

Dynamische Tourenplanung und -steuerung

Zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften

(Dr. rer. pol.)

der Wirtschaftswissenschaftlichen Fakultät
der Universität Passau
vorgelegte Dissertation

von

Dipl. Kfm. Christoph Boomgaarden

Gutachter:

Prof. Dr. Hans Ziegler,

Prof. Dr. Peter Kleinschmidt

Tag der letzten Fachprüfung:

03. November 2007

Inhaltsverzeichnis

Symbolverzeichnis	vii
Abkürzungsverzeichnis	xv
1 Einführung	1
2 Charakterisierung von dynamisch-deterministischen Tourenplanungsproblemen (dTP)	5
2.1 Stand der Forschung zu Tourenplanungsproblemen	5
2.1.1 Lösungsansätze für statische Tourenplanungsprobleme mit Zeitfensterrestriktionen	5
2.1.2 Lösungsansätze für dynamisch-deterministische Tourenplanungsprobleme mit Zeitfensterrestriktionen	11
2.1.3 Abgrenzung zu verwandten Problemstellungen	13
2.2 Dynamisch-deterministische Tourenplanungsprobleme	15
2.2.1 Verkehrsnetzwerke zur Abbildung von dynamisch-deterministischen Tourenplanungsproblemen	15
2.2.2 Dynamische Fahrzeiten	20
2.2.2.1 Das FIFO-Prinzip	20
2.2.2.2 Der Ansatz von Ahn und Shin (1991) zur Abbildung variabler Fahrzeiten	21
2.2.2.3 Der Ansatz von Ichoua et al. (2003) zur Abbildung variabler Fahrzeiten	22
2.2.2.4 Neuer Ansatz zur Abbildung von dynamischen Fahrzeiten bei alternativen Wegen in Verkehrsnetzwerken	26
2.2.3 Zielsetzungen für dynamisch-deterministische Tourenplanungsprobleme	31
2.2.3.1 Mögliche Zielsetzungen und Zielkonflikte	31
2.2.3.2 Lexikographische Zielfunktion für dynamisch-deterministische Tourenplanungsprobleme	38

2.2.4	Formales Modell zu dynamisch-deterministischen Tourenplanungsproblemen	40
3	Charakterisierung von dynamisch-stochastischen Tourensteuerungsproblemen (dTS)	43
3.1	Störereignisse zur Abbildung von stochastischen Fahrzeiten	43
3.2	Formales Modell zu dynamisch-stochastischen Tourensteuerungsproblemen	48
3.2.1	Lexikographische Zielfunktion für dynamisch-stochastische Tourensteuerungsprobleme	48
3.2.2	Problembeschreibung (deskriptives Modell)	50
4	Ausgewählte Heuristiken für Tourenplanungs- und -steuerungsprobleme	53
4.1	Exkurs: Zulässigkeitsbedingungen nach Ahn und Shin (1991)	53
4.2	Konstruktionsverfahren für Tourenplanungsprobleme	57
4.2.1	Savings-Heuristik	57
4.2.2	Nearest-Neighbor-Heuristik	59
4.2.3	Insertion-Heuristik	62
4.3	Verbesserungsverfahren für Tourenplanungsprobleme	65
4.3.1	2-opt Nachbarschaftssuche	65
4.3.2	Cross-Exchange Nachbarschaftssuche	67
4.4	Ameisenkoloniealgorithmen für Tourenplanungsprobleme	73
4.4.1	Die ACO-Metaheuristik	73
4.4.2	Der Ansatz von Gambardella et al. (1999) für statische Tourenplanungsprobleme mit Zeitfensterrestriktionen	77
4.4.2.1	Zielsetzung und Lösungsabbildung	77
4.4.2.2	Kolonie ACS-VEI	79
4.4.2.3	Kolonie ACS-TIME	83
4.4.2.4	Pheromonupdate	84
4.4.2.5	Lokale Komponente	84
4.4.3	Der modifizierte Ansatz von Donati et al. (2003a) für dynamisch-deterministische Tourenplanungsprobleme	85
4.4.3.1	Zielsetzung	85
4.4.3.2	Pheromonmatrix	86
4.4.3.3	Lokale Komponente	86
4.4.3.4	Verbesserungsverfahren	87
4.4.3.5	Konstruktion von neuen Lösungen	88
4.4.3.6	Weitere Unterschiede zu Gambardella et al. (1999)	89
4.5	Ein neuer Ameisenkoloniealgorithmus für dTP (dynACO)	90
4.5.1	Aufbau	90

4.5.2	Initialisierung	90
4.5.3	Ameisen-Iteration	92
4.5.3.1	Konstruktionsphase	92
4.5.3.2	Verbesserungsphase	94
4.5.3.3	Pheromonupdate	94
4.6	Störungsreaktionsverfahren zu dynamisch-stochastischen Tourensteuerungsproblemen	95
4.6.1	Handlungsalternative keine Reaktion (ALT)	95
4.6.2	Anpassung der Routen im Verkehrswegenetzwerk (ROUTE)	96
4.6.3	Lokale Verbesserungsverfahren	96
4.6.3.1	Paarweiser Tausch (PT)	97
4.6.3.2	Tausch der Positionen von zwei Kundenaufträgen (EX)	99
5	Experimentelle Analyse	101
5.1	Ziel der experimentellen Analyse zu dynamisch-deterministischen Tourenplanungsproblemen (dTP)	101
5.2	Netzwerkgenerator	102
5.2.1	Eigenschaften von realen Verkehrswegenetzwerken	102
5.2.2	Netzwerkgenerator: Struktur	103
5.2.3	Netzwerkgenerator: Bewertung der Pfeile des Verkehrswegenetzwerks	108
5.2.3.1	Distanzbewertung	108
5.2.3.2	Geschwindigkeitsfunktionen für dynamisch-deterministische Tourenplanungsprobleme (dTP)	108
5.2.3.3	Geschwindigkeitsfunktionen für dynamisch-stochastische Tourensteuerungsprobleme (dTS)	112
5.3	Neue Probleminstanzen zu dynamisch-deterministischen Tourenplanungsproblemen	114
5.4	Parameter der untersuchten Verfahren	116
5.5	Ergebnisse und Interpretation der experimentellen Analyse zu dTP	120
5.5.1	Ergebnisse der experimentellen Analyse	120
5.5.1.1	Probleminstanzen A - enge Zeitfenster, geringe Fahrzeugkapazität	121
5.5.1.2	Probleminstanzen B - enge Zeitfenster, hohe Fahrzeugkapazität	126
5.5.1.3	Probleminstanzen C - weite Zeitfenster, geringe Fahrzeugkapazität	130

5.5.1.4	Probleminstanzen D - enge Zeitfenster, geringe Fahrzeugkapazität, 33% der Kundenaufträge ohne Zeitfenster	134
5.5.1.5	Probleminstanzen E - enge Zeitfenster, geringe Fahrzeugkapazität, 65 Kundenaufträge	138
5.5.1.6	Probleminstanzen F - enge Zeitfenster, hohe Fahrzeugkapazität, 65 Kundenaufträge	142
5.5.2	Parametrische Analyse:	147
5.5.2.1	Probleminstanz A3: Zusätzliche Versuche zu dynACO .	147
5.5.2.2	Probleminstanz F3: Parametrische Analyse zu dynACO	148
5.5.3	Zusammenfassung der Ergebnisse der experimentellen Analyse von dynamisch-deterministischen Tourenplanungsproblemen (dTP)	152
5.6	Experimentelle Analyse von dynamisch-stochastischen Tourensteuerungsproblemen	156
5.6.1	Probleminstanzen zur Untersuchung von dynamisch-stochastischen Tourensteuerungsproblemen (dTS) . .	156
5.6.2	Ergebnisse und Interpretation der experimentellen Analyse zu dTS	159
5.6.2.1	Übersicht Störpläne	159
5.6.2.2	Detaillerggebnisse der experimentellen Analyse	161
5.6.2.3	Ausgewählte Detaillerggebnisse der experimentellen Analyse	162
5.6.2.4	Zusammenfassung der Ergebnisse der experimentellen Analyse von dynamisch-stochastischen Tourensteuerungsproblemen (dTS)	175
6	Fazit	181
	Literaturverzeichnis	xix

Symbolverzeichnis

dynamisch-deterministisches Tourenplanungsproblem

C	Menge der entscheidungsrelevanten Kundenaufträge
$ C $	Anzahl der entscheidungsrelevanten Kundenaufträge
C_0	Menge der entscheidungsrelevanten Kundenaufträge und Depot, $C_0 = C \cup \{0\}$
c_{pos}^z	Kunde, der als <i>pos</i> -tes auf der Tour z bedient wird.
$d_{ij}(\tau_i)$	Entfernung auf dem schnellsten Weg von i nach j bei Abfahrt in i zum Zeitpunkt τ_i
d_{mn}	Entfernung zwischen den Knoten m und n des Verkehrswegenetzwerks
e_0	frühestmöglicher Zeitpunkt zu dem das Depot verlassen werden darf, Beginn des Planungszeitraums
e_i	frühestmöglicher Zeitpunkt zu dem mit der Bedienung des Kunden i begonnen werden darf
h, i, j	Bezeichner von Knoten aus der Menge C_0 (Kunden, Depot)
(i, j)	„Fahrt“, Lösungsbestandteil
l_0	spätestmöglicher Zeitpunkt zu dem das Depot verlassen werden darf
l_i	spätestmöglicher Zeitpunkt zu dem mit der Bedienung des Kunden i begonnen werden darf
M	hinreichend große Konstante
m, n	Bezeichner von Knoten aus der Menge N (Kunden, Depot, Kreuzungen)
(m, n)	Pfeil, Bestandteil des Verkehrswegenetzwerks
N	Menge aller Knoten des Verkehrswegenetzwerks

Q	Maximalkapazität eines Lieferfahrzeugs
q_i	Kapazitätsbedarf der Lieferung die an den Kunden i geht
s_i	Servicedauer: Zeitspanne, die für die Bedienung des Kunden i benötigt wird
T_z	Tour, die durch Fahrzeug z durchgeführt wird
$t_{ij}(\tau_i)$	Fahrzeit von i nach j , bei Abfahrt in i zum Zeitpunkt τ_i
$t_{ij}^{-1}(\tau_j)$	Fahrzeit von i nach j , bei Ankunft in j zum Zeitpunkt τ_j
$t_{mn}(\tau_m)$	Fahrzeit von Knoten m zu Knoten n bei Abfahrt von m zum Zeitpunkt τ_m im Verkehrswegenetzwerk
τ	Zeitpunkt
u	Kundenbezeichner
w_i	Wartezeit vor der Belieferung des Kunden i
X	Tourenplan
x_{ij}^z	Entscheidungsvariable, $x_{ij}^z = \begin{cases} 1 & \text{wenn } j \text{ direkt im Anschluss an } i \text{ mit Fahrzeug } z \text{ angesteuert wird} \\ 0 & \text{sonst} \end{cases}$
y_0^z	Entscheidungsvariable, Zeitpunkt zu dem die Tour z beendet wird
y_i	Entscheidungsvariable, Zeitpunkt zu dem begonnen wird, den Kunden i zu bedienen
Z	Menge aller Lieferfahrzeuge
z	Bezeichner von Fahrzeugen und Touren
ζ_z	Entscheidungsvariable, $\zeta_z = \begin{cases} 1 & \text{wenn Fahrzeug } z \text{ eingesetzt wird} \\ 0 & \text{sonst} \end{cases}$

dynamische Fahrzeiten

$ankunft_{mn}(\tau)$	Ankunft beim Knoten n , wenn der Knoten m zum Zeitpunkt τ verlassen wird
δ	noch zurückzulegende Entfernung zum Zielknoten

faz_ψ	frühestmöglicher Ankunftszeitpunkt beim Knoten ψ im Verkehrswegenetzwerk
$\Gamma^+(m)$	Menge der Knoten im Verkehrswegenetzwerk, die mit m direkt verbunden sind
g_{mn}^{int}	Geschwindigkeit auf der Verbindung (m, n) des Verkehrswegenetzwerks während des Intervalls int
int	Intervalls, je Intervall gilt eine konstante Geschwindigkeit
Ψ_1	Menge der Kandidaten-Knoten des Verkehrswegenetzwerks
Ψ_2	Menge der Knoten des Verkehrswegenetzwerks, zu denen der schnellste Weg bekannt ist
τ'	voraussichtlicher Ankunftszeitpunkt beim Zielknoten
τ_e^{int}	Zeitpunkt zu dem das Intervall int beginnt
τ_l^{int}	Zeitpunkt zu dem das Intervall int endet
vor_m	Vorgänger des Knoten m auf dem schnellsten Weg von i nach j im Verkehrswegenetzwerk

dynamisch-stochastisches Tourensteuerungsproblem

b_σ	Zeitpunkt, zu dem das Störereignis σ bekannt gegeben wird
Δ_σ	Multiplikator für das Störereignis σ
e_σ	Zeitpunkt, zu dem das Störereignis σ zu wirken beginnt
l_σ	Zeitpunkt, zu dem das Störereignis σ zu wirken endet
$late_i$	Binärvariable, $late_i = \begin{cases} 1 & \text{wenn der Kunde } i \text{ verspätet beliefert wird } (y_i > l_i) \\ 0 & \text{sonst} \end{cases}$
m	Knoten des Verkehrswegenetzwerks, in dem sich das Fahrzeug zum frühestmöglichen Reaktionszeitpunkt τ_m befindet
(m_σ, n_σ)	Durch das Störereignis σ betroffener Pfeil des Verkehrswegenetzwerks
S	Störplan, Menge von Störereignissen
σ	Bezeichner eines Störereignisses

τ_m	Frühestmöglicher Reaktionszeitpunkt
τ_{reakt}	Reaktionszeitpunkt, Zeitpunkt zu dem die Tourensteuerung durchgeführt wird
U	Menge der noch nicht bedienten Kundenaufträge
U_0	Menge der noch nicht bedienten Kundenaufträge und Depot, $U_0 = U \cup \{0\}$
X	Plan-Tourenplan, legt Zuordnung von Kunden zu Fahrzeugen und vorläufige Touren und Routen fest
X^{Ist}	Ist-Tourenplan, Lösung des dynamisch-stochastischen Tourensteuerungsproblems nach Abschluß der Tourdurchführung
x_{ij}	Entscheidungsvariable
y_0	Zeitpunkt zu dem die betrachtete Tour beendet wird

Heuristiken

$abstand_{ij}$	Verfahren des besten Nachfolgers: Abstandswert
$anz_{ameisen}$	MACS: Anzahl Ameisen je Ameisenkolonie
anz_{iter}	dynACO: Anzahl durchgeführter Ameisen-Iterationen, ohne daß eine verbesserte Lösung gefunden wurde
anz_{iter}^{max}	MACS und dynACO: maximale Anzahl an Ameisen-Iterationen, Abbruchkriterium
β	MACS: Gewichtungsfaktor für lokale Komponente
$CX_{abschnitt}^{max}$	maximale Anzahl Kunden auf dem zu verschiebenden oder zu vertauschenden Abschnitt
CX_{iter}^{max}	dynACO: maximale Anzahl Iterationen Cross-Exchange je Ameisen-Iteration
$c(i, u, j)$	Einfügeheuristik: Einfügekosten, wenn Kunde u zwischen i und j eingefügt wird
$[e_V; l_V]$	Abschnittszeitfenster
η_{ij}	MACS: lokale Komponente, statisches Problem

$\eta_{ij}(\tau)$	MACS und dynACO: lokale Komponente, dynamisch-deterministisches Problem
faz_i	frühestmöglicher Ankunftszeitpunkt beim Kunden i auf der aktuellen Tour
saz_i	spätestmöglicher Ankunftszeitpunkt beim Kunden i auf der aktuellen Tour
γ	Vorteilhaftigkeitsparameter zur Beurteilung, ob Verbesserungsverfahren durchgeführt werden soll
IN_j	MACS: Bestandteil der lokalen Komponente der Kolonie ACS-VEI, die angibt seit wievielen Iterationen der Knoten j nicht mehr einer Tour zugeordnet werden konnte
κ_{1-4}	Gewichtungsfaktoren
λ_{1-4}	Gewichtungsfaktoren für die lokale Komponente bei dynACO
p_j	MACS und dynACO: Auswahlwahrscheinlichkeit für den Kunden oder das Depot j
π	dynACO: Gewichtungsfaktor für die lokale Komponente
ϕ_{init}	MACS: Initialisierungs-Pheromonwert
ϕ_{ij}	MACS und dynACO: Pheromonwert für die Fahrt (i, j)
ϕ_{ij}^{extra}	dynACO: zusätzlich abzulegendes Pheromon
ϕ_{min}	dynACO: Mindestpheromonwert
ϕ_{max}	dynACO: Höchstpheromonwert
q_0	MACS: Ausbeutungsparameter
ρ	MACS: Verdunstungsrate für das Pheromon, dynACO: Pheromonerhaltungsrate
sav_{ij}	Savings-Verfahren: Savingswert
θ	MACS: zeitliches Intervall für mehrperiodige Pheromonmatrix
T_z^{neu}	Tourbezeichner für ggf. unzulässige Tour (MACS)
$ T_z^{neu} $	Anzahl Knoten, die auf der Tour T_z^{neu} besucht werden
U	Menge der unzugeordneten Kunden
U'	Menge der aktuell zulässigen Nachfolgerknoten

V	Tour in Vektordarstellung, $V = (m, v_{[1]}, \dots, v_{[U]}, 0)$
$X^{aktuell}$	Cross-Exchange: Ausgangslösung
$X^{aktuell}$	MACS und dynACO: aktuelle Lösung
X^{best}	Cross-Exchange: beste bekannte Nachbarlösung
X^{best}	MACS und dynACO: beste bekannte Lösung
$X^{best'}$	MACS: unzulässiger Tourenplan, mit maximaler Anzahl zugeordneter Kundenaufträge bei vorgegebener Tourenzahl
X^{init}	Ausgangslösung
X^{neu}	Cross-Exchange: zu untersuchende Nachbarlösung
\mathcal{Z}	MACS: Anzahl Duplikate des Depots

Lösungen

$anz(X)$	Anzahl Touren des Tourenplans X
$dist(X)$	(Plan-)Gesamtentfernung des Tourenplans X
$sched(X)$	(Plan-)Gesamtbediendauer des Tourenplans X
$sched(\bar{X})$	mittlere Plan-Gesamtbediendauer
$sched(X^{Ist})$	mittlere Ist-Gesamtbediendauer
$travel(X)$	(Plan-)Gesamtfahrzeit des Tourenplans X

Netzwerkgenerator

$anz_{zugeordnet}$	Zählvariable für den Netzwerkgenerator
d_{mn}^W	Länge des kürzesten Wegs zwischen den Knoten m und n im Graphen W
d_{mn}^{euklid}	euklidische Distanz zwischen den Knoten m und n
K	Kantenmenge eines Graphen
K_{max}	Anzahl Kanten, die dem Graphen höchstens hinzugefügt werden soll

$katgorie_{mn}$	Kategorie des Pfeils (m, n) im Verkehrswegenetzwerk W , tageszeitabhängiger Multiplikator
$[m, n]$	Kante eines ungerichteten Graphen
R	Teilmenge von Nachbarknoten
r	vorgegebene Richtung (Nord, Süd, Ost oder West)
r_{mn}	Richtungsindikator des Pfeils (m, n)
typ_{mn}	Typ des Pfeils (m, n) im Verkehrswegenetzwerks W , tageszeitunabhängiger Multiplikator
W	Graph / Verkehrswegenetzwerk
x_m^{koord}	X-Koordinate des Knoten m
y_m^{koord}	Y-Koordinate des Knoten m

Abkürzungsverzeichnis

ACS-VEI	Ant Colony System, Ziel: Minimierung der Anzahl der Touren
ACS-TIME	Ant Colony System, Ziel: Minimierung der Gesamtfahrzeit
ALT	Störungsreaktionsverfahren: keine Reaktion
CX	Verbesserungsverfahren: Cross-Exchange
dTP	dynamisch-deterministisches Tourenplanungsproblem mit Zeitfensterrestriktionen
dTS	dynamisch-stochastisches Tourensteuerungsproblem mit Zeitfensterrestriktionen
dynACO	Ameisenkolonialgorithmus für dynamisch-deterministische Tourenplanungsprobleme
EE	Entfernungseinheiten
EX	Störungsreaktionsverfahren: Tausch der Positionen von zwei Kundenaufträgen
GE	Geldeinheiten
INS	Einfügeheuristik
Kat.	Kategorie
MACS	Multiple Ant Colony System
Multipl.	Multiplikator
NN	Verfahren des besten Nachfolgers
PT	Störungsreaktionsverfahren: paarweiser Tausch

PZR	Planungszeitraum, entspricht dem Zeitfenster des Depots $[e_0; l_0]$
ROUTE	Störungsreaktionsverfahren: neue Routen ermitteln
VRPTW	statisch-deterministisches Tourenplanungsproblem mit Zeitfensterrestriktionen (vehicle routing problem with time windows)
ZE	Zeiteinheiten

Abbildungsverzeichnis

2.1	Verkehrswegenetzwerk	16
2.2	Touren	17
2.3	Abschnitt	18
2.4	Beispiel: Alternative Wege vom Kunden 1 zum Kunden 3	19
2.5	Routen	20
2.6	Abschnittsweise definierte Fahrzeitfunktion	21
2.7	Fahrzeitfunktion: links: Geschwindigkeitsfunktion; rechts: daraus resultierende Fahrzeitfunktion	24
2.8	Beispiel zur Ermittlung von kostenminimalen Wegen	28
2.9	Beispiel 1: Zielkonflikt zwischen Minimierung der Entfernung und Minimierung der Fahrzeit	36
2.10	Beispiel 2: links: Lösung mit minimaler Tourenzahl, rechts: Lösung mit minimaler Gesamtentfernung	37
2.11	Beispiel 3: links: Lösung mit minimaler Distanz, rechts: Lösung mit minimaler Bediendauer	38
2.12	Beispiel 3: gewichtete Zielfunktion	38
3.1	Anpassung der Route als Reaktion auf ein Störereignis	45
4.1	Zu Zulässigkeitsbedingung 3 nach Ahn und Shin (1991)	56
4.2	Cross-Exchange: Tausch von zwei Abschnitten zwischen zwei Touren	67
4.3	Or-Opt als Spezialfall von Cross-Exchange	70
4.4	Cross-Exchange innerhalb einer Tour	71
4.5	Unzulässige Cross-Exchange Operation	71
4.6	2-opt* als Spezialfall von Cross-Exchange	73
4.7	Prinzip der Ausbildung einer Ameisenstraße vom Bau zur Futterquelle	74
4.8	Aufteilung der Pheromonmatrix in mehrere Intervalle nach Donati et al. (2003a)	87
4.9	Störungsreaktionsverfahren PT	98
4.10	Störungsreaktionsverfahren EX	99
5.1	Links: reales Verkehrswegenetzwerk Passau und Umgebung; Rechts: Ergebnis des Netzwerkgenerators	103

5.2	Ermittlung des Schnittpunkts zweier Kanten	105
5.3	Verkehrswegenetzwerke: oben 50a, unten 50b	106
5.4	Verkehrswegenetzwerke: oben 65a, unten 65b	107

1 Einführung

Die Logistik befasst sich mit der Aufgabenstellung Rohstoffe, Zwischenerzeugnisse und fertige Produkte in der richtigen Qualität, zur richtigen Zeit, in der richtigen Menge und am richtigen Ort zu möglichst geringen Kosten bereitzustellen. Es wird zwischen Beschaffungslogistik, Produktionslogistik und Distributionslogistik unterschieden. Die Beschaffungslogistik betrachtet dabei den Material- und Informationsfluss auf der Lieferantenseite eines Unternehmens. Es sind die Bedarfe an Rohstoffen zunächst zu ermitteln und dann Entscheidungen über die anzuwendenden Beschaffungsprinzipien und Bestellmengen zu treffen. Die Produktionslogistik behandelt die logistischen Fragestellungen, die sich innerhalb eines Unternehmens ergeben können, insbesondere Problemstellungen, die sich mit innerbetrieblichen Standort- und Layoutentscheidungen, sowie Transport- Lager- und Kommissionierungsvorgängen befassen. Die Distributionslogistik betrachtet den Material- und Informationsfluss der Fertigwaren auf der Absatzseite, also Problemstellungen die sich mit der Verteilung von Fertigwaren vom Unternehmen bis zum Kunden befassen.

