stojkovic, G., Soumis, F., and Desrosiers, J. (2001). Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35(4):375–388.

[Cormen et al., 2000] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. (2000). *Introduction to Algorithms*. The MIT Press.

[CPLEX, 2003] CPLEX (2003). *CPLEX 9.0, User's Manual*. ILOG®.

[Crowder, 1976] Crowder, H. (1976). Computational improvements for subgradient optimization. In *Symposia Mathematica*, volume XIX, pages 357–372. Academic Press, London.

[Daduna and Mojsilovic, 1988] Daduna, J. R. and Mojsilovic, M. (1988). Computer-aided vehicle and duty scheduling using the HOT programme system. In Daduna, J. R. and Wren, A., editors, *Computer-Aided Transit Scheduling*, Lecture Notes in Economics and Mathematical Systems, pages 133–146. Springer Verlag, Berlin.

[Daduna and Paixão, 1995] Daduna, J. R. and Paixão, J. M. P. (1995). Vehicle scheduling for public mass transit - an overview. In Daduna, J. R., Branco, I. M., and Paixão, J. M. P., editors, *Computer-Aided Transit Scheduling*, pages 76–90. Springer Verlag, Berlin.

[Dallaire et al., 2004] Dallaire, A., Fleurent, C., and Rousseau, J. M. (2004). Dynamic constraint generation in crewopt, a column generation approach for transit crew scheduling. *To appear in: Computer-Aided Scheduling of Public Transport.*

[Dantzig and Wolfe, 1960] Dantzig, G. B. and Wolfe, P. (1960). Decomposition principles for linear programming. *Operations Research*, 8:101–111.

[Darby-Dowman et al., 1988] Darby-Dowman, K., Jachnik, J. K., Lewis, R. L., and Matra, G. (1988). Integrated decision support system for urban transport scheduling: Discussion of implementation and experience. In Daduna, J. R. and Wren, A., editors, *Computer-Aided Transit Scheduling*, Lecture Notes in Economics and Mathematical Systems, pages 226–239. Springer Verlag, Berlin.

[de Groot and Huisman, 2006] de Groot, S. W. and Huisman, D. (2006). Vehicle and crew scheduling: Solving large real-world instances with an integrated approach. Technical Report EI2004-13, Econometric Institute, Erasmus University Rotterdam.

[Dell'Amico et al., 1993] Dell'Amico, M., Fischetti, M., and Toth, P. (1993). Heuristic algorithms for the multiple depot vehicle scheduling problem. *Management Science*, 39:115–125.

[Desaulniers et al., 2001] Desaulniers, G., Desrosiers, J., and Solomon, M. M. (2001). Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems. In Ribeiro, C. C. and Hansen, P., editors, *Essays and Surveys in Metaheuristics*, pages 309–324. Kluwer, Boston.

[Desaulniers et al., 2005] Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors (2005). *Column Generation*. Springer.

[Desrochers et al., 1992] Desrochers, M., Gilbert, J., Sauvé, M., and Soumis, F. (1992). Crew-opt: Subproblem modelling in a column generation approach to urban crew scheduling. In Desrochers, M. and Rousseau, J. M., editors, *Computer-Aided Transit Scheduling*, Lecture Notes in Economics and Mathematical Systems, pages 395–406. Springer-Verlag, Berlin.

[Desrochers and Soumis, 1989] Desrochers, M. and Soumis, F. (1989). A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, 23:1–13.

[Desrosiers et al., 1995] Desrosiers, J., Dumas, Y., Solomon, M. M., and Soumis, F. (1995). Time constrained routing and scheduling. In Ball, M. O., Magnanti, T. L., Monma, C. L., and Nemhauser, G. L., editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 35–139. North-Holland, Amsterdam.

[Desrosiers and Lübbecke, 2005] Desrosiers, J. and Lübbecke, M. E. (2005). A primer in column generation. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*. Springer.