Die Tourenplanung als Teil der Distributionslogistik befasst sich mit der Planung von Transportvorgängen ausgehend von einem zentralen Auslieferungslager (im folgenden als Depot bezeichnet) zu einer Vielzahl von Kunden, mit dem Ziel eine gegebene Menge von Kundenaufträgen möglichst kostengünstig zu erfüllen. Ein Kundenauftrag ist dabei die Anlieferung einer gegebenen Menge von kundenspezifischen Gütern. Während der Auslieferung sind Kapazitätsrestriktionen der ausliefernden Fahrzeuge, Kundenterminwünsche (in Form von Zeitfensterrestriktionen) und die Öffnungszeiten des Depots zu berücksichtigen. Im Unterschied zum Traveling-Salesman-Problem (TSP) ist es in der Regel nicht möglich, alle Kunden auf einer einzigen Rundreise zu besuchen. Das Tourenplanungsproblem besteht aus einem Zuordnungsproblem und einer Menge von Rundreiseproblemen mit Zeitfensterrestriktionen, die simultan zu lösen sind.

Die Planung der Transportvorgänge zur Belieferung einer Menge von Restaurants in einer Region, mit jeweils kundenspezifischen Waren, ausgehend von einem Großmarkt ist ein typisches Tourenplanungsproblem. Jedes Restaurant wird einem Lieferfahrzeug zugeordnet und für jedes Lieferfahrzeug werden eine Belieferungsreihenfolge und

die zu wählenden Wege bestimmt. Ein ähnliches Beispiel ist die Belieferung einer Menge von Supermarktfilialen ausgehend von einem Zentrallager. In jedem Supermarkt ist das Sortiment täglich wieder aufzufüllen, aus den Verkäufen des Vortags in den einzelnen Supermärkten ergeben sich unterschiedliche Bedarfe in jedem Supermarkt. Das Tourenplanungsproblem besteht darin eine Belieferungsreihenfolge der Supermärkte festzulegen, unter Berücksichtigung der Bedarfe jedes einzelnen Supermarkts und der vorhandenen Kapazitäten der Lieferfahrzeuge. In produzierenden Unternehmen sind Tourenplanungsprobleme zu lösen, wenn beispielsweise Güter ausgehend von einem Fertigungsstandort an eine Menge von Kunden möglichst kostengünstig zu verteilen sind. Ein Kunde ist dabei nicht notwendigerweise ausserhalb des Unternehmens anzusiedeln, vielmehr kann es sich bei einem Kunden auch um regionale Vertriebszentren des gleichen Unternehmens handeln, die aus Sicht des ausliefernden Standorts wie Kunden zu behandeln sind. Ein Tourenplanungsproblem aus Sicht eines Lieferanten von Fenstern und Türen liegt beispielsweise vor, wenn die bestimmte Fenster und Türen an eine vorgegebene Menge von Baustellen auszuliefern sind. Je Lieferfahrzeug kann nur eine beschränkte Menge an Fenstern und Türen transportiert werden, jeder Kunde hat ganz bestimmte, individuell unterschiedliche Modelle bestellt, die zu einem bestimmten Zeitpunkt angeliefert werden sollten.

Üblicherweise sind die Standorte der Kunden, deren Nachfrage nach Gütern, der Zeitbedarf für den Belieferungsvorgang beim Kunden (üblicherweise die Dauer für das Abladen der Güter beim Kunden, im folgenden als Servicedauer bezeichnet) und die Zeitfenster für die Belieferung vorgegeben.¹ Ein Zeitfenster ist das zeitliche Intervall innerhalb dessen mit der Belieferung eines Kunden begonnen werden muss. Es ist beispielsweise vorstellbar, daß ein zu belieferndes Restaurant oder ein zu beliefernder Supermarkt in einer Fußgängerzone liegt und daher nur zwischen 6.00 und 9.00 Uhr morgens beliefert werden darf, während in einem anderen Restaurant oder Supermarkt erst ab 14.00 Uhr überhaupt Personal die Warenlieferung annimmt. Auch auf einer Baustelle ist ein Belieferungszeitfenster einzuhalten, innerhalb dessen die Handwerker für den Einbau der Fenster und Türen bereitstehen. Eine verspätete Anlieferung ist zu vermeiden, da dadurch die Bauarbeiten verzögert werden und wartende Handwerker auf der Baustelle trotzdem zu bezahlen sind. Umgekehrt kann bei einer zu frühen Anlieferung das Lieferfahrzeug ggf. nicht entladen werden oder den Kunden überhaupt nicht erreichen.

Das Depot und jeder Kunde wird durch einen eigenen Knoten eines Netzwerks abgebildet. Die räumlichen oder zeitlichen Entfernungen zwischen den Kunden werden durch Pfeile abgebildet. Die Entfernungen, Fahrzeiten und die Fahrtkosten sind bekannt, sie werden durch Pfeilbewertungen abgebildet. Das Tourenplanungsproblem

¹Für eine formale Beschreibung des in dieser Arbeit untersuchten Problems vgl. Abschnitt (2.2.4)

mit Zeitfensterrestriktionen (Vehicle Routing Problem with Time Windows, VRPTW) ist ein NP-schweres Problem². Ein Problem wird als NP-schwer bezeichnet, wenn kein Algorithmus zu seiner Lösung bekannt ist, dessen Rechenaufwand durch ein Polynom der Problemgröße nach oben beschränkt ist.

Es kann zwischen statischen und dynamischen sowie zwischen deterministischen und stochastischen Tourenplanungsproblemen unterschieden werden. Statische Probleme unterscheiden sich von dynamischen Problemen primär dadurch, daß alle entscheidungsrelevanten Informationen im Zeitablauf konstant sind. Deterministische Probleme sind dadurch charakterisiert, daß alle entscheidungsrelevanten Informationen und Variablenausprägungen zum Planungszeitpunkt, d.h. vor Beginn der Belieferung der Kunden, bekannt sind. Bei stochastischen Problemstellungen sind die tatsächlichen Ausprägungen von entscheidungsrelevanten Informationen teilweise erst nach dem Abschluss der Planung verfügbar.

In dieser Arbeit werden heuristische Lösungsansätze zu dynamisch-deterministischen Tourenplanungsproblemen (dTP) und dynamisch-stochastischen Tourensteuerungsproblemen (dTS) mit Zeitfensterrestriktionen und dynamischen Fahrzeiten untersucht. Unter *Tourensteuerung* wird in dieser Arbeit die Reaktion auf neue, entscheidungsrelevante Informationen im Rahmen der Tourendurchführung verstanden. Die Tourenplanung bestimmt zunächst eine vorläufige Lösung zum Tourenplanungsproblem. In dem Moment, in dem begonnen wird, die geplanten Transportvorgänge durchzuführen, wird vom Tourensteuerungsproblem gesprochen. Die jeweils aktuelle Verkehrslage kann von der geplanten Verkehrssituation abweichen. Die Tourensteuerung befasst sich mit möglichen Reaktionen auf veränderte Ausprägungen der dynamisch-stochastischen Fahrzeiten während der Durchführung der Transportvorgänge.

In Kapitel 2 dieser Arbeit wird zunächst ein Überblick über den Stand der Forschung zur Tourenplanung gegeben. Anschließend wird auf unterschiedliche Möglichkeiten eingegangen, im Zeitablauf variable Fahrzeiten für dynamische Tourenplanungsprobleme zu modellieren. Mögliche Zielsetzungen aus der Literatur zur Tourenplanung werden kritisch diskutiert. Schliesslich folgt eine Darstellung der in dieser Arbeit untersuchten dynamisch-deterministischen Tourenplanungsprobleme (dTP) anhand eines deskriptiven Modells.

Das Kapitel 3 führt dynamisch-stochastische Tourensteuerungsprobleme (dTS) ein. Es ist „festzustellen, daß sich in der aktuellen Literatur kein auf einer realitätsnahen Problemstellung aufbauendes Konzept findet, welches der Steuerung ein Maß an Reaktionsmöglichkeiten einräumen kann, daß der - durch Einsatz moderner

²Vgl. Lenstra und Rinnooy Kan 1981 für das Tourenplanungsproblem ohne Zeitfenster, zusätzliche Zeitfensterrestriktionen erhöhen die Komplexität des Problems.

Kommunikations- und Informationstechnologien möglich gewordenen - gesteigerten Verfügbarkeit der Daten in Echtzeit hinreichend gerecht wird.“³ Es wird in dieser Arbeit erstmals ein Modell zur Abbildung von zufällig eintretenden Störereignissen, welche die Fahrzeiten beeinflussen, entwickelt und experimentell untersucht.

In Kapitel 4 werden ausgewählte Lösungsansätze für Tourenplanungs- und -steuerungsprobleme vorgestellt. Es wird auf Modellierungsaspekte eingegangen, die es ermöglichen die Rechendauern für die experimentelle Analyse zu verkürzen und daher bei allen folgenden Verfahren verwendet werden. Es werden die einfachen Heuristiken, die von Ahn und Shin (1991) an das dTP angepasst wurden, in Abschnitt 4.2.2 und 4.2.3 vorgestellt. Das Cross-Exchange Verbesserungsverfahren wird in Abschnitt 4.3.2 näher betrachtet. In Abschnitt 4.4.2 folgt die Darstellung des Ameisenkoloniealgorithmus von Gambardella et al. (1999) und darauf aufbauend eine kurze Darstellung der Modifikationen von Donati et al. (2003a) in Abschnitt 4.4.3. In Abschnitt 4.5 wird ein neuer Ameisenkolonie-Algorithmus zur Lösung von dynamisch-deterministischen Tourenplanungsproblemen vorgestellt. Schließlich werden in Abschnitt 4.6 zur Lösung von dTS vier unterschiedliche Reaktionsmöglichkeiten auf Störungen während der Auslieferung von Waren kurz vorgestellt.

In Kapitel 5 werden die vorgestellten Verfahren experimentell untersucht. Zunächst wird die Erzeugung neuer Probleminstanzen für dynamisch-deterministische und dynamisch-stochastische Tourenplanungs- und steuerungsprobleme beschrieben. Schließlich erfolgt eine ausführliche Darstellung und Diskussion der Ergebnisse der Analyse.

³Bock (2004), S. 163

2 Charakterisierung von dynamisch-deterministischen Tourenplanungsproblemen (dTTP)

2.1 Stand der Forschung zu Tourenplanungsproblemen

2.1.1 Lösungsansätze für statische Tourenplanungsprobleme mit Zeitfensterrestriktionen

Dantzig und Ramser (1959) und Clarke und Wright (1964) befassen sich mit statisch-deterministischen Tourenplanungsproblemen.¹ Sie berücksichtigen dabei keine Zeitfensterrestriktionen. Ein grundlegender Artikel zu statisch-deterministischen Tourenplanungsproblemen mit Zeitfensterrestriktionen (im folgenden als VRPTW bezeichnet) stammt von Solomon (1987).² Darin werden neben einer Reihe von einfachen Konstruktionsverfahren (Savings-, Insertion- und Nearest-Neighbor-Heuristik) auch 56 Benchmarkprobleme zum VRPTW vorgestellt, auf die die Mehrzahl der späteren Veröffentlichungen zurückgreift, um die Lösungsgüte ihrer Verfahren zu vergleichen.

Exakte Lösungsverfahren zum VRPTW wurden beispielsweise von Desrochers et al. (1992), Kohl und Madsen (1997), Fisher et al. (1997) und Cook und Rich (1999) vorgestellt.³ Eine Gemeinsamkeit dieser exakten Lösungsverfahren ist der sehr hohe Zeitaufwand zur Lösungsfindung. Es existieren bislang noch nicht einmal zu allen Benchmark-Problemen von Solomon (1987) optimale Lösungen. Optimierende Lösungsverfahren sind nur sehr bedingt für einen praktischen Einsatz geeignet. Sie werden daher nicht näher betrachtet.

¹vgl. Dantzig und Ramser (1959) und Clarke und Wright (1964)

²vgl. Solomon (1987)

³vgl. hierzu und im folgenden Desrochers et al. (1992); Kohl und Madsen (1997); Fisher et al. (1997); Cook und Rich (1999)

Die große Mehrzahl der Veröffentlichungen zur Tourenplanung behandelt heuristische Lösungsverfahren.

- Solomon (1987) stellt ein modifiziertes Savings-Verfahren, mehrere sequentielle Einfügeheuristiken und ein Verfahren des besten Nachfolgers für VRPTW vor. Zur Verbesserung der resultierenden Lösungen wird ein 2-opt Nachbarschaftssuchverfahren verwendet. Die primäre Zielsetzung ist die Minimierung der Anzahl Touren, die sekundäre Zielsetzung die Minimierung der Gesamtbediendauer.⁴
- van Landeghem (1988) kombiniert ein modifiziertes Savings-Verfahren mit einem 3-opt Verbesserungsverfahren zur Lösung des VRPTW. Primäre Zielsetzung ist die Minimierung der Tourenzahl, als sekundäre Zielsetzung kann entweder die Gesamtentfernung oder die gesamte Wartezeit minimiert werden.⁵
- Savelsbergh (1992) untersucht diverse Nachbarschaftssuchverfahren für VRPTW. Ziel ist dabei die Minimierung der Gesamtbediendauer.⁶
- Potvin und Rousseau (1993) stellen eine parallele Einfügeheuristik zur Lösung des VRPTW vor. Primäres Ziel ist die Minimierung der Anzahl Touren, sekundäre Zielsetzung die Minimierung der Gesamtbediendauer.⁷
- Russell (1995) stellt eine parallele Einfügeheuristik und ein als „interchange heuristic“ bezeichnetes Verbesserungsverfahren zur Lösung des VRPTW vor.⁸ Die primäre Zielsetzung ist die Minimierung der Anzahl der Touren, sekundäres Ziel die Minimierung der Gesamtbediendauer, gefolgt von der Minimierung der Gesamtfahrzeit.
- Rochat und Taillard (1995) stellen zur Lösung des VRPTW ein Verfahren vor, welches sie mit „diversification, intensification and parallelization of local search“ bezeichnen. Zunächst wird eine große Menge von unterschiedlichen Tourenplänen erzeugt, deren Touren werden zusammen mit dem Zielfunktionswert des zugehörigen Tourenplans in einer Menge von Touren („adaptiver Speicher“) abgelegt. Aus dieser Menge werden sukzessive, zufallsgesteuert Touren entnommen, wobei die Auswahlwahrscheinlichkeit für eine bestimmte Tour umso höher ausfällt, je besser der Zielfunktionswert des zugehörigen Tourenplans ist. Primäre Zielsetzung ist die Minimierung der Tourenzahl, sekundäres Ziel ist die Minimierung der zurückgelegten Gesamtentfernung.⁹

⁴vgl. Solomon (1987)

⁵vgl. van Landeghem (1988)

⁶vgl. Savelsbergh (1992), S. 150

⁷vgl. Potvin und Rousseau (1993)

⁸vgl. Russell (1995)

⁹vgl. Rochat und Taillard (1995)

- Antes und Derigs (1996) entwerfen ein paralleles Konstruktionsverfahren zur Lösung des VRPTW. Primäre Zielsetzung ist die Minimierung der Tourenzahl, sekundäres Ziel ist die Minimierung der Gesamtbediendauer. Eine Einfügeheuristik konstruiert in Verbindung mit einer vorausschauenden Komponente gleichzeitig mehrere Touren, das resultierende Verfahren wird als Auktionsalgorithmus bezeichnet.¹⁰
- Potvin et al. (1996) entwerfen zur Lösung des VRPTW ein Tabu Search Verfahren, das auf einer 2-opt* und einer or-opt Nachbarschaft basiert. Primäre Zielsetzung ist die Minimierung der Tourenzahl, sekundäre Zielsetzung die Minimierung der Gesamtbediendauer.¹¹
- Potvin und Bengio (1996) stellen einen Genetischen Algorithmus für das VRPTW vor. Die Lösungskonstruktion basiert auf einer einfachen Einfügeheuristik, wobei nur die Reihenfolge, in der die Kundenaufträge eingefügt werden, durch den Genetischen Algorithmus festgelegt wird. Primäre Zielsetzung ist die Minimierung der Tourenzahl, sekundäre Zielsetzung die Minimierung der Summe aus Gesamtentfernung und gesamte Wartezeit.¹²
- Chiang und Russell (1996) verwenden ein Simulated Annealing Verfahren zur Lösungsfindung bei VRPTW . Ausgangslösung ist die parallele Einfügeheuristik von Russell (1995), zwei unterschiedliche Nachbarschaftsstrukturen werden untersucht. Primäres Ziel ist die Minimierung der Tourenzahl, sekundäre Zielsetzung die Minimierung der Gesamtfahrzeit.¹³
- Kilby et al. (1997) untersuchen ein Nachbarschaftssuchverfahren zur Lösung des VRPTW, welches als „guided local search“ bezeichnet wird. Um lokale Minima verlassen zu können, wird die Zielfunktion während der Suche modifiziert. Die lokale Suche wird dann basierend auf der modifizierten Zielfunktion fortgesetzt. Zielsetzung ist die Minimierung der Gesamtentfernung, die Anzahl der Touren wird dem Verfahren vorgegeben.¹⁴
- Chiang und Russell (1997) entwerfen zum VRPTW einen Tabu Search Algorithmus mit einer Tabuliste mit variabler Länge. Ausgangslösung ist die parallele Einfügeheuristik von Russell (1995). Primäre Zielsetzung ist die Minimierung der Tourenzahl, sekundäre Zielsetzung die Minimierung

¹⁰vgl. Antes und Derigs (1996)

¹¹vgl. Potvin et al. (1996), S. 158

¹²vgl. Potvin und Bengio (1996)

¹³vgl. Chiang und Russell (1996), S. 3f.

¹⁴vgl. Kilby et al. (1997)

der Gesamtentfernung und nachrangiges Ziel die Minimierung der gesamten Bediendauer.¹⁵

- Badeau et al. (1997) stellen zur Lösung des VRPTW ein Tabu Search Verfahren vor, welches auf der Cross-Exchange Nachbarschaft von Taillard et al. (1997) basiert. Die Summe aus Gesamtentfernung und Gesamtverspätung wird minimiert.¹⁶
- Shaw (1997) stellen ein Nachbarschaftssuchverfahren zum VRPTW vor.¹⁷ Dabei wird ein greedy-Verbesserungsverfahren mit einer sehr großen Nachbarschaft, die auf einem Constraint-Programming Ansatz beruht, kombiniert. Eine Menge von zufällig ausgewählten Kundenaufträgen wird aus bestehenden Touren entnommen. Mit Hilfe eines branch-and-bound Ansatzes werden die Kundenaufträge an den jeweils bestmöglichen Positionen wieder eingefügt. Der branch-and-bound Ansatz wird in jeder Iteration spätestens nach einer vorgegebenen Rechendauer abgebrochen, um die Rechendauern zu begrenzen. Wenn die neue erzeugte Lösung zu einer Verbesserung des Zielfunktionswerts führt, wird die neue Lösung akzeptiert. Die primäre Zielsetzung ist die Minimierung der Anzahl der Touren, sekundärs Ziel ist die Minimierung der Gesamtentfernung.
- Das Verbesserungsverfahren, das Shaw (1998) zum VRPTW vorstellt, entspricht weitestgehend dem Verfahren, das von Shaw (1997) beschrieben wird.¹⁸ Der wesentliche Unterschied besteht in der umfangreicheren parametrischen Analyse.
- Homberger und Gehring (1999a) versuchen mit einem Simulated Annealing Verfahren die Anzahl der Touren und anschließend die Gesamtentfernung bei VRPTW zu minimieren.¹⁹
- Homberger und Gehring (1999b) beschreiben zwei evolutionäre Algorithmen für das VRPTW. Primäres Ziel ist die Minimierung der Tourenzahl, sekundäres Ziel die Minimierung der Gesamtentfernung.²⁰
- Gambardella et al. (1999) entwerfen einen Ameisenkolonialgorithmus für das VRPTW. Primäres Ziel ist die Minimierung der Tourenzahl, sekundäres Ziel die Minimierung der Gesamtfahrzeit.²¹ Eine genauere Beschreibung dieses Verfahrens findet sich in Abschnitt (4.4.2).