[Dowsland, 1993] Dowsland, K. A. (1993). Simulated annealing. In Reeves, C. R., editor, *Modern Heuristics Techniques for Combinatorial Problems*. Blackwell Scientific Publications.

[du Merle et al., 1999] du Merle, O., Villeneuve, D., Desrosiers, J., and Hansen, P. (1999). Stabilized column generation. *Discrete Mathematics*, 194:229–237.

[Elhallaoui et al., 2005] Elhallaoui, I., Villeneuve, D., Soumis, F., and Desaulniers, G. (2005). Dynamic aggregation of set-partitioning constraints in column generation. *Operations Research*, 53(4):623–645.

[Fachwort, 1992] Fachwort (1992). Das Fachwort im Verkehr: Grundbegriffe des ÖPNV. Alba Fachverlag, Düsseldorf.

[Fahle et al., 2002] Fahle, T., Junker, U., Karisch, S. E., Koch, N., Sellmann, M., and Vaaben, B. (2002). Constrained programming based column generation for crew assignment. *Journal of Heuristics*, 8:59–81.

[Falkner and Ryan, 1992] Falkner, J. C. and Ryan, D. M. (1992). EXPRESS: Set partitioning for bus crew scheduling in christchurch. In Desrochers, M. and Rousseau, J. M., editors, *Computer-Aided Transit Scheduling*, Lecture Notes in Economics and Mathematical Systems, pages 359–378. Springer-Verlag, Berlin.

[Fischetti et al., 2001] Fischetti, M., Lodi, A., Martello, S., and Toth, P. (2001). A polyhedral approach to simplified crew and vehicle scheduling problems. *Management Science*, 47:833–850.

[Fischetti et al., 1987] Fischetti, M., Martello, S., and Toth, P. (1987). The fixed job schedule problem with spread-time constraints. *Operations Research*, 35:849–858.

[Fischetti et al., 1989] Fischetti, M., Martello, S., and Toth, P. (1989). The fixed job schedule problem with working-time constraints. *Operations Research*, 37:395–403.

[Fischetti and Toth, 1988] Fischetti, M. and Toth, P. (1988). A new dominance procedure for combinatorial optimization problems. *Operations Research Letters*, 7(4):181–187.

[Forbes et al., 1994] Forbes, M. A., Holt, L. N., and Watts, A. M. (1994). An exact algorithm for multiple depot bus scheduling. *European Journal of Operational Research*, 72:115–124.

[Forsyth and Wren, 1997] Forsyth, P. and Wren, A. (1997). An ant system for bus driver scheduling. Research Report 97.25, University of Leeds.

[Frangioni and Manca, 2006] Frangioni, A. and Manca, A. (2006). A computational study of cost reoptimization for min cost flow problems. *Journal of Computing*, 18(1):61–70.

[Freling, 1997] Freling, R. (1997). *Models and Techniques for Integrating Vehicle and Crew scheduling*. PhD thesis, Erasmus University Rotterdam.

[Freling et al., 2001] Freling, R., Huisman, D., and Wagelmans, A. P. M. (2001). Applying an integrated approach to vehicle and crew scheduling in practice. In Voß, S. and Daduna, J. R., editors, *Computer-Aided Scheduling of Public Transport*, Lecture Notes in Economics and Mathematical Systems, pages 73–90. Springer Verlag, Berlin.

[Freling et al., 2003] Freling, R., Huisman, D., and Wagelmans, A. P. M. (2003). Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling*, 6:63–85.

[Freling et al., 1999] Freling, R., Wagelmans, A. P. M., and Paixão, J. M. P. (1999). An overview of models and techniques for integrating vehicle and crew scheduling. In Wilson, N. H. M., editor, *Computer-Aided Transit Scheduling*, pages 441–460. Springer Verlag, Berlin.