¹⁵vgl. Chiang und Russell (1997)

¹⁶vgl. Badeau et al. (1997), S. 111

¹⁷vgl. Shaw (1997)

¹⁸vgl. Shaw (1998) und Shaw (1997)

¹⁹vgl. Homberger und Gehring (1999a), S. 483

²⁰vgl. Homberger und Gehring (1999b)

²¹vgl. Gambardella et al. (1999)

- Liu und Shen (1999) schlagen eine neue Nachbarschaft, die sog. „route based neighborhood“ für VRPTW vor. Primäres Ziel ist die Minimierung der Tourenzahl, sekundäres Ziel die Minimierung der Gesamtdistanz.²²
- Schulze und Fahle (1999) entwerfen ein paralleles Tabu Search Verfahren zur Lösung des VRPTW. Basis ist eine Nachbarschaft, die Kundenaufträge verschiebt und dabei auch unzulässige Zwischenlösungen erlaubt. Primäres Ziel ist die Minimierung der Tourenzahl, sekundäres Ziel die Minimierung der Summe aus Gesamtfahrzeit und Gesamtwartzeit.²³
- de Backer et al. (2000) stellen ein Verbesserungsverfahren zu VRPTW vor, welches sie als Guided Local Search bezeichnen. Ein lokales Suchverfahren sucht Lösungen mit möglichst geringer Gesamtfahrzeit. Dabei wird in jeder Iteration die beste Nachbarlösung betrachtet. Sobald ein lokales Minimum erreicht wird, werden bestimmten Fahrten der Lösung „Strafkosten“ zugeordnet und die Suche fortgesetzt. Die „Strafkosten“ steigen mit der Häufigkeit des Auftretens bestimmter Fahrten während der lokalen Suche.²⁴
- Homberger (2000) entwirft eine verteilt parallele Metaheuristik zur Lösung des VRPTW, die unterschiedliche Metaheuristiken kombiniert. Beispielsweise wird eine Evolutionsstrategie zur Minimierung der Tourenzahl mit einem Tabu Search Verfahren zur Minimierung der Gesamtdistanz kombiniert. Die primäre Zielsetzung ist die Minimierung der Tourenzahl, sekundäres Ziel ist die Minimierung der Gesamtdistanz.²⁵
- Schrimpf et al. (2000) stellen ein Verbesserungsverfahren für VRPTW vor, welches sie mit „ruin-and-recreate“ bezeichnen.²⁶ Primäre Zielsetzung ist die Minimierung der Tourenzahl, sekundäres Ziel ist die Minimierung der Gesamtdistanz. Die Grundidee entspricht einem Simulated Annealing Ansatz. Die Besonderheit ist die Nachbarschaftsdefinition. Eine Nachbarlösung wird erzeugt, indem aus einer gegebenen Lösung eine Menge von Kundenaufträgen entfernt wird. Die entfernten Kundenaufträge werden nicht länger bedient. Unterschiedliche Vorschriften zur Auswahl der Kunden, die entfernt werden sollen, werden beschrieben und untersucht. Anschließend werden die zuvor entfernten Kundenaufträge sukzessive anhand einer Einfügeheuristik wieder in die unvollständige Lösung eingefügt.

²²vgl. Liu und Shen (1999), S. 486

²³vgl. Schulze und Fahle (1999), S. 586

²⁴vgl. de Backer et al. (2000)

²⁵vgl. Homberger (2000), S. 19-21

²⁶vgl. Schrimpf et al. (2000)

- Ioannou et al. (2001) stellen ein sequentielles, vorausschauendes Konstruktionsverfahren zum VRPTW vor. Primäre Zielsetzung ist die Minimierung der Anzahl der Touren, sekundäres Ziel ist die Minimierung der Gesamtfahrzeit.²⁷
- Ibaraki et al. (2001) entwerfen eine neue Nachbarschaft für VRPTW, die sie mit „cyclic exchange“ bezeichnen. Mit einer Nachbarschaftssuche wird die Gesamtentfernung, bei einer vorgegebenen Tourenzahl, minimiert.²⁸
- Berger et al. (2001) stellen einen parallelen genetischen Algorithmus zum VRPTW vor.²⁹ Eine Population verfolgt das Ziel der Minimierung der zurückgelegten Gesamtentfernung, eine zweite Population versucht Lösungsbestandteile zu generieren, die die Zeitfensterrestriktionen möglichst wenig verletzen.
- Czech und Czarnas (2002) schlagen ein paralleles Simulated Annealing Verfahren zur Lösung des VRPTW vor.³⁰ Die primäre Zielsetzung ist die Minimierung der Tourenzahl, sekundäre Zielsetzung ist die Minimierung der Gesamtentfernung. Es werden parallel mehrere Lösungen mit Simulated Annealing verbessert, wobei zwischen jeweils die global beste bekannte Lösung ausgetauscht wird.
- Ombuki et al. (2002) beschreiben ein hybrides Verfahren zur Lösung des VRPTW, das einen Genetischen Algorithmus mit einem Tabu-Search Verfahren und Simulated Annealing kombiniert. Aus der Menge der Lösungen mit minimaler Tourenzahl ist diejenige zu wählen, welche die Gesamtentfernung minimiert.³¹
- Li und Lim (2003) kombinieren zur Lösung des VRPTW eine Simulated Annealing basierte Cross Exchange Nachbarschaftssuche mit einer Tabuliste. Ausgangslösung ist das Ergebnis der Einfügeheuristik von Solomon (1987). Primäre Zielsetzung ist die Minimierung der Tourenzahl, gefolgt von der Minimierung der Gesamtfahrzeit. Nachfolgende Zielsetzungen sind die Minimierung der Bediendauer und schließlich die Minimierung der Gesamtwartzeit.³²
- Bent und Van Hentenryck (2004) verbinden zur Lösung des VRPTW Simulated Annealing mit einem „Large Neighborhood“ Suchverfahren. Es wird primär die Anzahl der Touren und sekundär die Gesamtfahrzeit minimiert.³³

²⁷vgl. Ioannou et al. (2001)

²⁸vgl. Ibaraki et al. (2001)

²⁹vgl. Berger et al. (2001)

³⁰vgl. Czech und Czarnas (2002)

³¹vgl. Ombuki et al. (2002)

³²vgl. Li und Lim (2003), S. 120

³³vgl. Bent und Van Hentenryck (2004)

- Le Bouthillier und Crainic (2005) kombinieren zur Lösung des VRPTW verschiedene Metaheuristiken zur parallelen Konstruktion einer großen Menge von Touren („solution warehouse“ Prinzip). Aus dieser Menge werden Touren zu Tourenplänen zusammengesetzt. Ziel ist die Minimierung der gesamten Auslieferungskosten. Für die experimentelle Analyse werden der Tourenzahl hinreichend hohe Kosten zugeordnet, so daß die Gesamtentfernung als sekundäre Zielsetzung dient.³⁴ Le Bouthillier und Crainic (2005) geben außerdem einen aktuellen Literaturüberblick zu VRPTW.³⁵

Für detaillierte Beschreibungen der einzelnen Verfahren sei auf die genannten Veröffentlichungen verwiesen. Eine Gegenüberstellung der Lösungsqualität einiger ausgewählter Verfahren und der jeweils erforderlichen Rechenzeiten findet sich bei Bräysy und Gendreau (2005a) und Bräysy und Gendreau (2005b).³⁶

Die jeweils aktuell besten bekannten Lösungen zu den Benchmarkproblemen von Solomon (1987) stehen im Internet auf der Website von Prof. Solomon zur Verfügung.³⁷

2.1.2 Lösungsansätze für dynamisch-deterministische Tourenplanungsprobleme mit Zeitfensterrestriktionen

Die in dieser Arbeit behandelten dynamisch-deterministischen Tourenplanungsprobleme mit Zeitfensterrestriktionen berücksichtigen - im Gegensatz zu statischen VRPTW - variable Fahrzeiten zwischen allen Kunden und den Kunden und dem Depot. Die Fahrzeiten sind also nicht während des gesamten Planungszeitraums konstant.

- Dynamisch-deterministische Tourenplanungsprobleme mit im Zeitablauf variablen Fahrzeiten werden erstmals von Ahn und Shin (1991) vorgestellt.³⁸ Primäre Zielsetzung ist die Minimierung der Tourenzahl, gefolgt von der Minimierung der Summe der Fahrzeiten.
- Gietz (1994) greift die Problemstellung und den Ansatz von Ahn und Shin (1991) auf und untersucht dynamisch-deterministische Tourenplanungsprobleme anhand von realen Problemstellungen. Es werden drei Verfahren, die von Solomon (1987) für das VRPTW untersucht wurden - eine Einfügeheuristik, ein Verfahren des Besten Nachfolgers und das Savings-Verfahren - an dynamische

³⁴vgl. Le Bouthillier und Crainic (2005), S. 1687

³⁵vgl. Le Bouthillier und Crainic (2005), S. 1687f.

³⁶vgl. Bräysy und Gendreau (2005a), S. 104-118; Bräysy und Gendreau (2005b), S. 119-139

³⁷vgl. <http://web.cba.neu.edu/~msolomon/heuristi.htm> (26. Juni 2005, 17:50 Uhr)

³⁸vgl. Ahn und Shin (1991), S. 393-400

Tourenplanungsprobleme angepasst und durch ein Verbesserungsverfahren ergänzt.³⁹ Die Fahrzeiten zwischen den Kunden werden dabei jeweils als abschnittsweise definierte Funktionen in Abhängigkeit vom Abreisezeitpunkt modelliert. Um die Zulässigkeitsprüfung bei der Konstruktion von Lösungen zu beschleunigen, diskutieren Ahn und Shin (1991) notwendige und hinreichende Bedingungen für zulässige Nachbarschaftstauschoperationen.⁴⁰ Dafür ist die Gültigkeit des FIFO-Prinzips (von Gietz (1994) als Überholverbot bezeichnet) Voraussetzung.⁴¹ Der Ansatz von Ahn und Shin (1991) kann dessen Einhaltung allerdings nicht gewährleisten.

- Ichoua et al. (2003) wählen einen anderen Ansatz zur Modellierung der variablen Fahrzeiten als Ahn und Shin (1991).⁴² Die im Zeitablauf variablen Fahrzeiten zwischen zwei Kunden werden mit Hilfe von abschnittsweise definierten Geschwindigkeitsfunktionen bestimmt.⁴³ Die Primäre Zielsetzung ist die Minimierung der Anzahl der Touren, aus der Menge der Lösungen mit minimaler Tourenzahl ist die Lösung mit der geringsten Summe der Fahrzeiten auszuwählen.⁴⁴ Ichoua et al. (2003) beschränken das Netzwerk, das der Tourenplanung zugrunde gelegt wird, auf die zu beliefernden Kunden und das Depot. Alternative Wege zwischen Kunden werden im Rahmen des Modells nicht explizit abgebildet, sie sind vor Beginn der Planung festzulegen.⁴⁵ Ichoua et al. (2003) stellen basierend auf dieser Modellierung der Fahrzeiten ein paralleles Tabu-Search Verfahren zur dynamischen Tourenplanung vor. Allerdings berücksichtigt der Ansatz keine Kapazitätsrestriktionen. Weiterhin wird angenommen, daß Zeitfensterrestriktionen verletzt werden dürfen, obwohl es sich um ein dynamisch-deterministisches Modell handelt. Die resultierenden Lösungen sind somit nicht notwendigerweise zulässige Lösungen für das in dieser Arbeit untersuchte dynamisch-deterministische Tourenplanungsproblem. Die Anzahl der erforderlichen Touren wird nicht durch den Algorithmus ermittelt, sondern es wird dafür auf den Autoren bekannte Lösungen zu ähnlichen, statisch-deterministischen Tourenplanungsproblemen zurückgegriffen (vgl. Ichoua et al. (2003), S. 388) - ein Vorgehen, das für praktische Zwecke ungeeignet ist, weil für praktische Problemstellungen in der Regel noch keine Lösungen bekannt sind. Die Darstellung des Tabu-Search Verfahrens ist

³⁹or-opt Verbesserungsverfahren, vgl. Ahn und Shin (1991), S. 398

⁴⁰vgl. dazu Abschnitt (4.1)

⁴¹vgl. Ahn und Shin (1991), S. 394 und Gietz (1994) S. 154

⁴²vgl. Ichoua et al. (2003), S. 382-385

⁴³vgl. hierzu und im folgenden Ichoua et al. (2003), S. 384

⁴⁴vgl. Ichoua et al. (2003), S. 386

⁴⁵Alternative Wege können vor der eigentlichen Durchführung des Verfahrens berücksichtigt werden. Auf die Ermittlung der zu wählenden Wege zwischen den Kunden gehen Ichoua et al. (2003) allerdings nicht ein.

undurchsichtig und teilweise unverständlich. Nachvollziehbare Aussagen über die Lösungsqualität des Verfahrens fehlen. Das Verfahren wird daher im folgenden nicht näher betrachtet.

- Donati et al. (2003a) stellen, basierend auf der Modellierung der Fahrzeiten von Ichoua et al. (2003), einen auf dynamisch-deterministische Tourenplanungsprobleme angepassten Ameisenalgorithmus vor.⁴⁶ Der Ameisenalgorithmus basiert auf dem Ansatz von Gambardella et al. (1999) für statische Tourenplanungsprobleme.⁴⁷ Zielsetzung des Ansatzes von Donati et al. (2003a) ist primär die Minimierung der Anzahl Touren, gefolgt von der Minimierung der Summe der Fahrzeiten.⁴⁸ Zu einer genauen Beschreibung des Ameisenkolonialgorithmus von Donati et al. (2003a) vgl. Abschnitt 4.4.3.

2.1.3 Abgrenzung zu verwandten Problemstellungen

Neben den aufgeführten Veröffentlichungen gibt es eine Reihe verwandter Problemstellungen, die häufig unter dem Begriff Tourenplanung zusammengefasst werden.

- Balakrishnan (1993) untersucht eine Variante des Tourenplanungsproblems mit „weichen“ Zeitfensterrestriktionen: Der Beginn der Bedienung eines Kunden ist auch nach dem Ende seines Zeitfensters zulässig, bei einer verspäteten Bedienung fallen allerdings Strafkosten an.⁴⁹ Unterschiedliche Tabu-Search Ansätze zu diesem Problem werden beispielsweise von Taillard et al. (1997) und Chiang und Russell (2004) vorgestellt.⁵⁰
- Lau et al. (2003) stellen ein Tourenplanungsproblem mit beschränkter Fahrzeugzahl vor. Zulässige Lösungen umfassen dabei nicht notwendigerweise alle Kundenaufträge, zudem dürfen Zeitfensterrestriktionen verletzt werden.⁵¹ Die Zielsetzung ist die Bedienung einer möglichst großen Anzahl von Kunden im Planungszeitraum.⁵²
- Taillard et al. (1995) betrachtet Tourenplanungsprobleme bei denen ein Fahrzeug im Planungszeitraum mehrere Touren durchführen kann.⁵³

⁴⁶ vgl. Donati et al. (2003a)

⁴⁷ vgl. Gambardella et al. (1999)

⁴⁸ vgl. Donati et al. (2003a), S. 11

⁴⁹ vgl. Balakrishnan (1993)

⁵⁰ vgl. Taillard et al. (1997); Chiang und Russell (2004)

⁵¹ vgl. Lau et al. (2003), S. 559

⁵² vgl. Lau et al. (2003), S. 560

⁵³ vgl. Taillard et al. (1995)

- Sariklis und Powell (2000) untersuchen das „open vehicle routing problem“, bei dem die Lieferfahrzeuge nach Beendigung der Bedienung aller Kunden nicht ins Depot zurückkehren müssen.⁵⁴ Fu et al. (2005) entwerfen ein Tabu Search Verfahren für das gleiche Problem.⁵⁵
- Cordeau et al. (2001) stellen zwei Varianten des Tourenplanungsproblems mit Zeitfensterrestriktionen vor: Das „periodic vehicle routing problem with time windows“ (PVRPTW) und das „multi-depot vehicle routing problem with time windows“ (MVRPTW).⁵⁶ Beim PVRPTW wird ein mehrperiodiger Planungshorizont untersucht. Kunden sind mehrfach in unterschiedlichen Perioden zu besuchen, neben der Belieferungsreihenfolge und der Zuordnung zu Fahrzeugen sind auch die Belieferungsperioden zu bestimmen. Das MVRPTW ist durch mehrere Depots an unterschiedlichen Standorten gekennzeichnet. Neben der Zuordnung von Kundenaufträgen zu Touren muss auch eine Zuordnung von Kundenaufträgen zu Depots erfolgen.
- Tarantilis et al. (2003) betrachten ein Tourenplanungsproblem mit heterogenem Fuhrpark ohne Zeitfensterrestriktionen.⁵⁷
- Mit Tourenplanungsproblemen mit heterogenem Fuhrpark und Zeitfensterrestriktionen befassen sich beispielsweise Dullaert et al. (2002).⁵⁸
- Einen aktuellen Literaturüberblick über das Tourenplanungsproblem mit gleichzeitiger Auslieferung und Abholung von Gütern („vehicle routing problem with pickups and deliveries“), geben Nagy und Salhi (2005).⁵⁹
- Eine Abgrenzung des VRPTW von verwandten Problemstellungen, wie z.B. dem „Vehicle Routing Problem with Backhauls“ (im Anschluss an die Auslieferung aller Güter erfolgt mit dem gleichen Fahrzeug eine Sammeltour um Güter abzuholen), Dial-a-Ride-Problemen und Pickup-and-Delivery-Problemen, sowie eine Einordnung der Tourenplanung in den Rahmen von allgemeinen Transportproblemen gibt Bock (2004).⁶⁰

In der Literatur werden gelegentlich Problemstellungen (abweichend von der in dieser Arbeit verwendeten Definition) als dynamische Tourenplanungsprobleme (dynamic

⁵⁴vgl. Sariklis und Powell (2000)

⁵⁵vgl. Fu et al. (2005)

⁵⁶vgl. Cordeau et al. (2001)

⁵⁷vgl. Tarantilis et al. (2003)

⁵⁸vgl. Dullaert et al. (2002)

⁵⁹vgl. Nagy und Salhi (2005), S. 129

⁶⁰vgl. Bock (2004), S. 96-102

vehicle routing problem, DVRP) bezeichnet, bei denen während der Auslieferung neue Kundenaufträge eingehen.⁶¹

Mit stochastischen Tourenplanungsproblemen ohne Zeitfensterrestriktionen befassen sich Keynon und Morton (2003). Es wird ein Ansatz der stochastischen Programmierung vorgestellt. Die Lieferfahrzeuge weisen keine Kapazitätsrestriktionen auf. Die Reise- und Servicedauern sind zufällig verteilt, die zugehörigen Wahrscheinlichkeitsfunktionen sind bekannt. Eine Nachoptimierung der Touren nach Bekanntwerden neuer Informationen wird nicht zugelassen. Die Zielsetzung lautet *Minimierung des Zeitpunkts, in dem die letzte Tour beendet wird* (completion time).⁶² Es ist fraglich, ob überhaupt von einem Tourenplanungsproblem gesprochen werden kann, da weder Kundenzeitfenster- noch Fahrzeugkapazitätsrestriktionen berücksichtigt werden.⁶³

2.2 Dynamisch-deterministische Tourenplanungsprobleme

2.2.1 Verkehrswegenetzwerke zur Abbildung von dynamisch-deterministischen Tourenplanungsproblemen

Zur Untersuchung von VRPTW wird der Standort eines jeden Kunde und des Depots in der Regel vereinfacht durch Koordinaten in der Ebene abgebildet.⁶⁴ Die Entfernungen zwischen zwei Kunden und zwischen je einem Kunden und dem Depot werden durch die euklidischen Distanzen in der Ebene approximiert, die Fahrzeiten werden aus Vereinfachungsgründen den Entfernungen gleichgesetzt.

Diese Form der Abbildung ist für dynamisch-deterministische Tourenplanungsprobleme nicht geeignet. Um Entfernungen und Fahrzeiten unabhängig voneinander modellieren zu können, ist es erforderlich die räumlichen und zeitlichen Beziehungen zwischen Standorten eindeutig zu definieren. Zum einen muß für jede Beziehung eine Entfernung festgelegt werden, die unabhängig vom betrachteten Zeitpunkt ist. Zum anderen ist die Fahrzeit variabel, in Abhängigkeit vom betrachteten Zeitpunkt, zu modellieren, damit Schwankungen der Fahrzeiten im Zeitablauf abgebildet werden können. Es wird daher das Tourenplanungsproblem auf Basis eines Verkehrswegenetzwerks abgebildet.

Es ist im folgenden davon auszugehen, daß das jeweilige Tourenproblem auf einem

⁶¹vgl. dazu Bertsimas und van Ryzin (1991); Ichoua et al. (2000); Larsen (2000); Larsen et al. (2002)

⁶²vgl. Keynon und Morton (2003), S. 70

⁶³vgl. Keynon und Morton (2003), S. 70

⁶⁴vgl. hierzu und im folgenden beispielsweise Solomon (1987), S. 255f.

bewerteten Graphen gegeben ist.⁶⁵ Der Ort, an dem Auslieferungsfahrten beginnen und enden, wird als *Depot* bezeichnet und durch den Knoten 0 repräsentiert.⁶⁶

Bei einem *Verkehrswegenetzwerk* handelt es sich um einen gerichteten Graphen. Die Standorte der Kunden und des Depots werden durch Knoten abgebildet. Darüber hinaus enthält der Graph weitere Zwischenknoten (im folgenden als *Kreuzungen* bezeichnet), die für die Abbildung von alternativen Wegen zwischen zwei Kunden und zwischen den Kunden und dem Depot erforderlich sind. Die Pfeile des Graphen bilden die räumlichen und zeitlichen Beziehungen (Entfernungen und Fahrzeiten) zwischen je zwei Knoten (Depot, Kunden und Kreuzungen) ab. Sie entsprechen den Strassen, die zwei Standorte direkt miteinander verbinden. In Abbildung 2.1 ist beispielhaft ein einfaches Verkehrswegenetzwerk abgebildet. Je zwei entgegengerichtete Pfeile werden der Übersichtlichkeit halber durch eine Kante dargestellt.

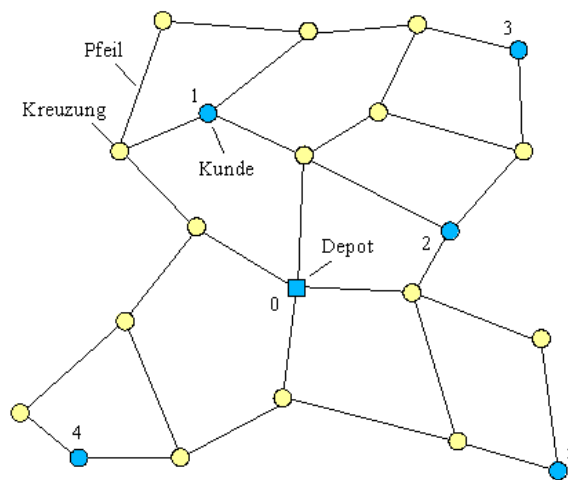


Abbildung 2.1: Verkehrswegenetzwerk

Im folgenden wird die Menge aller Kunden mit C bezeichnet. Mit C_0 wird die Menge der Kunden zuzüglich des Depots bezeichnet. Die Menge aller Knoten des Verkehrswegenetzwerks wird mit N bezeichnet, sie enthält neben dem Depot und den Kunden auch die Kreuzungen. In dem Verkehrswegenetzwerk, das in Abbildung 2.1 abgebildet ist, entspricht die Menge $C = \{1; 2; 3; 4; 5\}$, die Menge $C_0 = C \cup \{0\} = \{0; 1; 2; 3; 4; 5\}$ und die Menge N allen abgebildeten Knoten. Die Bezeichner m und n werden im folgenden für Knoten aus der Menge N verwendet. Die Bezeichner i und j werden für Knoten aus der Menge C_0 verwendet. Die Menge der Knoten die mit dem Knoten m direkt (über einen Pfeil) verbunden ist, wird als benachbart bezeichnet, sie erhält das Symbol $\Gamma^+(m)$. Die Entfernung zwischen zwei benachbarten Knoten m und n wird mit d_{mn} bezeichnet, es gilt $d_{mn} = d_{nm}$. Die Fahrzeit von m nach n wird mit $t_{mn}(\tau_m)$ bezeichnet. Dabei ist τ_m der Zeitpunkt zu dem der Knoten m verlassen

⁶⁵vgl. Domschke (1997), S. 205

⁶⁶vgl. Domschke (1997), S. 205

wird. Die Fahrzeit ist im Zeitablauf nicht konstant, sondern kann in Abhängigkeit vom Abfahrtszeitpunkt variieren. Die Fahrzeit wird als Pfeil abgebildet, da die Fahrzeit zwischen zwei Knoten des Verkehrswegenetzwerks nicht notwendigerweise in beiden Richtungen immer gleich ist.

Eine *Tour* ist eine nach der Belieferungsreihenfolge sortierte Liste von direkt nacheinander besuchten Kunden, die auf ein und derselben, in dem Depot beginnenden und in dem Depot endenden Rundreise bedient werden.⁶⁷ Nur Knoten aus der Menge C_0 (Depot und Kundenaufträge) können Bestandteil einer Tour sein. Eine Tour wird immer durch genau ein Fahrzeug durchgeführt. Eine zulässige Lösung eines Tourenproblems wird als *Tourenplan* bezeichnet. Ein Tourenplan ist eine Menge von Touren, die alle Bedingungen des betrachteten Problems erfüllen.⁶⁸ Ein Tourenplan erhält im folgenden den Bezeichner X . In Abbildung 2.2 ist ein Tourenplan, der aus zwei Touren besteht, abgebildet.

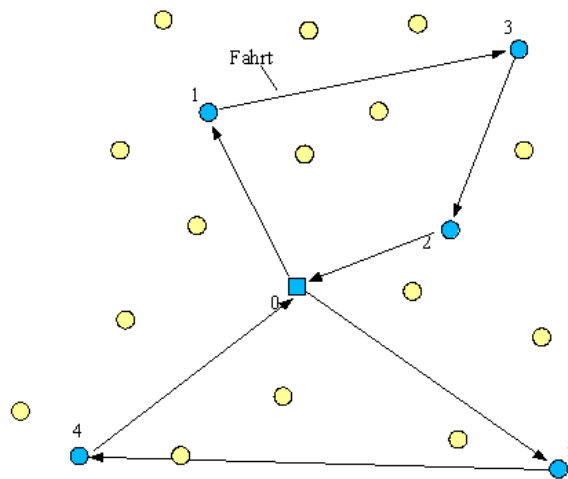


Abbildung 2.2: Touren

Die erste Tour läßt sich als Liste $(0, 1, 3, 2, 0)$ darstellen - die zweite Tour wird als Liste $(0, 5, 4, 0)$ dargestellt. Eine Tour enthält Informationen über die Zuordnung von Kunden zu Touren und über die Belieferungsreihenfolge der Kunden. Die Tour, die Fahrzeug z durchführt, wird im folgenden mit T_z bezeichnet. Im folgenden wird der Bezeichner z sowohl für eine Tour als auch für das Lieferfahrzeug, welches die Tour durchführt verwendet. Es wird nicht zwischen Lieferfahrzeug und Tour unterschieden, da ein homogener Fuhrpark unterstellt wird und angenommen wird, daß jedes Lieferfahrzeug im Planungszeitraum nur eine Tour durchführen darf.⁶⁹ Allgemein wird eine Tour in Listenform im folgenden als $T_z = (c_0^z, c_1^z, \dots, c_{pos}^z, \dots, c_{|T_z|}^z, c_{|T_z|+1}^z)$ dargestellt. $|T_z|$ bezeichnet die Anzahl Kunden, die auf der Tour T_z bedient werden. c_{pos}^z ist der Kunde,

⁶⁷ vgl. Domschke (1997), S. 206

⁶⁸ vgl. Domschke (1997), S. 206

⁶⁹ vgl. die Problemdefinition in Abschnitt 2.2.4

der als *pos*-tes auf der Tour T_z bedient wird. c_0^z und $c_{|T_z|+1}^z$ entsprechen dem Depot, das den Beginn bzw. das Ende jeder Tour kennzeichnen. Neben der Darstellung als Liste läßt sich eine Tour auch als Menge von Fahrten abbilden.

Die *Fahrt* (i, j) eines Fahrzeugs z von i nach j beschreibt den Sachverhalt, daß der Standort j direkt im Anschluss an den Standort i mit Fahrzeug z angesteuert wird, wobei i und j das Depot oder ein Kundenstandort sein können. Die erste Tour aus Abbildung 2.2 enthält die Fahrten $(0, 1)$, $(1, 3)$, $(3, 2)$ und $(2, 0)$, die zweite Tour die Fahrten $(0, 5)$, $(5, 4)$ und $(4, 0)$. Es gilt $T_1 = \{(0, 1), (1, 3), (3, 2), (2, 0)\}$ und $T_2 = \{(0, 5), (5, 4), (4, 0)\}$. Für die formale Darstellung ist es zweckmäßig die Fahrten als binäre Entscheidungsvariablen x_{ij}^z abzubilden. Gilt $x_{ij}^z = 1$, ist die Fahrt (i, j) Bestandteil der Tour T_z , der Kunde oder das Depot j wird direkt im Anschluß an i durch das Fahrzeug z angesteuert. Der Tourenplan X aus Abbildung 2.2 läßt sich als $x_{01}^1 = 1$, $x_{13}^1 = 1$, $x_{32}^1 = 1$, $x_{20}^1 = 1$, $x_{05}^2 = 1$, $x_{54}^2 = 1$, $x_{40}^2 = 1$ und $x_{ij}^z = 0$ für alle übrigen $i, j \in C_0$, $z \in Z$ darstellen.

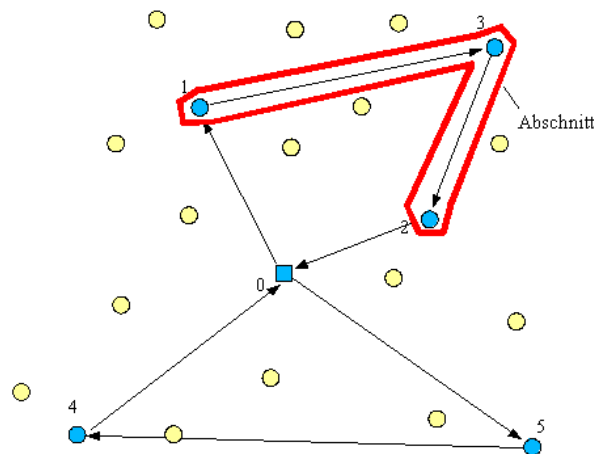


Abbildung 2.3: Abschnitt

Als *Abschnitt* wird eine Menge von direkt nacheinander bedienten Kundenaufträgen bezeichnet, die auf einer Tour von dem gleichen Fahrzeug beliefert werden. Ein Abschnitt ist ein Teil einer Tour bzw. eine Menge von direkt aufeinanderfolgenden Fahrten. In Abbildung 2.3 ist der Abschnitt $(1, 3, 2)$ eingekreist, er besteht aus den Fahrten $(1, 3)$ und $(3, 2)$.

Um die zurückgelegte Entfernung einer Tour und die dafür benötigte Fahrzeit zu ermitteln, muß zu jeder Fahrt ein zugehöriger Weg im Verkehrsnetzwerk bestimmt werden. In Abhängigkeit vom Abreisezeitpunkt τ_i können unterschiedliche Wege im Verkehrsnetzwerk von i nach j vorteilhaft sein. Somit ist auch die Entfernung $d_{ij}(\tau_i)$ im Zeitablauf variabel. Die Entscheidungsvariablen x_{ij}^z des dynamisch-deterministischen Tourenplanungsproblems legen nur fest, ob die Fahrt (i, j) mit dem Fahrzeug z Bestandteil des Tourenplans wird.

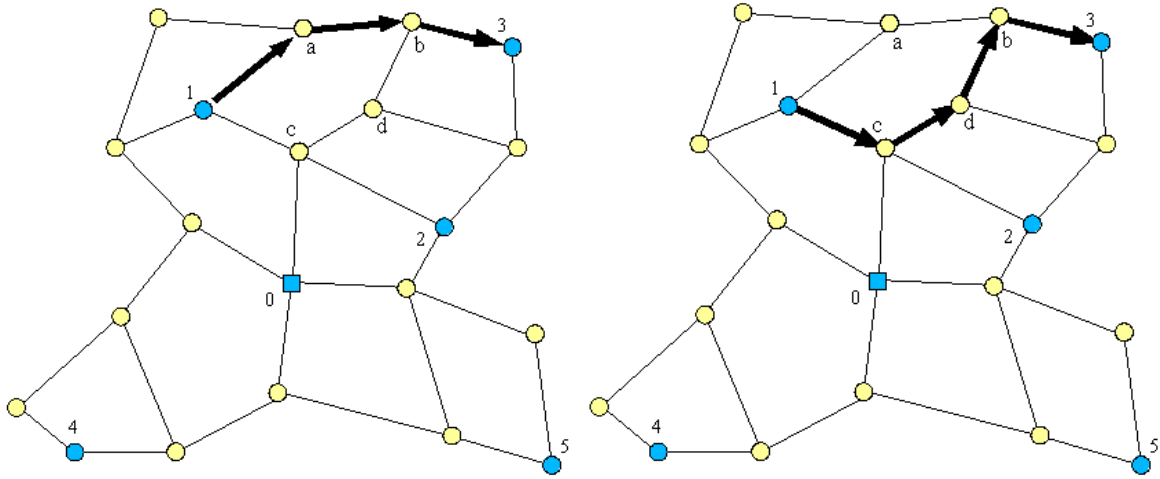


Abbildung 2.4: Beispiel: Alternative Wege vom Kunden 1 zum Kunden 3

In Abbildung 2.4 sind zwei alternative Wege vom Kunden 1 zum Kunden 3 dargestellt. Es handelt sich dabei jeweils um die gleiche Fahrt $(1, 3)$ ($x_{13}^1 = 1$). Links ist der Weg $(1, a, b, 3)$, rechts der Weg $(1, c, d, b, 3)$ abgebildet. Welcher Weg zu wählen ist, hängt vom Abreisezeitpunkt τ_1 beim Kunden 1 ab. Wie die zu einer Fahrt zu einem bestimmten Abreisezeitpunkt zugehörigen Wege zu bestimmen sind, wird in Abschnitt 2.2.2.4 näher beschrieben.

Eine *Route* ist die Folge von Pfeilen (m, n) im Verkehrsnetzwerk, welche die zu einer Tour zugehörigen Wege im Verkehrsnetzwerk eindeutig beschreibt. Eine Route entspricht einer Folge von Wegen, bei denen der Endpunkt des vorangehenden Wegs dem Anfangspunkt des nachfolgenden Wegs entspricht, der erste Weg im Depot beginnt und der letzte Weg wieder im Depot endet. Zu jeder Fahrt aus jeder Tour gehört ein Weg. Zu einer gegebenen Tour gehört immer eine eindeutig bestimmte Route. Die Wege bilden die Grundlage zur Ermittlung der Zeitpunkte y_i , zu denen mit der Bedienung des Kunden i begonnen wird, zur Berechnung der dynamischen Fahrzeiten $t_{ij}(\tau_i)$ einer Fahrt vom Kunden i zum Kunden j in Abhängigkeit vom Abreisezeitpunkt τ_i , und zur Bestimmung der zurückzulegenden Entfernung $d_{ij}(\tau_i)$ auf einer Fahrt vom Kunden i zum Kunden j . In Abbildung 2.5 sind die Routen die zu dem Tourenplan aus Abbildung 2.2 gehören, abgebildet.

Als Gesamtentfernung $dist(X)$ wird die Summe der Entfernungen über alle Fahrten aus allen Touren eines Tourenplans X bezeichnet. $dist(X) = \sum_{z \in Z} \sum_{i \in C_0} \sum_{j \in C_0} d_{ij}(\tau_i) \cdot x_{ij}^z$.

Die Gesamtfahrzeit $travel(X)$ bezeichnet die Summe der Fahrzeiten über alle Fahrten aus allen Touren eines Tourenplans X . $travel(X) = \sum_{z \in Z} \sum_{i \in C_0} \sum_{j \in C_0} t_{ij}(\tau_i) \cdot x_{ij}^z$.

Die Gesamtbediendauer $sched(X)$ ist die Summe der Bediendauern über alle Touren. Sie entspricht der Summe aus Fahrzeiten $t_{ij}(\tau_i)$, Servicezeiten s_i und Wartezeiten w_i . Wird angenommen, daß das Depot zum Zeitpunkt $e_0 = 0$ verlassen wird, lassen sich die

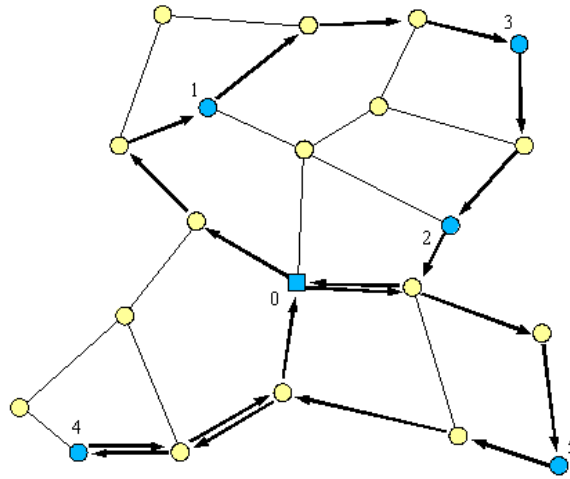


Abbildung 2.5: Routen

Gesamtbediendauer allerdings auch einfacher als die Summe der Zeitpunkte ermitteln, zu denen die Fahrzeuge ins Depot zurückkehren. Ist y_0^z ist der Zeitpunkt, zu dem die Tour z beendet ist, gilt $sched(X) = \sum_{z \in Z} y_0^z$.

$anz(X)$ bezeichnet die Anzahl Touren, die der Tourenplan X benötigt, um alle Kunden zu beliefern. ζ_z bezeichnet eine Binärvariable, gilt $\zeta_z = 1$ wird das Fahrzeug z eingesetzt, sonst gilt $\zeta_z = 0$. Formal folgt daraus: $anz(X) = \sum_{z \in Z} \zeta_z$

2.2.2 Dynamische Fahrzeiten

Zur Abbildung von dynamisch-deterministischen Tourenplanungsproblemen werden vom Abreisezeitpunkt τ_i abhängige Fahrzeiten $t_{ij}(\tau_i)$ und Entfernungen $d_{ij}(\tau_i)$ benötigt. Im folgenden wird dargestellt, wie diese Fahrzeiten und Entfernungen ermittelt werden. Zur Erläuterung wird das FIFO-Prinzip benötigt, welches zunächst dargestellt wird. Anschließend wird ein Überblick über Modelle zur Abbildung von dynamischen Fahrzeiten gegeben, bevor schließlich die Darstellung der Vorgehensweise zur Ermittlung der dynamischen Fahrzeiten in einem Verkehrsnetzwerk, die in dieser Arbeit verwendet wird, folgt.

2.2.2.1 Das FIFO-Prinzip

Der Ankunftszeitpunkt bei Knoten n , der aus der Abfahrt bei Knoten m zum Zeitpunkt τ resultiert, wird mit $ankunft_{mn}(\tau)$ bezeichnet. Das FIFO-Prinzip besagt, daß für jeden beliebigen Pfeil (m, n) des Verkehrsnetzwerks und zwei beliebige Abfahrtszeitpunkte τ_1 und τ_2 gilt: $\tau_1 < \tau_2 \Rightarrow ankunft_{mn}(\tau_1) < ankunft_{mn}(\tau_2)$, d.h. daß „eine frühere Abfahrt bei $[m]$... auch immer eine frühere Ankunft bei $[n]$ garantieren

muss]“.⁷⁰ Für das dem dynamisch-deterministischen Tourenplanungsproblem (dTP) zugrunde liegende Verkehrswegenetzwerk wird angenommen, daß das FIFO-Prinzip, von Gietz (1994) als „Überholverbot“ bezeichnet, für alle Pfeile (m, n) des Verkehrswegenetzwerks gilt.

Die FIFO-Eigenschaft muss für alle Pfeile des Verkehrswegenetzwerks gelten, damit ausgehend vom Ankunftszeitpunkt bei dem Knoten n auf den erforderlichen Abreisezeitpunkt beim Knoten m geschlossen werden kann. Während für Tourenplanungsprobleme mit statischen Fahrzeiten das FIFO-Prinzip für alle Pfeile gilt, ist dessen Gültigkeit bei dynamischen Fahrzeiten nicht automatisch gewährleistet.

2.2.2.2 Der Ansatz von Ahn und Shin (1991) zur Abbildung variabler Fahrzeiten

Ahn und Shin (1991) modellieren dynamische Fahrzeiten mit Hilfe von abschnittsweise definierten Fahrzeitfunktionen zwischen einzelnen Kunden und zwischen Kunden und Depot.⁷¹ Diese Modellierung kann die Einhaltung des FIFO-Prinzips nicht für alle Fahrzeiten gewährleisten, wie folgendes Beispiel zeigt: Zwei Fahrzeuge fahren auf dem gleichen Weg von i zu j . Fahrzeug 1 verlässt i zum Zeitpunkt $\tau_1 = 19$, Fahrzeug 2 zum Zeitpunkt $\tau_2 = 21$. Die Fahrzeiten sind im Zeitablauf variabel und in folgender Funktion definiert:

$$t_{ij}(\tau) := \begin{cases} 10 \text{ [ZE]} & 0 \leq \tau < 20 \\ 5 \text{ [ZE]} & 20 \leq \tau < \infty \end{cases}$$

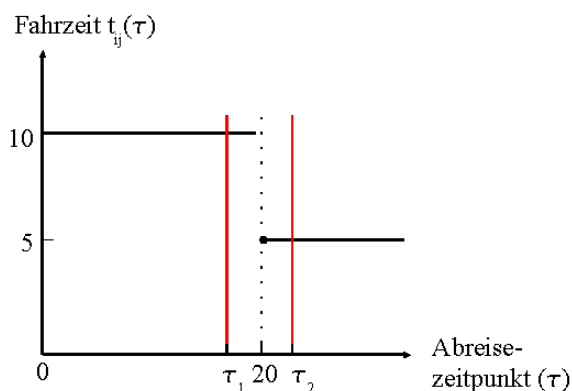


Abbildung 2.6: Abschnittsweise definierte Fahrzeitfunktion (vgl. Ichoua et al. (2003), S. 383)

Die Abreise bei m zum Zeitpunkt $\tau_2 := 21$ führt dazu, daß Fahrzeug 2 früher (zum

⁷⁰Gietz (1994), S. 155

⁷¹vgl. Ahn und Shin (1991)

Zeitpunkt $21 + 5 = 26$) bei n ankommt, als das Fahrzeug 1, das zum früheren Zeitpunkt $\tau_1 := 19$ abreist (Ankunft zum Zeitpunkt $19 + 10 = 29$). Inhaltlich bedeutet das, daß Fahrzeug B unterwegs Fahrzeug A überholt haben müsste. Wird von homogenen Fahrzeugen ausgegangen, ist ein solcher Überholvorgang nicht plausibel.