[Gaffi and Nonato, 1999] Gaffi, A. and Nonato, M. (1999). An integrated approach to extra-urban crew and vehicle scheduling. In Wilson, N. H. M., editor, *Computer-Aided Transit Scheduling*, pages 103–128. Springer Verlag, Berlin.

[Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Coputers and Intractability: a Guide to the Theory of NP-Completeness*. Freeman, San Francisco.

[Gintner et al., 2005a] Gintner, V., Kliewer, N., and Suhl, L. (2005a). Solving large multiple-depot multiple-vehicle-type bus scheduling problems in practice. *OR Spectrum*, 27(4):507–523.

[Gintner et al., 2007] Gintner, V., Kliewer, N., and Suhl, L. (2007). A crew scheduling approach for public transit enhanced with aspects from vehicle scheduling. In n. N., editor, *Computer-Aided Scheduling of Public Transport (CASPT 2004)*, Lecture Notes in Economics and Mathematical Systems, page in press. Springer, Berlin.

[Gintner et al., 2005b] Gintner, V., Kramkowski, S., Steinzen, I., and Suhl, L. (2005b). Adaptive Dienst- und Umlaufplanung im ÖPNV. In *Operations Research Proceedings*. GOR, Bremen.

[Gintner et al., 2005c] Gintner, V., Steinzen, I., and Suhl, L. (2005c). A variable fixing heuristic for the multiple-depot integrated vehicle and crew scheduling problem. In *Proceedings of 10th EWGT Meeting*. Poznan.

[Gopalakrishnan and Johnson, 2005] Gopalakrishnan, B. and Johnson, E. L. (2005). Airline crew scheduling: State-of-the-art. *Annals of Operations Research*, 140:305–337.

[Grötschel et al., 1997] Grötschel, M., Löbel, A., and Völker, M. (1997). Optimierung des Fahrzeugumlaufs im Öffentlichen Nahverkehr. In Hoffmann, K. H., Jäger, W., Lohmann, T., and Schunck, H., editors, *Mathematik - Schlüsseltechnologie für die Zukunft*, pages 609–624. Springer Verlag.

[Grünert and Irnich, 2005] Grünert, T. and Irnich, S. (2005). *Optimierung im Transport*, volume 1. Shaker Verlag, Aachen.

[Haase et al., 2001] Haase, K., Desaulniers, G., and Desrosiers, J. (2001). Simultaneous vehicle and crew scheduling in urban mass transit systems. *Transportation Science*, 35:286–300.

[Haase and Friberg, 1999] Haase, K. and Friberg, C. (1999). An exact branch and cut algorithm for the vehicle and crew scheduling problem. In Wilson, N. H. M., editor, *Computer-Aided Transit Scheduling*, pages 63–80. Springer Verlag, Berlin.

[Hadjar et al., 2006] Hadjar, A., Marcotte, O., and Soumis, F. (2006). A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. *Operations Research*, 54(1):130–149.

[Haghani and Banihashemi, 2002] Haghani, A. and Banihashemi, M. (2002). Heuristic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints. *Transportation Research A*, 36:309–333.

[Hane et al., 1995] Hane, C., Barnhart, C., Johnson, E. L., Marsten, R. E., Nemhauser, G. L., and Sigismondi, G. (1995). The fleet assignment problem: Solving a aarge integer program. *Mathematical Programming*, 70(2):211–232.

[Held and Karp, 1971] Held, M. and Karp, R. M. (1971). The traveling salesman problem and minimum spanning trees: part ii. *Mathematical Programming*, 1:6–25.

[Hoffman and Padberg, 1993] Hoffman, K. L. and Padberg, M. (1993). Solving airline crew scheduling problems by Branch-and-Cut. *Management Science*, 39:657–682.

[Holmberg and Yuan, 2000] Holmberg, K. and Yuan, D. (2000). A lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Research*, 48(3):461–481.