Gietz (1994) und Fleischmann et al. (2004) modellieren dynamische Fahrzeiten aufbauend auf dem Ansatz von Ahn und Shin (1991) und versuchen das FIFO-Prinzip dabei durch eine „Linearisierung“ der Fahrzeitfunktionen zu gewährleisten.⁷²

Die Fahrzeitfunktionen sind dabei jeweils für ein Paar von Knoten (i, j) (zwischen Kunden untereinander oder zwischen Kunden und Depot, nicht zwischen Kreuzungen) abschnittsweise definiert. Noch vor Beginn der Ermittlung von Lösungen zum Tourenplanungsproblem werden dazu die kürzesten Wege im Verkehrsnetzwerk zwischen allen Paaren von Kunden bestimmt.⁷³ Da allerdings die Wege damit vor Beginn der Tourenplanung festgelegt worden sind, können keine alternativen Routen zwischen Kunden berücksichtigt werden. Dieser Modellierungsansatz wird daher nicht näher betrachtet.

2.2.2.3 Der Ansatz von Ichoua et al. (2003) zur Abbildung variabler Fahrzeiten

Da eine Abbildung der Fahrzeiten zwischen einzelnen Kunden und zwischen Kunden und dem Depot durch eine sprungfix definierte Fahrzeitfunktion offenbar ungeeignet ist, schlagen Ichoua et al. (2003) folgende alternative Modellierung der Fahrzeiten für das dTP vor: Die Fahrzeiten werden in Abhängigkeit von der jeweils aktuellen Geschwindigkeit während der Fahrt von i nach j und der Entfernung zwischen i und j ermittelt. Die Geschwindigkeiten g_{ij} sind nach zeitlichen Intervallen abschnittsweise definiert, wobei der komplette Planungszeitraum $[e_0; l_0]$ in INT Intervalle unterteilt wird, die nicht notwendigerweise gleich lang sind (vgl. 2.1). Der Planungszeitraum wird mit dem Zeitfenster des Depots gleichgesetzt: e_0 bezeichnet den Zeitpunkt, zu dem das Depot frühestmöglich verlassen werden kann (üblicherweise wird $e_0 = 0$ gesetzt), l_0 den Zeitpunkt zu dem alle Fahrzeuge spätestens das Depot wieder erreicht haben müssen. Die Geschwindigkeit ist in jedem Intervall int konstant. τ_e^{int} bezeichnet den Beginn des aktuellen Intervalls, τ_l^{int} das dazugehörige Ende und τ den aktuellen Betrachtungszeitpunkt. Dabei muss gelten $\tau_e^0 = e_0$ (Das erste Intervall beginnt zum Beginn des Planungszeitraum.), $\tau_l^{INT} = l_0$ (Das letzte Intervall endet mit dem Ende des Planungszeitraum.) und $\tau_l^{int} = \tau_e^{int+1}$ (Das Ende jedes Intervalls $int \neq INT$ entspricht dem Beginn des nachfolgenden Intervalls).

⁷²vgl. Gietz (1994) S. 160-168; Fleischmann et al. (2004) S. 164-165; Fleischmann et al. (2004) fassen im Wesentlichen die Ergebnisse von Gietz (1994) auf Englisch zusammen.

⁷³vgl. Gietz (1994), S. 174

$$g_{ij} = \begin{cases} g_{ij}^0 & \text{für } \tau \in [\tau_e^0; \tau_l^0[\\ \vdots & \vdots \\ g_{ij}^{int} & \text{für } \tau \in [\tau_e^{int}; \tau_l^{int}[\\ \vdots & \vdots \\ g_{ij}^{INT} & \text{für } \tau \in [\tau_e^{INT}; \tau_l^{INT}] \end{cases} \quad (2.1)$$

Die Entfernung zwischen i und j wird von Ichoua et al. (2003) als konstant angesehen. Die Fahrzeit $t_{ij}(\tau_i)$ in Abhängigkeit vom Zeitpunkt, zu dem i verlassen wird, wird mit Hilfe der aktuellen Geschwindigkeit und der Distanz zwischen i und j errechnet. Die Anpassung der Geschwindigkeit bei Erreichen eines neuen Zeitintervalls erfolgt unverzüglich. Die sofortige Anpassung der Geschwindigkeit beim Übergang von einem Zeitintervall zum Nächsten ist realistischer als eine sofortige Anpassung der gesamten Fahrzeit auf dem gesamten Weg von i zu j , wie sie bei abschnittsweise definierten Fahrzeiten unterstellt wird.⁷⁴ Die Fahrzeiten bei der Fahrt von i zu j bei Abreise zum Zeitpunkt τ_i werden folgendermaßen berechnet:

Algorithmus 1 Ermittlung der Fahrzeiten in Abhängigkeit von der Geschwindigkeit

1. Initialisierung:

- a) aktuelles Intervall $int := \{int \mid \tau_i \in [\tau_e^{int}; \tau_l^{int}[\}$
- b) $\tau := \tau_i$
- c) $\delta := d_{ij}$
- d) $\tau' := \tau + \frac{\delta}{g_{ij}^{int}}$

2. Iteration: Wiederhole solange ($\tau' > \tau_l^{int}$)

- a) $\delta := \delta - g_{ij}^{int} \cdot (\tau_l^{int} - \tau)$
- b) $\tau := \tau_l^{int}$
- c) $int := int + 1$
- d) $\tau' := \tau + \frac{\delta}{g_{ij}^{int}}$

3. Ergebnis: Reisedauer $t_{ij}(\tau_i) := \tau' - \tau_i$

τ bezeichnet den aktuellen Zeitpunkt, τ' den voraussichtlichen Ankunftszeitpunkt bei j , δ die noch verbleibende Entfernung zwischen i und j . Zunächst wird das für den Abreisezeitpunkt τ_i aktuelle Intervall ermittelt (1a). Der aktuelle Zeitpunkt wird auf den Abreisezeitpunkt bei i gesetzt (1b). Es bleibt noch die komplette Entfernung d_{ij}

⁷⁴vgl. Ichoua et al. (2003), S. 385

zwischen den Kunden i und j zurückzulegen (1c). Wenn sich die Geschwindigkeit während der Fahrt von i nach j nicht ändert, kommt das Lieferfahrzeug zum Zeitpunkt τ' bei j an (1d).

Liegt τ' nach dem Ende des aktuellen Intervalls ($\tau' > \tau_l^{int}$), wird das Lieferfahrzeug ab dem Zeitpunkt τ_l^{int} die Geschwindigkeit anpassen. Die zum Zeitpunkt τ_l^{int} verbleibende Distanz δ wird ermittelt (2a) und der aktuelle Zeitpunkt auf das Ende des aktuellen Intervalls gesetzt (2b), das dem Beginn des nächsten, neuen aktuellen Intervalls entspricht (2c). Es wird ein neuer, voraussichtlicher Ankunftszeitpunkt τ' bestimmt (2d). Diese Vorgehensweise wird solange wiederholt, bis τ' innerhalb des aktuellen Intervalls int liegt (2).

Die Reisedauer $t_{ij}(\tau_i)$ ergibt sich aus der Differenz zwischen dem ermittelten Ankunftszeitpunkt τ' und dem Abreisezeitpunkt τ_i (3). Durch diese Modellierung wird das FIFO-Prinzip immer eingehalten.⁷⁵

Ichoua et al. (2003) lassen eine Verletzung der Zeitfensterrestriktionen für Kundenaufträge zu, bei einer verspäteten Anlieferung fallen Strafkosten an.⁷⁶ Für das Depot ist das Zeitfenster bindend.⁷⁷ Es werden keine Geschwindigkeiten nach l_0 definiert.

Das folgende Beispiel zeigt die Ermittlung der Fahrzeit basierend auf der Geschwindigkeitsfunktion zwischen i und j :

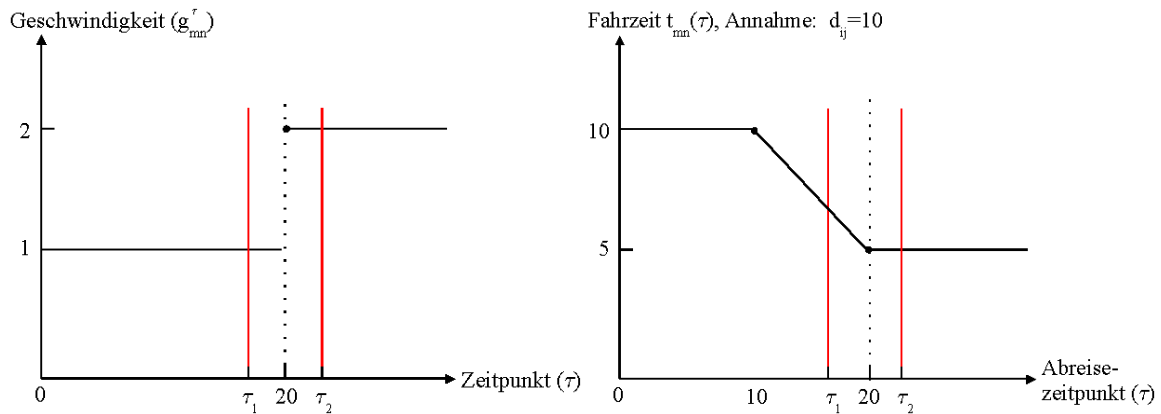


Abbildung 2.7: Fahrzeitfunktion: links: Geschwindigkeitsfunktion; rechts: daraus resultierende Fahrzeitfunktion (vgl. Ichoua et al. (2003), S. 383f.)

Die abschnittsweise definierte Geschwindigkeitsfunktion zwischen i und j (vgl. auch

⁷⁵vgl. Ichoua et al. (2003), S. 385

⁷⁶vgl. Ichoua et al. (2003), S. 386f.

⁷⁷vgl. Ichoua et al. (2003), S. 386f.

Abbildung 2.7 links) beträgt⁷⁸:

$$\begin{aligned} g_{ij}^{[0;20[} &:= 1 \text{ [EE/ZE]} \\ g_{ij}^{[20;\infty[} &:= 2 \text{ [EE/ZE]} \end{aligned}$$

und die Entfernung zwischen i und j sei:

$$d_{ij} := 10 \text{ [EE]}$$

Daraus ergibt sich die Fahrzeitfunktion, die in Abbildung 2.7 rechts dargestellt ist:

$$t_{ij}(\tau_i) := \begin{cases} 10 \text{ [ZE]} & \tau_i \leq 10 \\ (15 - 0,5 \cdot \tau_i) \text{ [ZE]} & 10 < \tau_i \leq 20 \\ 5 \text{ [ZE]} & \tau_i > 20 \end{cases}$$

Verlässt Fahrzeug A zum Zeitpunkt $\tau_2 := 21$ den Kunden i mit einer Geschwindigkeit von 2 EE/ZE, kommt es zum Zeitpunkt 26 bei j an (Fahrzeit 5). Ein Fahrzeug B , das zum früheren Zeitpunkt $\tau_1 := 19$ bei i mit einer Geschwindigkeit von 1 abreist, fährt ab Zeitpunkt 20 ebenfalls mit Geschwindigkeit 2, und kommt zum Zeitpunkt 24,5 bei j an (Fahrzeit 5,5).⁷⁹ Fahrzeug B hat zum Zeitpunkt 20 bereits einen Teil der Strecke zurückgelegt, es wird daher in jedem Fall vor Fahrzeug A bei j ankommen, wenn der gleiche Weg gewählt wird.

Der Modellierungsansatz von Ichoua et al. (2003) berücksichtigt keine alternativen Wege zwischen zwei Kunden i und j .⁸⁰ Vielmehr wird für jedes Paar von Knoten i und j mit $i, j \in C_0$ vor Beginn der Tourenplanung ein Weg festgelegt. Die durchschnittlichen Geschwindigkeiten im Zeitablauf auf diesem Weg werden geschätzt. In der Realität existieren häufig in Abhängigkeit von der Tageszeit unterschiedliche vorteilhafte Wege, die auch unterschiedlich lang (unterschiedliche Distanzen) sein können. Alternative Wege lassen sich allerdings mit dem Ansatz von Ichoua et al. (2003) nicht abbilden, da sowohl Entfernungen als auch Geschwindigkeitsfunktionen auf Paare von Knoten $i, j \in C_0$ (nur Kunden und Depot) definiert sind.

⁷⁸EE = Entfernungseinheiten; ZE=Zeiteinheiten

⁷⁹Zum Zeitpunkt 20 hat A bereits 1 EE zurückgelegt. Die restliche Strecke von 9 EE wird mit 2 EE/ZE zurückgelegt. Daraus resultiert: Ankunft zum Zeitpunkt 24,5.

⁸⁰vgl. Fleischmann et al. (2004), S. 161

2.2.2.4 Neuer Ansatz zur Abbildung von dynamischen Fahrzeiten bei alternativen Wegen in Verkehrsnetzwerken

Die Modelle zum dTP, die bisher in der Literatur vorgestellt wurden, bilden reale Verkehrsnetzwerke nur unzureichend ab. In der Regel werden die Entfernungen zwischen den einzelnen Kunden und zwischen den Kunden und dem Depot auf Basis von euklidischen Distanzen approximiert und die Fahrzeiten dann mithilfe von Multiplikatoren geschätzt (vgl. Ichoua et al. (2003)). Wird mit realen Daten gearbeitet, werden die kürzesten Wege und die dazugehörigen Fahrzeiten zwischen allen Kunden vor Beginn der Planung bestimmt. Das den Untersuchungen zugrunde liegende Netzwerk steht dann für Vergleichszwecke nicht zur Verfügung (vgl. Gietz (1994), Donati et al. (2003a)). Fleischmann (1997) weist zwar darauf hin, daß für eine sinnvolle Entfernungsbestimmung im Rahmen der Tourenplanung ein Verkehrsnetzwerk benötigt wird, welches neben Kunden und Depot auch „Kreuzungen“ enthält.⁸¹ Allerdings sind seiner Meinung nach die „eigentlich benötigten Daten [die Entfernungen] zwischen allen Kunden und zwischen Kunden und Depot.“⁸² Die kürzesten Wege sind dazu vor Beginn der Tourenplanung zu ermitteln und es ist dabei festzulegen, ob die geographisch kürzesten oder die schnellsten Wege ermittelt werden sollen.⁸³ Die Kunden und das Depot sind die einzigen Knoten im Netzwerk, auf dem die Planung basiert. Die Modellierung alternativer Wege im Verkehrsnetzwerk und kurzfristiger Störungen auf einzelnen Verbindungen ist damit nicht möglich. Somit ist auch jede „Reaktionsfähigkeit der Steuerung auf dieses Abstraktionslevel eingeengt.“⁸⁴, also eine Anpassung der Wege als Reaktion auf Störungen nicht vorgesehen. In dieser Arbeit wird ein neuer Ansatz zur Bestimmung der Fahrzeiten und Entfernungen im Verkehrsnetzwerk, unter Berücksichtigung von dynamischen Fahrzeiten und alternativen Wegen im Zeitablauf vorgestellt.

Bock (2004) schlägt ein Modell zur echtzeitfähigen Steuerung von Transportnetzwerken auf Basis von Verkehrsnetzwerken vor.⁸⁵ Zwar ermöglicht sein Ansatz alternative Wege abzubilden, es fehlt aber die Möglichkeit, Prognosen über die zu erwartenden Fahrzeiten bereits im Rahmen der Planung zu berücksichtigen. In seinem Ansatz werden variable Fahrzeiten nicht in die Planung einbezogen, sondern als mögliches, zufällig eintretendes Störereignis mit unbekannter Dauer modelliert.⁸⁶

Werden Tourenplanungsprobleme auf Basis von Verkehrsnetzwerken modelliert, welche neben dem Depot und den Kundenstandorten weitere Knoten (im

⁸¹vgl. Fleischmann (1997) S. 289

⁸²Fleischmann (1997) S. 290

⁸³vgl. Fleischmann (1997) S. 290

⁸⁴Bock (2004), S. 91

⁸⁵vgl. Bock (2004), S. 165-232

⁸⁶vgl. Bock (2004), S. 267

folgenden als Kreuzungen bezeichnet) umfassen, können alternative Wege zwischen Kunden im Rahmen der Tourenplanung und -steuerung berücksichtigt werden.⁸⁷ Verkehrswegenetze ermöglichen es weiterhin, asymmetrische Geschwindigkeiten (z.B. hohes Verkehrsaufkommen zu bestimmten Zeiten nur in einer Fahrtrichtung) und asymmetrische Entfernungen (z.B. Einbahnstraßen) abzubilden.

Erst wenn unterschiedliche Wege im Verkehrswegenetzwerk zur Verfügung stehen, ist es möglich Reaktionen auf Störungen im Verkehrswegenetzwerk, die erst nach Abschluss der Planungen bekannt werden, abzubilden. Der Einfluss stochastischer Ereignisse auf die ermittelten Lösungen wird in der Literatur zur Tourenplanung mit Zeitfensterrestriktionen und dynamischen Fahrzeiten bislang nicht untersucht.

Ermittlung von Wegen in dynamischen Netzwerken

In dieser Arbeit wird ein neuer Ansatz zur Abbildung von variablen Fahrzeiten im Rahmen der Tourenplanung vorgeschlagen. Der Ansatz zur Modellierung von variablen Fahrzeiten zwischen Kundenstandorten von Ichoua et al. (2003) wird zur Bestimmung der Fahrzeiten auf jedem einzelnen Pfeil (m, n) des Verkehrswegenetzwerks übernommen.

Um die Fahrzeiten zwischen zwei Kunden und die dazugehörigen Wege zu ermitteln, wird auf ein modifiziertes Verfahren von Dijkstra zurückgegriffen, welches denjenigen Weg im Netzwerk zwischen zwei Kunden ermittelt, der die Fahrzeiten minimiert.⁸⁸ Damit dieses Verfahren angewandt werden kann wird angenommen, daß sowohl für die Entfernungen als auch für die Fahrzeiten im Verkehrswegenetzwerk folgendes gilt: Jeder Weg zwischen einem beliebigen Knotenpaar auf dem kürzesten Weg eines gegebenen Netzwerks ist wiederum ein kürzester Weg.⁸⁹ Weiterhin wird angenommen, daß die Dreiecksungleichung für alle Pfeile zwischen allen Knoten des Verkehrswegenetzwerks gilt. Die Dreiecksungleichung besagt, daß der direkte Weg zwischen zwei Knoten immer höchstens genauso lang ist, wie jeder Weg, der über einen weiteren Knoten zum Zielknoten führt. Wird der distanzminimale Weg zwischen zwei Knoten gesucht, läßt er sich mit Hilfe eines Dijkstra-Algorithmus optimal ermitteln. Auch für den schnellsten Weg zwischen zwei Knoten läßt sich diese Aussage treffen, denn sollte der schnellste Weg über einen dritten Knoten eine geringere Fahrzeit erfordern als der direkte Weg, so kann der direkte Weg durch den Weg über den dritten Knoten ersetzt

⁸⁷vgl. Bock (2004) S. 89-91

⁸⁸Zhan und Noon (1998) empfehlen den Dijkstra-Algorithmus für die Bestimmung schnellster Wege zwischen zwei Knoten (vgl. Zhan und Noon (1998), S. 77). Vorteil des Dijkstra-Algorithmus (vgl. Dijkstra (1959)) ist die Möglichkeit, die Suche abzubrechen, sobald ein Weg zum Zielkunden gefunden wurde. Auch Bock (2004) greift auf einen Dijkstra-Algorithmus im Rahmen seines Modells zurück (vgl. Bock (2004), S. 244).

⁸⁹vgl. Miller-Hooks und Mahmassani (2000), S. 200

werden. Ein Problem ergibt sich wenn allerdings davon ausgegangen wird, daß sich die entscheidungsrelevanten Kosten der zurückzulegenden Wege als gewichtete Funktion aus Fahrzeiten und Entfernungen ergeben. Dann ist die Dreiecksungleichung nicht länger anwendbar, denn der schnellste Weg und der distanzminimale Weg können sich unterscheiden. Dies soll anhand des folgenden Beispiels gezeigt werden:

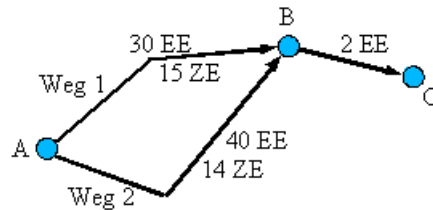


Abbildung 2.8: Beispiel zur Ermittlung von kostenminimalen Wegen

Die Entscheidungssituation ist in Abbildung 2.8 dargestellt. Es existieren zwei alternative Wege zwischen den Knoten A und B. Die Entfernung zwischen A und B beträgt 30 Entfernungseinheiten (EE) auf Weg 1 und $40EE$ auf Weg 2. Die Fahrzeiten entsprechen, unabhängig vom Abreisezeitpunkt, 15 Zeiteinheiten (ZE) auf Weg 1 und $14ZE$ auf Weg 2. Die Entfernung zwischen B und C beträgt $2EE$, die Fahrzeit ist abhängig vom Abreisezeitpunkt bei B (τ_B). Gilt $\tau_B < 15$, beträgt die Geschwindigkeit von B zu C $2EE/ZE$, für $\tau_B \geq 15$ beträgt die Geschwindigkeit $0,1EE/ZE$. Betragen die Kosten je ZE und je EE je eine Geldeinheit (GE), ergibt sich folgendes Ergebnis: der kostenminimale Weg von A zu B ist Weg 1. Dabei fallen Kosten von $30EE \cdot 1 \frac{GE}{EE} + 15ZE \cdot 1 \frac{GE}{ZE} = 45GE$ an. Der kostenminimale Weg von A zu C führt hingegen über Weg 2, wie folgende Rechnung zeigt:

Kosten bei Wahl von Weg 1:

$$\begin{aligned} & (30 + 2) \frac{GE}{EE} + (15ZE + 2EE/0,1 \frac{EE}{ZE}) \cdot 1 \frac{GE}{ZE} \\ &= (32 + 35)GE \\ &= 67GE \end{aligned}$$

Kosten bei Wahl von Weg 2:

$$\begin{aligned} & (40 + 2) \frac{GE}{EE} + (14ZE + 2EE/2 \frac{EE}{ZE}) \cdot 1 \frac{GE}{ZE} \\ &= (42 + 15)GE \\ &= 57GE \end{aligned}$$

Es gilt also nicht, daß jeder Weg zwischen einem beliebigen Knotenpaar auf dem kostenminimalen Weg zwischen zwei Knoten auch kostenminimal ist.

Da der Schwerpunkt dieser Arbeit auf dynamischen Tourenplanungsproblemen mit zeitabhängigen Zielen und nicht auf der Ermittlung von kostenminimalen Wegen in Verkehrswegenetzwerken liegt, wird für die Ermittlung von Wegen im Verkehrswegenetzwerk aus Vereinfachungsgründen immer der fahrzeitminimale (=schnellste) Weg zugrunde gelegt. Zur Ermittlung der schnellsten Wege im dynamischen Verkehrswegenetzwerk wird ein modifizierter Dijkstra-Algorithmus verwendet. Auf die Ermittlung kostenminimaler Wege in Verkehrswegenetzwerken wird nicht näher eingegangen.⁹⁰

Die Fahrzeit $t_{ij}(\tau_i)$ zwischen beliebigen Knoten $i, j \in C_0$ ist vom Abreisezeitpunkt τ_i abhängig. Sie wird als Funktion der zurückzulegenden Entfernung zwischen den Knoten i und j und der jeweils aktuellen Geschwindigkeit auf dem schnellsten Weg, in Anlehnung an Ichoua et al. (2003) modelliert. Auch die Entfernung $d_{ij}(\tau_i)$ zwischen zwei Kunden i und j ist dabei im Zeitablauf nicht notwendigerweise konstant. d_{mn} , die Entfernung zwischen zwei Knoten im Verkehrswegenetz ist im Zeitablauf immer gleich. Der schnellste Weg von i nach j kann sich aber in Abhängigkeit von τ_i ändern, und somit die Entfernung. Die Fahrzeit $t_{mn}(\tau_m)$ zwischen zwei Knoten m und n bei Verlassen des Knotens m zum Zeitpunkt τ_m wird analog zu Algorithmus 1 ermittelt, wobei allerdings i durch m und j durch n ersetzt wird (vgl. Algorithmus 2).

Algorithmus 2 Ermittlung der Fahrzeiten in Abhängigkeit von der Geschwindigkeit

1. Initialisierung:

- a) aktuelles Intervall $int := \{int \mid \tau_m \in [\tau_e^{int}, \tau_l^{int}[\}$
- b) $\tau := \tau_m$
- c) $\delta := d_{mn}$
- d) $\tau' := \tau + \frac{\delta}{g_{mn}^{int}}$

2. Iteration: Wiederhole solange ($\tau' > \tau_l^{int}$)

- a) $\delta := \delta - g_{mn}^{int} \cdot (\tau_l^{int} - \tau)$
- b) $\tau := \tau_l^{int}$
- c) $int := int + 1$
- d) $\tau' := \tau + \frac{\delta}{g_{mn}^{int}}$

3. Ergebnis: Reisedauer $t_{mn}(\tau_m) := \tau' - \tau_m$

Zunächst wird das aktuelle Intervall int gesucht (1a). Es wird der aktuelle Zeitpunkt τ gleich dem Abreisezeitpunkt beim Knoten m gesetzt (1b). Die noch verbleibende

⁹⁰Miller-Hooks und Mahmassani (2000) weisen auf S. 199f. darauf hin, daß die Fahrzeit durch andere Kosten ersetzt werden kann, allerdings nicht wie.

Entfernung δ entspricht der Distanz zwischen m und n (1c). Der voraussichtliche Ankunftszeitpunkt τ' beim Knoten n , wenn während der Fahrt von m zu n keine Geschwindigkeitsänderung eintritt, wird ermittelt (1d).

Solange der erwartete Ankunftszeitpunkt τ' nicht in das aktuelle Intervall fällt (2), wird die nach dem aktuellen Intervall noch zurückzulegende Entfernung δ bestimmt (2a). Der neue, aktuelle Zeitpunkt τ wird auf das aktuelle Intervallende τ_l^{int} gesetzt (2b). Das nachfolgende Intervall wird jetzt zum aktuellen Intervall (2c) und ein neuer voraussichtlicher Ankunftszeitpunkt τ' auf Basis der neuen, aktuellen Geschwindigkeit ermittelt (2d).

Die Fahrzeit ergibt sich aus der Differenz aus Ankunftszeitpunkt τ' und Abreisezeitpunkt τ_m (3).

Wird die Fahrzeit $t_{ij}(\tau_i)$ von einem Kunden i zu einem anderen Kunden j bei Abreise zum Zeitpunkt τ_i gesucht, lässt sich der schnellste Weg, die dafür erforderliche Fahrzeit und die Länge des Wegs mit Hilfe eines modifizierten Verfahrens von Dijkstra exakt ermitteln:

Algorithmus 3 Modifiziertes Verfahren von Dijkstra zur Ermittlung des schnellsten Wegs vom Kunden i zum Kunden j in dynamischen Verkehrsnetzwerken

1. Initialisierung:

- a) $\Psi_1 := \emptyset$
- b) $\Psi_2 := \emptyset$
- c) $faz_\psi := +\infty \forall \psi \in N$
- d) $m := i; faz_m := \tau_i; vor_m := i$
- e) $\tau' := \tau_i$
- f) füge m in Ψ_2 ein

2. wenn $j \in \Psi_2$ gehe zu (4)

- a) für alle $n \in \Gamma^+(m)$, für die gilt $n \notin \Psi_2$
 - i. Wenn $n \notin \Psi_1$ dann $\Psi_1 := \Psi_1 \cup \{n\}$
 - ii. $neu := n; faz_{neu} := t_{mn}(\tau')$; $vor_{neu} := m$
 - iii. Wenn $faz_{neu} < faz_n$ dann $faz_n := faz_{neu}$; $vor_n := vor_{neu}$
- b) setze $m := u$ und $\tau' := faz_u$ für u mit $faz_u = \min\{faz_\psi | \psi \in \Psi_1\}$
- c) Wenn $faz_u = +\infty$: abbrechen
- d) $\Psi_1 := \Psi_1 \setminus \{m\}$ und $\Psi_2 := \Psi_2 \cup \{m\}$

3. gehe zu (2)

4. Ergebnis: Ankunftszeitpunkt in j ist faz_j ; Reisedauer $t_{ij}(\tau_i) = faz_j - \tau_i$

Zunächst sind die Menge der Kandidaten-Knoten Ψ_1 (1a) und die Menge der Knoten, zu denen der schnellste Weg bereits bekannt ist Ψ_2 (1b) leer. Die frühesten Ankunftszeitpunkte faz_ψ bei allen Knoten des Verkehrsnetzwerks ($\psi \in N$) werden mit $+\infty$ initialisiert (1c). Die Suche nach dem schnellsten Weg beginnt im Knoten i , der früheste Ankunftszeitpunkt wird für i dem Abreisezeitpunkt τ_i gleichgesetzt und der Vorgänger auf dem schnellsten Weg von i zu i wird auf i gesetzt (1d). Der aktuelle Zeitpunkt τ' wird auf den Abreisezeitpunkt τ_i gesetzt (1e). Der Ausgangsknoten i wird anschließend in die Menge Ψ_2 eingefügt (1f).

Solange der Zielknoten j nicht in Ψ_2 enthalten ist, ist der schnellste Weg zu j noch nicht bekannt und die Suche nach dem schnellsten Weg von i zu j wird fortgesetzt (2). Alle Knoten n , die direkt mit m verbunden sind ($n \in \Gamma^+(m)$) und zu denen der schnellste Weg noch nicht bekannt ist ($n \notin \Psi_2$) (2a), werden in die Menge der Kandidatenknoten Ψ_1 eingefügt, sofern sie noch nicht enthalten sind (2a-i.). Dann wird der Ankunftszeitpunkt bei n ermittelt, der sich ergibt, wenn der Weg über m führt (2a-ii.). Es wird geprüft, ob der Weg über den aktuellen Knoten m zu n schneller ist, als der schnellste bekannte Weg zu n . Ist dies der Fall, wird n in Ψ_1 aktualisiert (2a-iii.). Der nächste zu untersuchende Knoten ist derjenige Knoten aus der Menge der Kandidatenknoten Ψ_1 , der als schnellstes zu erreichen ist, für den faz_m minimal ist (2b). Sollte kein Knoten mehr innerhalb des Planungszeitraums erreicht werden können, sind die Geschwindigkeitsfunktion g_{mn}^{int} und $t_{mn}(\tau')$ und dadurch auch faz_u nicht definiert (2c). In diesem Fall kann der Knoten j nicht erreicht werden. Die Suche nach dem schnellsten Weg wird dann ergebnislos beendet. Ansonsten wird der Knoten m in die Menge Ψ_2 aufgenommen und aus Ψ_1 entfernt (2d). Handelt es sich bei m um den Knoten j , ist der schnellste Weg von i zu j bekannt. Der Knoten j wird dann in die Menge Ψ_2 aufgenommen (2d) und die Suche anschließend beendet (2).

Die Fahrzeit auf dem schnellsten Weg von i nach j beträgt $faz_j - \tau_i$ (4). Die Entfernung $d_{ij}(\tau_i)$ wird nach der Berechnung von $t_{ij}(\tau_i)$ ermittelt, indem rekursiv mit Hilfe von vor_m ausgehend von j die jeweiligen Entfernungen d_{mn} auf dem schnellsten Weg aufsummiert werden.

2.2.3 Zielsetzungen für dynamisch-deterministische Tourenplanungsprobleme

2.2.3.1 Mögliche Zielsetzungen und Zielkonflikte

Ziel der Tourenplanung ist die Belieferung aller Kunden innerhalb ihrer Zeitfenster zu möglichst geringen Kosten. Allerdings ist es äußerst schwierig, die Kosten der Tourenplanung verursachungsgerecht zu operationalisieren. Es existieren eine Reihe

von Ersatzzielen für Tourenplanungsprobleme, welche anstelle der ökonomischen Zielsetzung "Minimierung der Belieferungskosten" in der Literatur Verwendung gefunden haben, allerdings ohne daß sie vor ihrer Verwendung in der Literatur näher untersucht wurden. Üblicherweise wird anstelle von einer Kostenfunktion eine lexikographische Zielordnung von Ersatzzielen verwendet.⁹¹ Zunächst wird in der Regel die Anzahl der Touren minimiert, anschließend wird dann aus der Menge der Lösungen mit minimaler Tourenzahl diejenige ausgewählt, welche die kürzeste Gesamtentfernung⁹², die geringste Summe der Fahrzeiten⁹³ oder die geringste Bediendauer⁹⁴ aufweist. In der Literatur zu dynamischen Tourenplanungsproblemen ist die Minimierung der Summe der Fahrzeiten üblich.⁹⁵

Ein Vorteil von lexikographischen Zielsetzungen ist, daß keine Kostenfunktionen für die einzelnen Kosteneinflussfaktoren bestimmt werden müssen. Lösungen aus verschiedenen Untersuchungen sind miteinander vergleichbar, ohne daß die gleichen Kostenkoeffizienten verwendet werden müssen oder sie auch nur bekannt sein müssen.

Zunächst werden im folgenden einige gängige Ersatzziele für Tourenplanungsprobleme und die damit verbundenen Kosten dargestellt. Anschließend werden mögliche Zielkonflikte zwischen den Ersatzzielen kurz beispielhaft aufgeführt. Die gesamten Auslieferungskosten können mit dem Fahrzeugeinsatz verbundene, sprungfixe Kosten, entfernungsabhängige Kosten, fahrzeitabhängige Kosten, von der gesamten Bediendauer abhängige Kosten und sonstige, im folgenden nicht näher betrachtete Kosten umfassen.

Minimiere die Anzahl der Touren

Wird davon ausgegangen, daß mit dem Einsatz eines Fahrzeugs sprungfixe Kosten entstehen, führt die Minimierung der Anzahl der Touren zu sinkenden Kosten. Kosten, die von der Anzahl der durchzuführenden Touren abhängen, sind beispielsweise:

- Anteil der Entlohnung der Fahrer, der pauschal pro Tour oder pro Arbeitstag anfällt
- Bereitstellungskosten für das Fahrzeug am Depot
- Mietfixkosten, wenn die Lieferfahrzeuge tageweise angemietet werden

⁹¹vgl. beispielsweise Savelsbergh (1992), S. 150; Potvin et al. (1996), S. 158; Liu und Shen (1999), S. 486;

⁹²vgl. beispielsweise Homberger und Gehring (2005), S. 220

⁹³vgl. beispielsweise Gambardella et al. (1999), S. 2

⁹⁴vgl. beispielsweise Solomon (1987) S. 259, Potvin und Rousseau (1995), S. 1437

⁹⁵vgl. Donati et al. (2003a), S. 11 und S. 13;

Es ist zu beachten, daß die Tourenplanung in der Regel einen Planungshorizont von einem Tag aufweist. Die mit dem Kauf von Lieferfahrzeugen verbundenen Kosten sind daher nicht entscheidungsrelevant.

Im folgenden wird der Bezeichner z sowohl für eine Tour als auch für das Lieferfahrzeug, welches die Tour durchführt, verwendet. Es wird nicht zwischen Lieferfahrzeug und Tour unterschieden, da ein homogener Fuhrpark unterstellt wird und jedes Lieferfahrzeug im Planungszeitraum nur eine Tour durchführen darf.⁹⁶ ζ_z bezeichnet eine binäre Entscheidungsvariable. Gilt $\zeta_z = 1$, wird das Lieferfahrzeug z eingesetzt. Z bezeichnet den Fuhrpark, also die Menge aller Fahrzeuge, die für die Tourenplanung zur Verfügung stehen. Die Anzahl der Touren eines Tourenplans ergibt sich somit zu $\sum_{z \in Z} \zeta_z$.

Minimiere die Gesamtentfernung

Es kann davon ausgegangen werden, daß die Kosten der Auslieferung mit zunehmender Entfernung steigen. Beispiele:

- streckenabhängiger Benzinverbrauch
- streckenabhängiger Verschleiß
- Kilometerpreise bei Mietfahrzeugen, wenn die Lieferfahrzeuge angemietet werden

$d_{ij}(\tau_i)$ bezeichnet die Entfernung die bei der Fahrt vom Knoten i zum Knoten j bei Verlassen von i zum Zeitpunkt τ_i zurückzulegen ist. Die binäre Entscheidungsvariable x_{ij}^z ist dann gleich 1, wenn die Fahrt von i nach j mit Fahrzeug z durchgeführt wird. Die Gesamtentfernung eines Tourenplans ergibt sich zu $\sum_{z \in Z} \sum_{i \in C_0} \sum_{j \in C_0} d_{ij} \cdot x_{ij}^z$.

Die streckenabhängigen Kosten sind sicherlich nicht ausreichend um die gesamten Auslieferungskosten zu approximieren, da neben der zurückgelegten Entfernung auch zeitliche Größen Einfluss auf die Auslieferungskosten haben.

Minimiere die Summe der Fahrzeiten

Für dynamische Probleme wird üblicherweise die Summe der Fahrzeiten minimiert.⁹⁷ Diese Zielsetzung erscheint zunächst sinnvoll. Implizit wird angenommen, daß nur dann Kosten anfallen, wenn das Lieferfahrzeug unterwegs ist. Allerdings wird dadurch auch unterstellt, daß keine Kosten während der Warte- und Servicezeiten entstehen. Es ist

⁹⁶vgl. die Problemdefinition in Abschnitt 2.2.4

⁹⁷vgl. Donati et al. (2003a), S. 11

nicht möglich, Kosten zu identifizieren, die ausschließlich von der bei der Auslieferung anfallenden Fahrzeit abhängen. Vielmehr ist die gesamte Zeitspanne vom Verlassen des Depots bis zu Rückkehr im Depot zu betrachten.

Minimiere die gesamte Bediendauer

Mit sinkenden Fahrzeiten steigen häufig gleichzeitig die Wartezeiten. Das ist der Fall, wenn ein Lieferfahrzeug bei einem Kunden vor Beginn des Bedienungszeitfensters ankommt, um dann dort warten zu müssen, bis die Güter abgeladen werden können. Statt Fahrzeiten und Wartezeiten einzeln zu minimieren, erscheint es sinnvoll, beide Zeiten zugleich zu berücksichtigen. Die Bediendauer eines Fahrzeugs z (in der englischsprachigen Literatur als „schedule-time“ bezeichnet) entspricht der Zeitspanne vom Beginn des Planungszeitraums (dem frühestmöglichen Zeitpunkt, zu dem das Depot verlassen werden kann, im folgenden mit e_0 bezeichnet) bis zur Beendigung der Tour im Depot nach Abschluss der Bedienung aller Kunden und der Rückkehr des Fahrzeugs ins Depot, im folgenden mit y_0^z bezeichnet.

Mit y_i wird im folgenden der Zeitpunkt bezeichnet, zu dem mit der Bedienung des Kunden i begonnen wird, e_i ist der frühestmögliche Zeitpunkt zu dem mit der Bedienung begonnen werden kann (Beginn des Zeitfensters) und w_i die Wartezeit vor der Bedienung des Kunden i .

Ist h der direkt vor i zu bedienende Kunde und τ_h der Zeitpunkt, zu dem h verlassen wird, ergibt sich die Wartezeit w_i als $w_i := \min \{e_i - \tau_h - t_{hi}(\tau_h); 0\}$. Gilt $w_i \geq 0$ ist $y_i := e_i$, ansonsten gilt $y_i := \tau_h + t_{hi}(\tau_h)$. Der Zeitpunkt τ_i zu dem der Kunde i fertig bedient worden ist und wieder verlassen wird wird im folgenden als $\tau_i := y_i + s_i$ (der frühestmögliche Zeitpunkt, zu dem i wieder verlassen werden kann) festgelegt. Da jeder Kunde genau einmal bedient werden darf, sind für die Zeitpunkte y_i , die den Bedienungsbeginn definieren, die Zeitpunkte τ_i , zu denen der Kunde i wieder verlassen wird, und die Wartezeiten w_i kein Tourenindex z erforderlich.

Aus Übersichtlichkeitsgründen wird im folgenden immer davon ausgegangen, daß der Beginn des Planungszeitraums $e_0 = 0$ gilt, somit entspricht y_0^z der Bediendauer der Tour z . Die gesamte Bediendauer ($\sum_{z \in Z} y_0^z$) entspricht der Summe der Bediendauern über alle Fahrzeuge. Sie entspricht der Summe der Fahrzeiten zuzüglich der Summe der Wartezeiten und der Summe der Servicezeiten ($\sum_{z \in Z} \sum_{i \in C_0} \sum_{j \in C_0} t_{ij}(\tau_i) \cdot x_{ij}^z + \sum_{i \in C} w_i + \sum_{i \in C} s_i$). Die Servicezeiten s_i sind in dieser Arbeit nicht entscheidungsrelevant, da im folgenden angenommen wird, daß sie unabhängig von der Belieferungsreihenfolge sind und jeder Kunde im Planungszeitraum zu beliefern ist.⁹⁸

⁹⁸In der Realität kann die Servicedauer sehr wohl von der Belieferungsreihenfolge abhängig sein, je nachdem wie die auszuliefernden Güter im Fahrzeug verstaut werden.