[Huisman, 2004] Huisman, D. (2004). *Integrated and Dynamic Vehicle and Crew Scheduling*. PhD thesis, Tinbergen Institute, Erasmus University Rotterdam.

[Huisman, 2005] Huisman, D. (2005). Random data instances for multiple-depot vehicle and crew scheduling. Online available under http://www.few.eur.nl/few/people/huisman/instances.htm.

[Huisman et al., 2005] Huisman, D., Freling, R., and Wagelmans, A. P. M. (2005). Multiple-depot integrated vehicle and crew scheduling. *Transportation Science*, 39:491–502.

[Jacobs and Brusco, 1995] Jacobs, L. W. and Brusco, M. J. (1995). Note: A local-search heuristic for large set-covering problems. *Naval Research Logistics*, 42:1129–1140.

[Jans and Degraeve, 2004] Jans, R. and Degraeve, Z. (2004). An industrial extension of the discrete lot sizing and scheduling problem. *IIE Transactions*, 36:47–58.

[Kiwiel, 1995] Kiwiel, K. C. (1995). Approximation in proximal bundle methods and decomposition of convex programs. *Journal of Optimization Theory and Applications*, 84(3):529–548.

[Klabjan, 2003] Klabjan, D. (2003). Parallel constrained shortest path. Presentation at ISMP'03, Copenhagen, Denmark.

[Klabjan et al., 2002] Klabjan, D., Johnson, E. L., Nemhauser, G. L., Gelman, E., and Ramaswamy, S. (2002). Airline crew scheduling with time windows and plane-count constraints. *Transportation Science*, 36:337–348.

[Kliewer, 2005] Kliewer, N. (2005). *Optimierung des Fahrzeugeinsatzes im öffentlichen Personennahverkehr*. PhD thesis, Universität Paderborn.

[Kliewer et al., 2006] Kliewer, N., Mellouli, T., and Suhl, L. (2006). A time-space network based exact optimization model for multi-depot bus scheduling. *European Journal of Operational Research*, 175:1616–1627.

[Kwan et al., 1999] Kwan, A. S. K., Kwan, R. S. K., and Wren, A. (1999). Driver scheduling using genetic algorithms with embedded combinatorial traits. In Wilson, N. H. M., editor, *Computer-Aided Transit Scheduling*, pages 81–102. Springer Verlag, Berlin.

[Kwan et al., 2001] Kwan, R. S. K., Kwan, A. S. K., and Wren, A. (2001). Evolutionary driver scheduling with relief chains. *Evolutionary Computation*, 9:445–460.

[Kwan and Wren, 1996] Kwan, R. S. K. and Wren, A. (1996). Hybrid genetic algorithms for bus driver scheduling. In Bianco, I. and Toth, P., editors, *Advanced Methods in Transportation Analysis*, pages 609–619. Springer Verlag.

[Lamatsch, 1988] Lamatsch, A. (1988). *Wagenumlaufplanung bei begrenzten Betriebshofkapazitäten*. PhD thesis, Universität Fridericiana zu Karlsruhe (TH).

[Larsen and Madsen, 1997] Larsen, A. and Madsen, O. B. G. (1997). Solving the multiple vehicle scheduling problem in a major scandinavian city. Technical Report IMM-REP-1997-10, Technical University of Denmark.

[Lemaréchal, 1989] Lemaréchal, C. (1989). Nondifferentiable optimization. In Nemhauser, G. L., Rinnooy Kan, A. H. G., and Todd, M. J., editors, *Optimization, Handbooks in Operations Research*, pages 529–572. North Holland, Amsterdam.

[Leuthardt, 1998] Leuthardt, H. (1998). Kostenstrukturen von Stadt-, Überland- und Reisebussen. *Der Nahverkehr*, 6:19–23.

[Li and Kwan, 2003] Li, J. and Kwan, R. S. K. (2003). A fuzzy genetic algorithm for driver scheduling. *European Journal of Operational Research*, 147:334–344.