Die gesamte Bediendauer ist eine relativ gute Approximation für die Belieferungskosten. Beispiele:

- Anteil der Entlohnung der Fahrer, der nach geleisteter Arbeitszeit bezahlt wird
- Mietpreise in Abhängigkeit der Mietdauer, wenn die Lieferfahrzeuge angemietet werden und die Miete nicht pauschal auf Tagesbasis abgerechnet wird

Von Donati et al. (2003a) wird argumentiert, daß durch das Verlassen des Depots zu einem späteren Zeitpunkt als dem Beginn des Planungszeitraums die Summe der Fahrzeiten geringer ausfallen kann als bei sofortigem Beginn der Auslieferung.⁹⁹ Mit dem späteren Verlassen verbundene Einsparungen bei der Fahrzeit führen allerdings bei gegebener Lieferreihenfolge zu ansteigenden Wartezeiten in gleicher Höhe vor Beginn der Auslieferung im Depot. Zudem wird das Depot erst später wieder erreicht. Es ist vorstellbar, die Bediendauer als Zeitspanne ab dem Verlassen des Depots zu definieren, dies erscheint allerdings nicht zweckmäßig. Wird angenommen, daß die Tourenplanung kurzfristig erfolgt und das für die Auslieferung benötigte Personal nicht täglich zu unterschiedlichen Zeitpunkten die Arbeit aufnimmt, erscheint es wenig sinnvoll einen anderen Zeitpunkt für das Verlassen des Depots zuzulassen als den Beginn des Planungszeitraums e_0 .

Zielkonflikte zwischen ausgewählten Ersatzzielen

Die Entfernung zwischen zwei Kunden i und j (d_{ij}) und die dazugehörige Fahrzeit (t_{ij}) werden für statische Tourenplanungsprobleme in der Regel gleichgesetzt ($d_{ij} \hat{=} t_{ij}$).¹⁰⁰ Selbst bei statischen Tourenplanungsproblemen ist diese Vereinfachung allerdings nur genau dann zulässig, wenn man annimmt, daß das Verhältnis von Entfernungen und Fahrzeiten für alle Verbindungen zwischen den Kunden untereinander sowie zwischen den Kunden und dem Depot gleich ist. Im folgenden Abschnitt soll dargestellt werden, weshalb diese Annahme nicht unproblematisch ist.

Beispiel 1: Zielkonflikt Gesamtentfernung und gesamte Fahrzeit Zwischen den Zielen minimiere Gesamtentfernung ($\min dist(X)$) und minimiere gesamte Fahrzeit ($\min travel(X)$) kann ein Konflikt bestehen: Ist das Verhältnis von Entfernungen zu Fahrzeiten nicht für alle Pfeile gleich, kann ein räumlich längerer Weg zwischen zwei Kunden weniger Zeit erfordern als ein kürzerer, direkter Weg.

In Abbildung (2.9) sind der schnellste und der kürzeste Weg vom Kunden i zum Kunden j beispielhaft dargestellt.

⁹⁹vgl. Donati et al. (2003a), S. 13

¹⁰⁰vgl. Solomon (1987), S. 255f.

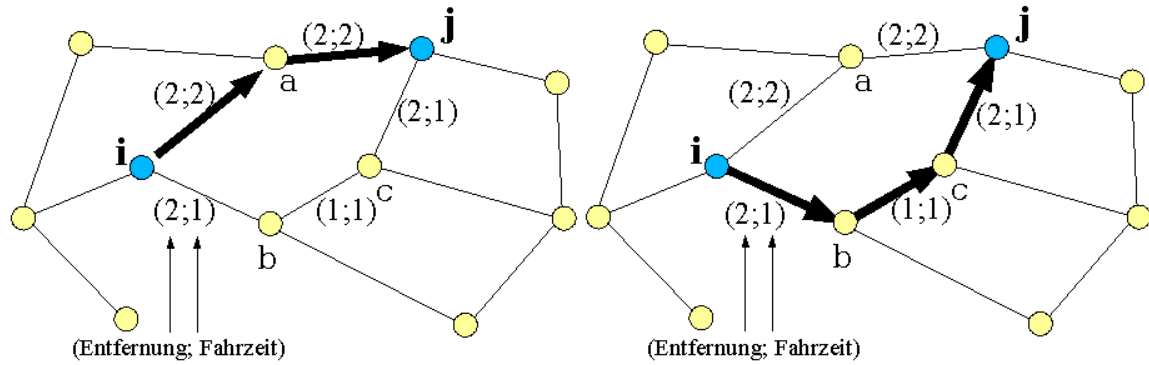


Abbildung 2.9: Beispiel 1: Zielkonflikt zwischen Minimierung der Entfernung und Minimierung der Fahrzeit

Der Weg 1 (i, a, j) hat zwar eine niedrigere Entfernung als Weg 2 (i, b, c, j), ($d^1 = 4$ Entfernungseinheiten (EE) vs. $d^2 = 5$ EE), Weg 2 ist dennoch schneller ($t^1 = 4$ Zeiteinheiten (ZE) vs. $t^2 = 3$ ZE). Dieser Konflikt lässt sich in der Realität häufig beobachten: Ein „Umweg“ über die Autobahn führt häufig schneller zum Ziel als ein direkter Weg über Nebenstraßen.

Beispiel 2: Zielkonflikt Gesamtentfernung und Anzahl Touren Einen weiteren möglichen Zielkonflikt zwischen der Anzahl der Touren und der zurückzulegenden Entfernung verdeutlicht das Beispiel in Tabelle 2.1. Gegeben sei folgendes VRPTW mit acht Kunden und einem Depot. Mit x und y sind die Koordinatenwerte in der Ebene bezeichnet, q_i ist die Nachfrage des Kunden i , ausgedrückt in Kapazitätseinheiten. Die Entfernungen zwischen den Kunden entsprechen den euklidischen Distanzen. Die Fahrzeugkapazität Q beträgt 45 Kapazitätseinheiten.

i	Depot	1	2	3	4	5	6	7	8
q_i		20	20	25	5	5	5	5	5
x	10	1	3	3	14	15	17	18	21
y	10	16	20	1	14	9	16	10	14

Tabelle 2.1: Beispiel 2: Gesamtentfernung vs. Anzahl Touren

Die Lösung mit der geringsten Zahl an Touren $X^1 = \{(0, 1, 2, 4, 0), (0, 6, 8, 7, 5, 3, 0)\}$, (zwei Touren, Gesamtentfernung $dist(X^1) \approx 81EE$, in Abbildung 2.10 links) entspricht offensichtlich nicht der Lösung mit der geringsten Gesamtentfernung $X^2 = \{(0, 1, 2, 0), (0, 4, 6, 8, 7, 5, 0), (0, 3, 0)\}$, (drei Touren, Gesamtentfernung $dist(X^2) \approx 77EE$, in Abbildung 2.10 rechts).¹⁰¹

¹⁰¹Die jeweils besten Lösungen wurden für die folgenden Beispiele jeweils durch vollständige Enumeration ermittelt.

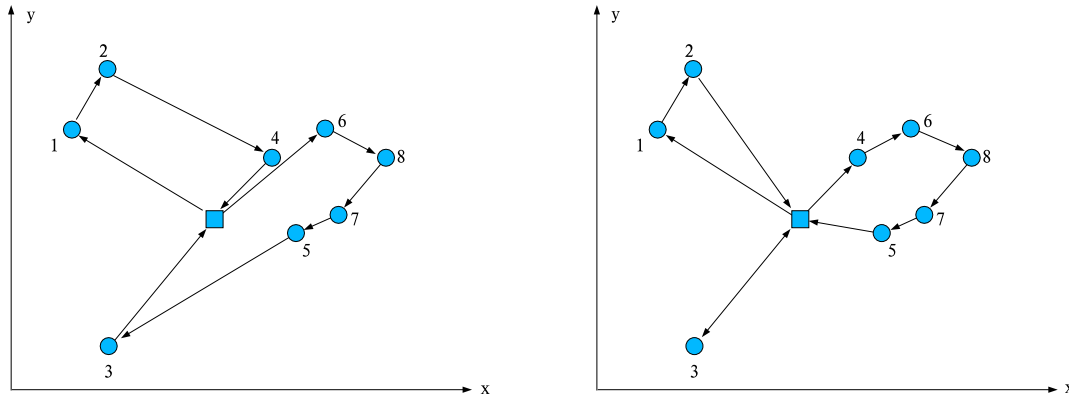


Abbildung 2.10: Beispiel 2: links: Lösung mit minimaler Tourenzahl, rechts: Lösung mit minimaler Gesamtentfernung

Beispiel 3: Zielkonflikt Gesamtentfernung und Bediendauer Das Beispiel in Tabelle 2.2 zeigt einen möglichen Zielkonflikt zwischen den beiden Zielen minimiere Gesamtentfernung ($dist(X)$) und minimiere Bediendauer ($sched(X)$), wenn Zeitfensterrestriktionen zu berücksichtigen sind. Die Entfernungen zwischen den Kunden entsprechen wiederum den euklidischen Distanzen, die Fahrzeiten entsprechen konstant einer Zeiteinheit je Entfernungseinheit. Mit e_i wird der frühestmögliche Beginn der Bedienung des Kunden i bezeichnet, l_i ist der spätestmögliche Beginn der Bedienung. s_i ist die Servicedauer beim Kunden i .

i	Depot	1	2	3	4	5	6	7	8
q_i		5	5	5	5	5	5	5	5
x	10	2	7	8	12	15	18	19	17
y	10	10	7	12	13	15	14	11	8
e_i	0	25	70	8	4	18	32	46	95
l_i	150	30	120	10	10	20	35	55	120
s_i	0	10	10	10	10	10	10	10	10

Tabelle 2.2: Beispiel 3: Gesamtentfernung vs. Bediendauer

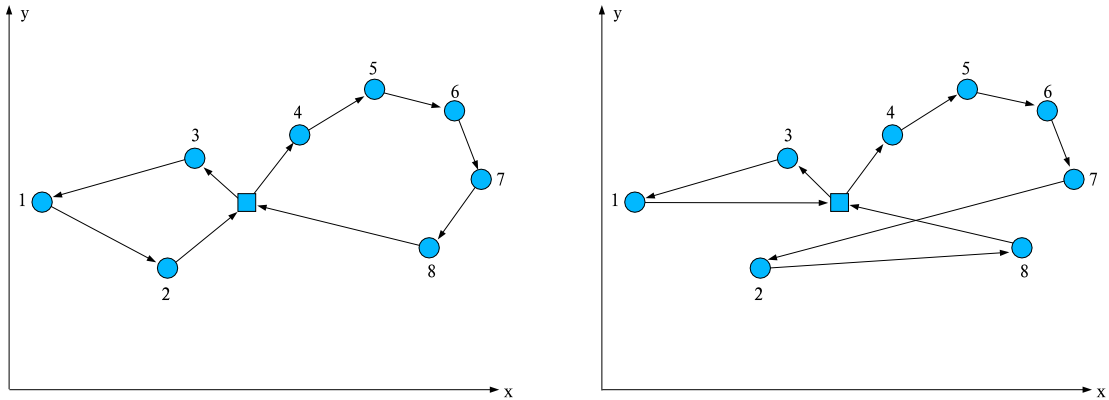


Abbildung 2.11: Beispiel 3: links: Lösung mit minimaler Distanz, rechts: Lösung mit minimaler Bediendauer

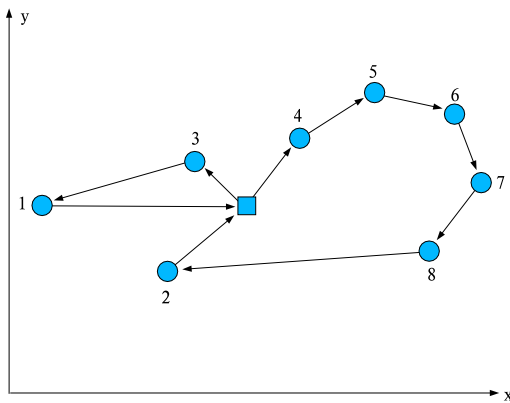


Abbildung 2.12: Beispiel 3: gewichtete Zielfunktion

Die Lösung mit der kleinsten möglichen Gesamtentfernung $dist(X^1) \approx 44EE$ ist $X^1 = \{(0, 3, 1, 2, 0)(0, 4, 5, 6, 7, 8, 0)\}$, mit einer Bediendauer $sched(X^1) \approx 197ZE$ (vgl. Abbildung 2.11, links). Die minimal mögliche Bediendauer $sched(X^2) \approx 155ZE$ wird mit der Lösung $X^2 = \{(0, 3, 1, 0)(0, 4, 5, 6, 7, 2, 8, 0)\}$, bei einer Gesamtentfernung von $dist(X^2) \approx 61EE$, erreicht (vgl. Abbildung 2.11, rechts). Werden die Ziele Gesamtentfernung und Bediendauer im Verhältnis 2 zu 1 gewichtet, ist die Lösung $X^3 = \{(0, 3, 1, 0)(0, 4, 5, 6, 7, 8, 2, 0)\}$, mit einer Entfernung von $dist(X^3) \approx 49EE$ und einer Bediendauer von $sched(X^3) \approx 173ZE$ die bestmögliche Kompromisslösung (vgl. Abbildung 2.12).

2.2.3.2 Lexikographische Zielfunktion für dynamisch-deterministische Tourenplanungsprobleme

Zusammenfassend lässt sich feststellen, daß die in der Literatur gängige Annahme, daß die für die Belieferung aller Kunden benötigte Entfernung und die dafür

erforderlichen Fahrzeiten in einem festen Verhältnis zueinander stehen, selbst für statisch-deterministische Probleme nur in Ausnahmefällen gilt. Für dTP ist diese Annahme grundsätzlich nicht tragbar: Da die Fahrzeiten auf einem Pfeil (m, n) im Zeitablauf schwanken, während die Entfernungen gleich bleiben, kann nicht von einem festen Verhältnis zwischen Fahrzeiten und Entfernungen ausgegangen werden.

Das Ersetzen der entfernungsabhängigen Zielsetzungen durch die Summe der Fahrzeiten, wie es Ichoua et al. (2003) und Donati et al. (2003a) vorgeschlagen¹⁰², lässt sich ökonomisch nicht rechtfertigen. Vielmehr sind zeit- und entfernungsabhängige Kosten differenziert zu berücksichtigen. Für das formale Modell in Abschnitt (2.2.4) wird angenommen, daß sich die Belieferungskosten aus sprungfixen Fahrzeugeinsatzkosten, die mit der Anzahl der Touren ansteigen, entfernungsabhängigen Kosten und bediendauerabhängigen Kosten zusammensetzen.

Für den praktischen Einsatz sind Kostensätze für die mit dem Fahrzeugeinsatz anfallenden Kosten, für die Kosten je zurückgelegter Entfernungseinheit und für die Kosten je Zeiteinheit Bediendauer zu ermitteln. Untersuchungen zu den Verhältnissen dieser Kostensätze für unterschiedliche Einsatzzwecke existieren bisher nicht in der Literatur. Die genaue Festlegung der Kostensätze ist ein von Speditionen gehütetes Geschäftsgeheimnis. Leider war kein befragtes Unternehmen bereit, genaue Zahlen oder auch nur die ungefähren Verhältnisse der Kosten zueinander zu nennen.

Für die experimentellen Untersuchungen in dieser Arbeit wird daher folgende lexikographische Zielfunktion für den relativen Vergleich der Vorteilhaftigkeit von Lösungen verwendet:

$$lexmin = \left\{ \sum_{z \in Z} \zeta_z, \sum_{z \in Z} y_0^z, \sum_{z \in Z} \sum_{i \in C_0} \sum_{j \in C_0} t_{ij}(\tau_i) \cdot x_{ij}^z, \sum_{z \in Z} \sum_{i \in C_0} \sum_{j \in C_0} d_{ij}(\tau_i) \cdot x_{ij}^z, \sum_{i \in C} y_i \right\} \quad (2.2)$$

Wie in der Literatur üblich wird vorrangig die Anzahl der Touren minimiert $(\sum_{z \in Z} \zeta_z)$. Aus der Menge aller Lösungen mit minimaler Tourenzahl werden dann diejenigen Lösungen präferiert, die eine möglichst geringe Gesamtbediendauer aufweisen $(\sum_{z \in Z} y_0^z)$. Dieses Ersatzziel ist geeignet die zeitabhängig anfallenden Kosten zu schätzen, denn es ist davon auszugehen, daß nicht nur mit zunehmenden Fahrzeiten sondern auch mit zunehmenden Wartezeiten steigende Kosten verbunden sind. Aus der verbleibenden Menge der Lösungen werden diejenigen Lösungen bevorzugt, die eine möglichst geringe Gesamtfahrzeit aufweisen $(\sum_{z \in Z} \sum_{i \in C_0} \sum_{j \in C_0} t_{ij}(\tau_i) \cdot x_{ij}^z)$. Da die Servicezeiten bei den einzelnen Kunden reihenfolgeunabhängig sind und jeder Kunde zu bedienen ist

¹⁰²vgl. Ichoua et al. (2003) S. 388; vgl. Donati et al. (2003a), S. 11 und S. 13

(und somit die Servicezeiten nicht entscheidungsrelevant sind), werden bei gleicher Bediendauer Lösungen mit höheren Wartezeiten vorgezogen. Dies ist insofern sinnvoll, als dadurch für unerwartete Störungen mehr „Spielraum“ zum reagieren verbleibt. Ist aus mehreren verbleibenden Lösungen eine auszuwählen, die die gleiche Tourenzahl, Bediendauer und Fahrzeit besitzt, wird diejenige Lösung bevorzugt, die eine geringere Gesamtentfernung aufweist, $\sum_{z \in Z} \sum_{i \in C_0} \sum_{j \in C_0} d_{ij}(\tau_i) \cdot x_{ij}^z$. Es werden aus der resultierenden Lösungsmenge diejenigen Lösungen bevorzugt, die alle Kundenaufträge frühestmöglich bedienen ($\sum_{i \in C} y_i$).¹⁰³ Alle übrigen Kosten werden im folgenden vernachlässigt.

2.2.4 Formales Modell zu dynamisch-deterministischen Tourenplanungsproblemen

Entscheidungsvariablen:

- $x_{ij}^z \in \{0, 1\}$: Die Fahrt von i nach j mit Fahrzeug z ist Teil des Tourenplans X , wenn $x_{ij}^z = 1$ ist.
- $\zeta_z \in \{0, 1\}$: Das Fahrzeug z wird eingesetzt, wenn $\zeta_z = 1$ ist.

Annahmen:

- Das Tourenplanungsproblem ist auf einem bewerteten Graphen (Verkehrswegenetzwerk) gegeben. Die Entfernungen und die Fahrzeiten zwischen den Kunden und zwischen Kunden und Depot sind zeitabhängig. Zwischen zwei Kunden wird stets der schnellste Weg auf Basis des Verkehrswegenetzwerks gewählt.
- Mit dem Einsatz eines zusätzlichen Fahrzeugs, mit steigender Bediendauer und mit steigender Entfernung sind steigende Kosten verbunden. Darüber hinaus fallen keine entscheidungsrelevanten Kosten an. Diese Kosten sind zu minimieren, es wird angenommen, daß sie über die lexikographische Zielordnung (vgl. 2.3) ausreichend genau approximiert sind.
- Jedes Fahrzeug kann höchstens eine Tour im Planungszeitraum durchführen. Jede Tour hat im Depot zu beginnen und zu enden. Ein eingesetztes Fahrzeug ($\zeta_z = 1$) muss das Depot genau einmal verlassen ($\sum_{j \in C} x_{0j}^z = \zeta_z$) und genau einmal wieder

¹⁰³Auf den Zweck dieser Festlegung wird in Kapitel 4.6 näher eingegangen.

erreichen ($\sum_{i \in C} x_{i0}^z = \zeta_z$).¹⁰⁴ Wird ein Fahrzeug z nicht eingesetzt ($\zeta_z = 0$), kann es das Depot auch nicht verlassen (vgl. Bedingung 2.4 und 2.5).

- Mit der Bedienung eines jeden Kunden kann frühestens begonnen werden, nachdem das Depot ist mit Beginn des Planungszeitraums verlassen wurde ($e_0 + t_{0j}(e_0) \leq y_j$), vgl. Bedingung (2.6). Alle Fahrzeuge müssen spätestens zum Ende des Planungszeitraums wieder in das Depot zurückgekehrt sein ($e_0 \leq y_0^z \leq l_0$), vgl. Bedingung (2.7). Die Bedingung $y_i + s_i + t_{i0}(\tau_i) - M \cdot (1 - x_{i0}^z) \leq y_0^z$ stellt dabei sicher, daß das Depot durch Fahrzeug z erst wieder erreicht wird, nachdem alle Kunden auf der Tour z bedient worden sind und das Fahrzeug anschließend ins Depot zurückgefahren ist, vgl. Bedingung (2.8).
- Es steht ein gegebener Fuhrpark von homogenen Fahrzeugen mit einer Gesamtkapazität von maximal Q Kapazitätseinheiten zur Verfügung, jeder Kunde $i \in C$ hat einen spezifischen Bedarf an Gütern, die q_i Kapazitätseinheiten im Lieferfahrzeug belegen: $\sum_{i \in C} q_i \cdot \sum_{j \in C_0} x_{ij}^z \leq Q$, vgl. Bedingung (2.9). Dieser Kapazitätsbedarf ist bei einem Kunden durch genau ein Fahrzeug zu decken, Teillieferungen durch mehrere Fahrzeuge sind nicht möglich.
- Jeder Kunde ist zu bedienen, vgl. Bedingung (2.10).
- Bei jedem Kunden $i \in C$ ist innerhalb eines vorgegebenen Zeitfensters $[e_i; l_i]$ mit der Bedienung zu beginnen ($e_i \leq y_i \leq l_i \forall i \in C$), vgl. Bedingung (2.11).
- Jeder Kunde wird unverzüglich nach Beendigung der Bedienung $y_i + s_i$ zum Zeitpunkt τ_i wieder verlassen, vgl. Bedingung (2.12). Jeder Kunde wird durch genau das Fahrzeug verlassen, durch das er auch erreicht wird, vgl. Bedingung (2.13).
- Bedingung (2.14) legt fest, daß in einer Tour direkt nacheinander bediente Kundenaufträge auch zeitlich aufeinander folgen müssen. y_i ist der Zeitpunkt, zu dem mit der Bedienung des Kunden i begonnen wird, s_i die Dauer der Bedienung. Der Kunde i wird nach Beendigung der Bedienung (zum Zeitpunkt τ_i) verlassen. Für die Fahrt von i nach j werden $t_{ij}(\tau_i)$ Zeiteinheiten benötigt. Beim Kunden j kann also nicht mit der Bedienung begonnen werden, bevor der direkte Vorgängerauftrag i abgeschlossen worden ist und das Lieferfahrzeug z den Kunden j anschließend erreicht hat. M bezeichnet eine hinreichend große

¹⁰⁴Wird angenommen, daß ein Fahrzeug mehrere Touren im Planungszeitraum durchführen kann, entspricht die Anzahl Touren nicht immer der Anzahl der eingesetzten Fahrzeuge. Die hier genannten Bedingungen schliessen einen Mehrfacheinsatz eines Fahrzeugs im Rahmen des hier verwendeten Modells innerhalb des Planungszeitraums aus. Vgl. zum Tourenplanungsproblem mit mehrfachem Einsatz eines Fahrzeugs Taillard et al. (1995).