[Li and Kwan, 2005] Li, J. and Kwan, R. S. K. (2005). A self-adjusting algorithm for driver scheduling. *Journal of Heuristics*, 11:351–367.

[Löbel, 1996] Löbel, A. (1996). Solving large-scale real-world minimum-cost flow problems by a network simplex method. Technical Report SC 96-7, Konrad-Zuse-Zentrum fur Informationstechnik Berlin (ZIB).

[Löbel, 1997] Löbel, A. (1997). *Optimal Vehicle Scheduling in Public Transit*. PhD thesis, Technische Universität Berlin.

[Löbel, 1999] Löbel, A. (1999). Solving large-scale multiple-depot vehicle scheduling problems. In Wilson, N. H. M., editor, *Computer-Aided Transit Scheduling*, pages 195–222. Springer Verlag, Berlin.

[Löbel and Strubbe, 1996] Löbel, A. and Strubbe, U. (1996). Wagenumlaufoptimierung - Methodischer Ansatz und praktische Anwendung. In *Heureka '96: Optimierung in Verkehr und Transport*, pages 341–355. Forschungsgesellschaft für Straßen- und Verkehrswesen, Köln.

[Lourenco et al., 2001] Lourenco, H. R., Paixão, J. M. P., and Portugal, R. (2001). Multiobjective metaheuristics for the bus-driver scheduling problem. *Transportation Science*, 35(3):331–343.

[Lübbecke and Desrosiers, 2005] Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53:1007–1023.

[Marchiori and Steenbeek, 2000] Marchiori, E. and Steenbeek, A. (2000). An evolutionary algorithm for large scale set covering problems with application to airline crew scheduling. In *Real World Applications of Evolutionary Computing*, pages 367–381. Springer Verlag.

[Mellouli and Kliewer, 2002] Mellouli, T. and Kliewer, N. (2002). Umlaufplanung im öffentlichen Verkehr mit mehreren Depots und Fahrzeugtypen: Neue Lösungsmodelle und praktische Aspekte. In *Heureka '02: Optimierung in Verkehr und Transport*, pages 63–75. Forschungsgesellschaft für Straßen- und Verkehrswesen, Köln.

[Mercier et al., 2005] Mercier, A., Cordeau, J.-F., and Soumis, F. (2005). A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations research*, 32:1451–1476.

[Mesquita et al., 2005] Mesquita, M., Paias, A., and Respicio, A. (2005). Branch-and-price for integrated multi-depot vehicle and crew scheduling problem. In Jaszkiewicz, A., Kaczmarek, M., Zak, J., and Kubiak, M., editors, *Advanced OR and AI Methods in Transportation*, pages 553–558, Poland. Publishing House of Poznan University of Technology.

[Mesquita and Paixão, 1992] Mesquita, M. and Paixão, J. M. P. (1992). Multiple depot vehicle scheduling problem: A new heuristic based on quasi-assigment algorithms. In Desrochers, M. and Rousseau, J. M., editors, *Computer-Aided Transit Scheduling*, Lecture Notes in Economics and Mathematical Systems, pages 167–180. Springer Verlag, Berlin.

[Mesquita and Paixão, 1999] Mesquita, M. and Paixão, J. M. P. (1999). Exact algorithms for the multiple-depot vehicle scheduling problem based on multicommodity network flow type formulations. In Wilson, N. H. M., editor, *Computer-Aided Transit Scheduling*, pages 223–246. Springer Verlag, Berlin.

[Mingozzi et al., 1999] Mingozzi, A., Boschetti, M. A., Ricciardelli, S., and Bianco, L. (1999). A set partitioning approach to the crew scheduling problem. *Operations Research*, 47:873–888.

[Paixão and Branco, 1987] Paixão, J. M. P. and Branco, I. M. (1987). A quasi-assignment algorithm for bus scheduling. *Networks*, 17:249–269.

[Patrikalakis and Xerocostas, 1992] Patrikalakis, I. and Xerocostas, D. (1992). A new decomposition scheme of the urban public transport scheduling problem. In Desrochers, M. and Rousseau, J. M., editors, *Computer-Aided Transit Scheduling*, Lecture Notes in Economics and Mathematical Systems, pages 407–425. Springer-Verlag, Berlin.

[Ribeiro and Soumis, 1994] Ribeiro, C. and Soumis, F. (1994). A column generation approach to the multiple-depot vehicle scheduling problem. *Operations Research*, 42:41–52.

[Rousseau and Blais, 1985] Rousseau, J. M. and Blais, J. Y. (1985). Hastus: An interactive system for bus and crew scheduling. In Rousseau, J. M., editor, *Computer Scheduling of Public Transport 2*, pages 45–60. North Holland, Amsterdam.

[Ryan and Foster, 1981] Ryan, D. M. and Foster, B. A. (1981). An integer programming approach to scheduling. In Wren, A., editor, *Computer Scheduling of Public Transport*, pages 269–280. North-Holland Publishing Company.

[Sandhu and Klabjan, 2005] Sandhu, R. and Klabjan, D. (2005). Integrated airline planning. Working paper (submitted for publication), Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Campaign, Urbana, IL, USA.

[Scott, 1985] Scott, D. (1985). A large linear programming approach to the public transport scheduling and cost model. In Rousseau, J. M., editor, *Computer Scheduling of Public Transport 2*, pages 473–491. North Holland, Amsterdam.

[Shen and Kwan, 2001] Shen, Y. and Kwan, R. S. K. (2001). Tabu search for driver scheduling. In Voß, S. and Daduna, J. R., editors, *Computer-Aided Scheduling of Public Transport*, Lecture Notes in Economics and Mathematical Systems, pages 121–135. Springer Verlag, Berlin.

[Silva et al., 1999] Silva, G., Wren, A., Kwan, R., and Gualda, N. (1999). An arc generation approach to solving the bus scheduling problem. Research Report Series 99.01, School of Computer Studies, University of Leeds.

[Song and Zhou, 1990] Song, T. and Zhou, L. (1990). A new algorithm for quasi-assignment problem. *Annals of Operations Research*, 24:205–223.

[Steinzen et al., 2006] Steinzen, I., Gintner, V., and Suhl, L. (2006). Network models for a decomposed pricing problem in crew scheduling. Technical report,

University of Paderborn. (Submitted for proceedings to the 10th International Conference on Computer-Aided Scheduling in Public Transport (CASPT) 2006).

[Suhl, 2000] Suhl, U. (2000). Mops - mathematical optimization system. *OR News*, 8:11–16.

[Tosini and Vercellis, 1988] Tosini, E. and Vercellis, C. (1988). An interactive system for extra-urban vehicle and crew scheduling problems. In Daduna, J. R. and Wren, A., editors, *Computer-Aided Transit Scheduling*, Lecture Notes in Economics and Mathematical Systems, pages 41–53. Springer Verlag, Berlin.

[Wedelin, 1995] Wedelin, D. (1995). An algorithm for large scale 0-1 integer programming with application to airline crew scheduling. *Annals of Operations Research*, 57:283–301.

[Wolf, 1975] Wolf, P. (1975). A method of conjucate subgradiens for minimizing nondifferentiable functions. *Mathematical Programming Study*, 3:145–173.

[Wolsey, 1998] Wolsey, L. A. (1998). *Integer Programming*. Wiley, New York.

[Wren and Wren, 1995] Wren, A. and Wren, D. O. (1995). A genetic algorithm for public transport driver scheduling. *Computers & Operations Research*, 22:101–110.

[Yunes et al., 2005] Yunes, T. H., Moura, A. V., and de Souza, C. C. (2005). Hybrid column generation approaches for urban transit crew management problems. *Transportation Science*, 39(2):273–288.