Konstante. Das Konstrukt $-M \cdot (1 - x_{ij}^z)$ stellt sicher, daß die Restriktion nur für direkt aufeinander folgend bediente Kunden gilt. Gilt $x_{ij}^z = 0$, dann wird der Kunde j nicht direkt im Anschluss an i durch Fahrzeug z bedient - die Bedingung ist dann in jedem Fall erfüllt.

- Die Servicedauer s_i ist bei allen Kunde von der Belieferungsreihenfolge unabhängig.

Zielfunktion

$$lexmin = \left\{ \sum_{z \in Z} \zeta_z, \sum_{z \in Z} y_0^z, \sum_{z \in Z} \sum_{i \in C_0} \sum_{j \in C_0} t_{ij}(\tau_i) \cdot x_{ij}^z, \sum_{z \in Z} \sum_{i \in C_0} \sum_{j \in C_0} d_{ij}(\tau_i) \cdot x_{ij}^z, \sum_{i \in C} y_i \right\} \quad (2.3)$$

unter den Nebenbedingungen:

$$\sum_{j \in C} x_{0j}^z = \zeta_z \quad \forall z \in Z \quad (2.4)$$

$$\sum_{i \in C} x_{i0}^z = \zeta_z \quad \forall z \in Z \quad (2.5)$$

$$e_0 + t_{0j}(e_0) \leq y_j \quad \forall j \in C \quad (2.6)$$

$$e_0 \leq y_0^z \leq l_0 \quad \forall z \in Z \quad (2.7)$$

$$y_i + s_i + t_{i0}(\tau_i) - M \cdot (1 - x_{i0}^z) \leq y_0^z \quad \forall i \in C, \forall z \in Z \quad (2.8)$$

$$\sum_{i \in C} q_i \cdot \sum_{j \in C_0} x_{ij}^z \leq Q \quad \forall z \in Z \quad (2.9)$$

$$\sum_{z \in Z} \sum_{j \in C_0} x_{ij}^z = 1 \quad \forall i \in C \quad (2.10)$$

$$e_i \leq y_i \leq l_i \quad \forall i \in C \quad (2.11)$$

$$y_i + s_i = \tau_i \quad \forall i \in C \quad (2.12)$$

$$\sum_{i \in C_0} x_{iu}^z - \sum_{j \in C_0} x_{uj}^z = 0 \quad \forall z \in Z, \forall u \in C \quad (2.13)$$

$$y_i + s_i + t_{ij}(\tau_i) - M \cdot (1 - x_{ij}^z) \leq y_j \quad \forall i \in C, \forall j \in C, \forall z \in Z \quad (2.14)$$

$$x_{ij}^z \in \{0, 1\} \quad \forall i, j \in C_0, \forall z \in Z \quad (2.15)$$

$$\zeta_z \in \{0, 1\} \quad \forall z \in Z \quad (2.16)$$

3 Charakterisierung von dynamisch-stochastischen Tourensteuerungsproblemen (dTS)

3.1 Störereignisse zur Abbildung von stochastischen Fahrzeiten

Nach Abschluss der Tourenplanung liegt ein Tourenplan X vor. Der Tourenplan X , der zu einem dynamisch-deterministischen Tourenplanungsproblem (dTP) ermittelt wurde, legt die Zuordnung von Kundenaufträgen zu Lieferfahrzeugen, eine vorläufige Belieferungsreihenfolge (Touren) und die zugehörigen Routen fest. Jedes Fahrzeug wird mit den auszuliefernden, kundenspezifischen Gütern beladen, die der Tour zugeordnet sind, die das Lieferfahrzeug zurückzulegen hat. Anschließend wird mit der Durchführung der geplanten Touren begonnen. Sobald das Depot verlassen wird, ist eine Veränderung der Zuordnung von Kundenaufträgen zu Lieferfahrzeugen nicht mehr möglich.

Mit dem Verlassen des Depots wird von der Tourenplanung zur Tourensteuerung übergegangen. Ausgangsbasis für die Tourensteuerung ist eine vorläufige Tour T_z und die dazugehörige Route. Aus einem dynamisch-deterministischen Tourenplanungsproblem entsteht für jedes eingesetzte Fahrzeug z ein eigenes dynamisches Tourensteuerungsproblem, alle $anz(X)$ Tourensteuerungsprobleme sind unabhängig voneinander, parallel zu lösen. Der Einfachheit halber wird im folgenden auf den Tourenindex z verzichtet, da zu jeder Tour ein separates Tourensteuerungsproblem zu lösen ist.

Die Fahrzeiten auf den Pfeilen des Verkehrsnetzwerks, die bei der Ermittlung des Tourenplans X zugrundegelegt wurden, können von den tatsächlichen Ausprägungen der dynamisch-stochastischen Fahrzeiten bei der Auslieferung abweichen. Im Verlauf der Tourdurchführung treten Störereignisse σ auf, die die Fahrzeiten zwischen den

Kunden beeinflussen. Einzelne Strecken können beispielsweise durch unerwartet auftretende Staus oder unangekündigte Schwertransporte eine höhere Fahrzeit erfordern als ursprünglich geplant worden ist. Unerwartet auftretende Änderungen der Witterungsbedingungen können zu einer Veränderung der Geschwindigkeiten auf mehreren Pfeilen des Verkehrswegenetzwerks gleichzeitig führen. Denkbar sind auch Verbesserungen der Verkehrssituation, es kann ein prognostizierter Stau nicht oder weniger stark auftreten und somit die erwartete Fahrzeit geringer ausfallen als vor Beginn der Auslieferung berücksichtigt wurde.

Die Tourensteuerung ist die Reaktion auf veränderte Ausprägungen der stochastischen Geschwindigkeitsfunktionen auf den Pfeilen des Verkehrswegenetzwerks. Der Planungszeitraum $[e_0; l_0]$ wird in regelmäßigen Abständen dahingehend untersucht, ob neue entscheidungsrelevante Störereignisse aufgetreten sind. Auf die Reaktion auf jedes einzelne, neu eintreffende Störereignis wird aus zwei Gründen verzichtet. Zum einen ist bei einer regelmäßig wiederholten Anpassung die Zeitspanne bekannt, bis zu der ein Störungsreaktionsverfahren beendet sein muss. Zum anderen werden kurzfristige Störungen, die sich als irrelevant erweisen, bei der Betrachtung vernachlässigt und die Anzahl der Plananpassungen bleibt beschränkt. Schließlich müssen die gegebenenfalls veränderten Touren und Routen an die Fahrer der Lieferfahrzeuge kommuniziert werden. Nachteil dieser Vorgehensweise ist, daß auf manche kurzfristige Störung möglicherweise nicht rechtzeitig reagiert wird, da sie erst kurz nach einem Reaktionszeitpunkt bekannt wird.

Im Folgenden wird angenommen, daß die Belieferungsreihenfolge der Kunden auf einer Tour und die Routen auch nach dem Verlassen des Depots verändert werden können. τ_{reakt} bezeichnet den Zeitpunkt, in dem die Tourensteuerung durchgeführt werden soll. Zum Zeitpunkt τ_{reakt} werden alle Ausprägungen von Störereignissen $\sigma \in \{S | b_\sigma \leq \tau_{reakt}\}$ berücksichtigt. Im Reaktionszeitpunkt $\tau_{reakt} > e_0$ (nachdem die Lieferfahrzeuge das Depot verlassen haben) sind nur noch die Belieferungsreihenfolgen und die zugehörigen Routen zwischen denjenigen Kundenaufträgen U entscheidungsrelevant, die nicht bereits vor τ_{reakt} besucht worden sind. U bezeichnet im folgenden die Menge der Kundenaufträge, die noch auf der aktuell betrachteten Tour zu bedienen sind. U_0 bezeichnet die Menge der Kundenaufträge zuzüglich des Depots $U := U \cup \{0\}$. Mit y_0 wird der Zeitpunkt bezeichnet, in dem die betrachtete Tour beendet ist.

Die Belieferungsreihenfolge aller Kunden, die bereits besucht worden sind, und die Wege, die bereits zurückgelegt worden sind, sind nicht veränderbar. Bei der Ermittlung der Menge der noch zu bedienenden Kunden U und der tatsächlichen Belieferungszeitpunkte aller bereits bedienten Kunden, sind alle zum Zeitpunkt τ_{reakt} bekannten Ausprägungen der Fahrzeiten zu berücksichtigen. Selbst wenn die

Belieferungsreihenfolge der Kundenaufträge, unverändert bleibt, kann es vorteilhaft sein, von den ursprünglich geplanten Routen abzuweichen und alternative Wege im Verkehrswegenetzwerk zu betrachten.

Mit m wird im folgenden der Knoten des Verkehrswegenetzwerks bezeichnet, der als nächstes nach τ_{reakt} erreicht wird. Dabei muss m nicht ein Kundenstandort sein, vielmehr kann es sich um einen beliebigen Knoten des Verkehrswegenetzwerks handeln ($m \in N$). Mit τ_m wird der Zeitpunkt bezeichnet, zu dem der Knoten m erreicht wird. Vor dem Erreichen des Knotens m ist keine Plananpassung möglich: Zwischen zwei Kreuzungen (z.B. Ausfahrten auf einer Autobahn) besteht weder die Möglichkeit zu wenden noch oder auf eine andere Straße auszuweichen, auch wenn ein Stau auftritt. Erst bei Erreichen der nächsten Kreuzung besteht die Möglichkeit, von der geplanten Route abzuweichen. Für alle dann noch nicht bedienten Kunden können die Belieferungsreihenfolge und die Routen dazwischen angepasst werden.

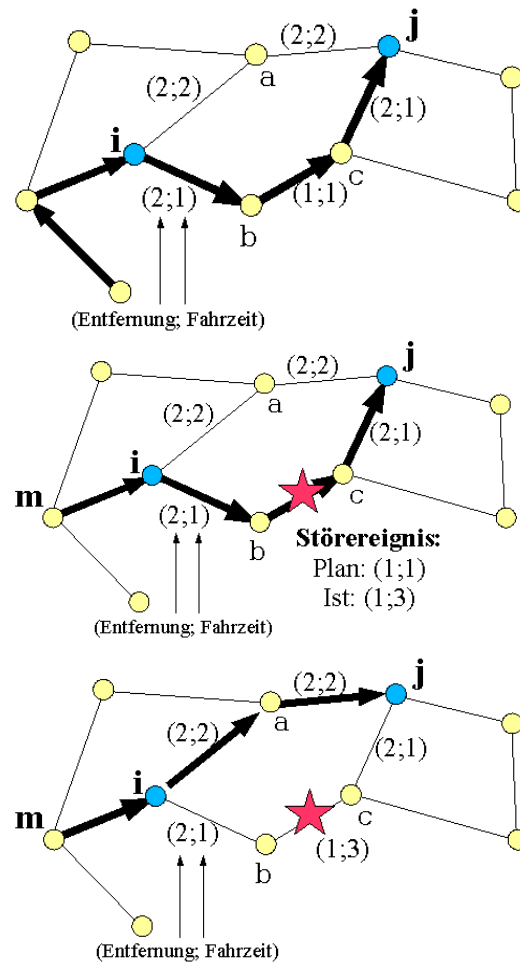


Abbildung 3.1: Anpassung der Route als Reaktion auf ein Störereignis

In Abbildung 3.1 ist ein Ausschnitt aus einer Tour dargestellt. Zunächst wird geplant, den Kunden i vor dem Kunden j zu bedienen. Der schnellste Weg zum Planungszeitpunkt (in Abb. oben) von i zu j ist (i, b, c, j) . Während der Auslieferung

wird ein Störereignis bekannt (in mittlerer Abbildung), das die Fahrzeit zwischen b und c erhöht. Die ursprünglich geplante Route ist nicht mehr vorteilhaft, stattdessen wird der Weg von i zu j über a führen (in Abb. unten). Im Vergleich zur geplanten Tour erhöht sich nach der Reaktion auf das Störereignis die Gesamtfahrzeit um eine Zeiteinheit und gleichzeitig sinkt die Gesamtentfernung um eine Entfernungseinheit. Ohne Reaktion (Anpassung der Route) würde die Gesamtfahrzeit um zwei Zeiteinheiten ansteigen und die Gesamtentfernung gleich bleiben.

Die Geschwindigkeitsfunktionen für dTS werden analog zu dTP auf Basis eines Verkehrswegenetzwerks modelliert. Die Abbildung der stochastischen Bestandteile der Fahrzeiten basiert auf dem Ansatz von Bock (2004) zur Modellierung von Störereignissen im Rahmen der Speditionellen Planung.¹ Ein Störereignis σ verändert die Geschwindigkeitsfunktion auf einem Pfeil zwischen zwei Knoten im Verkehrswegenetzwerk. Bock (2004) betrachtet jede Veränderung von Fahrzeiten im Planungszeitraum als Störereignis.² Regelmäßige, prognostizierbare Schwankungen der Fahrzeiten, die beispielsweise auf starkes Verkehrsaufkommen zu den Stosszeiten zurückzuführen sind, werden in seinem Ansatz nicht im Rahmen der Planung berücksichtigt. Vielmehr werden variable Fahrzeiten abgebildet, als handelte es sich um unerwartete Störungen. Selbst wenn bekannt ist, wann eine Störung endet, wird dies von Bock (2004) nicht berücksichtigt.³

In dieser Arbeit werden regelmäßige Verkehrsschwankungen durch abschnittsweise definierte Geschwindigkeitsfunktionen für alle Pfeile des Verkehrswegenetzwerks bereits in die Planung, bei der Ermittlung eines Tourenplans X zum dTP, einbezogen. Nur Veränderungen der Geschwindigkeiten, die erst nach Abschluss der Planung bekannt werden, gehen in Form von Störereignissen in die Tourensteuerung ein.

Nachdem die Lieferfahrzeuge das Depot verlassen haben, werden sukzessive die Ausprägungen der Störereignisse bekannt und für die Berechnung der tatsächlich benötigten Fahrzeiten und der daraus resultierenden Ankunftszeitpunkte und Bedienzeitpunkte bei den Kunden verwendet. Der Zeitpunkt, zu dem ein Störereignis bekannt wird, kann vom Beginn der Wirkung abweichen. Wenn während der Auslieferung plötzlich und unerwartet starker Regen einsetzt, sinkt beispielsweise die Geschwindigkeit auf der betroffenen Verbindung sofort ohne Vorwarnung. Auch wenn ein Unfall zu einer sofortigen Beeinträchtigung des Verkehrs führt, tritt die Wirkung des Störereignisses sofort mit Bekanntwerden ein. Wird der Verkehr durch einen Unfall erst langsam aufgestaut und die Unfallmeldung sofort bekanntgegeben, kann die Wirkung des Störereignisses später einsetzen als es bekanntgegeben wird. Das gleiche gilt für eine Wanderbaustelle. Es ist zwar nur kurzfristig bekannt, wann welcher Teilabschnitt

¹vgl. Bock (2004), S. 266

²vgl. hierzu und im folgenden Bock (2004), S. 266

³vgl. Bock (2004), S. 266

einer Autobahn betroffen sein wird, allerdings ist die Wirkung bereits abzusehen, noch bevor die Baustelle den Verkehr beeinträchtigt. Im folgenden bezeichnet b_σ den Zeitpunkt des Bekanntwerdens des Störereignisses σ . Zum Zeitpunkt τ_{reakt} sind alle Ausprägungen der Störereignisse σ , für die $b_\sigma \leq \tau_{reakt}$ gilt, bekannt. Mit e_σ wird der Beginn der Wirkung des Störereignisses bezeichnet, l_σ bezeichnet den Zeitpunkt, zu dem die Wirkung des Störereignisses endet. Nicht alle Störereignisse sind erst mit dem Eintritt ihrer Wirkung bekannt. Teilweise können Störereignisse zu einem Zeitpunkt b_σ bekannt werden, zu dem noch keine entscheidungsrelevante Änderung der Geschwindigkeitsfunktion auf einem Pfeil eintritt. Spätestens mit Beginn der Wirkung des Störereignisses wird allerdings angenommen, daß es bekannt ist, da sich die Störung ab dem Zeitpunkt e_σ beobachten lässt. Daher gilt im folgenden immer $b_\sigma \leq e_\sigma$. Mit m_σ und n_σ wird der Pfeil (m, n) des Verkehrsnetzwerks ($m, n \in N$) spezifiziert, der durch das Störereignis σ betroffen ist. Δ_σ bildet die Wirkung des Störereignisses, also die Veränderung der Geschwindigkeit, ab. Die jeweils aktuelle Geschwindigkeit wird für die Dauer des Störereignisses mit Δ_σ multipliziert. Grundsätzlich gilt $\Delta_\sigma \geq 0$. Liegt Δ_σ zwischen 0 und 1, liegt eine Beeinträchtigung der Geschwindigkeit vor. Wenn keine Störung vorliegt, gilt $\Delta_\sigma = 1$ und bei einer Verbesserung der Verkehrssituation im Vergleich zur Planung gilt $\Delta_\sigma > 1$. Gegebenenfalls wird die Geschwindigkeitsfunktion des betroffenen Pfeils durch ein oder zwei zusätzliche Intervalle ergänzt:

Mit int_1 wird das Intervall bezeichnet, in dem das Störereignis beginnt ($\tau_e^{int_1} \leq e_\sigma < \tau_l^{int_1}$), int_2 ist das Intervall in dem das Störereignis endet ($\tau_e^{int_2} < l_\sigma \leq \tau_l^{int_2}$). Gilt $\tau_e^{int_1} \neq e_\sigma$ wird int_1 in zwei disjunkte Intervalle int'_1 und int''_1 ($int_1 = int'_1 \cup int''_1$) geteilt, so daß der Beginn des Störereignisses stets auf eine Intervallgrenze fällt. Die Intervallgrenzen ergeben sich zu $int'_1 := [\tau_e^{int_1}; e_\sigma[$ und $int''_1 := [e_\sigma; \tau_l^{int_1}[$. Analog wird int_2 an der Stelle l_σ geteilt, wenn $l_\sigma \neq \tau_l^{int_2}$ gilt. Anschließend wird die Geschwindigkeitsfunktion für den Pfeil (m, n) angepasst:

$$g_{mn}^{int} := \begin{cases} g_{mn}^{int} \cdot \Delta_\sigma & \forall int \text{ mit } \tau_e^{int} \geq e_\sigma \text{ und } \tau_l^{int} < l_\sigma \\ & \text{und } m = m_\sigma \text{ und } n = n_\sigma \\ g_{mn}^{int} & \text{sonst} \end{cases}$$

Die Menge aller Störereignisse wird im folgenden als Störplan S bezeichnet.

3.2 Formales Modell zu dynamisch-stochastischen Tourensteuerungsproblemen

3.2.1 Lexikographische Zielfunktion für dynamisch-stochastische Tourensteuerungsprobleme

Zur Beurteilung der Vorteilhaftigkeit von Lösungen zum dynamisch-stochastischen Tourensteuerungsproblem wird eine lexikographische Zielordnung verwendet, da wie beim dynamisch-deterministischen Tourenplanungsproblem die Kostensätze für die einzelnen Kosteneinflussfaktoren nicht allgemeingültig bestimmt werden können und empirische Ausprägungen der Kostengrößen nicht erhältlich sind.⁴ Zu beurteilen ist eine Tour und die zugehörige Route, sie werden im folgenden auch als Lösung des dTS bezeichnet.

Die primäre Zielsetzung für dTP „Minimierung der Tourenzahl“ ist für dTS nicht entscheidungsrelevant, da mit dem Verlassen des Depots die Anzahl Touren feststeht. Vielmehr wird beim dynamisch-stochastischen Tourensteuerungsproblem jede Tour isoliert betrachtet. Als primäre Zielsetzung wird die Minimierung der Gesamtverspätung $\sum_{i \in U_0} \max \{(y_i - l_i); 0\}$ gewählt. Eine Überschreitung des Zeitfensters für einen Kundenauftrag ist zu vermeiden, da die Abläufe des Kunden dadurch beeinträchtigt werden können. Im Rahmen von Just-in-Time-Lieferverträgen werden üblicherweise hohe Strafkosten für verspätete Anlieferungen vereinbart. Darüber hinaus kann sich ein verspätet bedienter Kunde langfristig für einen anderen Lieferanten entscheiden. Es wird daher davon ausgegangen, daß durch eine Verspätung verursachte Kosten vorrangig vermieden werden sollen.

In dem im folgenden vorgeschlagenen Modell zur Tourensteuerung wird unterstellt, daß die Strafkosten proportional mit der Verspätung ansteigen und für alle Kundenaufträge sowie für das Depot in gleicher Höhe anfallen. Alle Kundenaufträge sind, wie bei dynamisch-deterministischen Tourenplanungsproblemen, frühestmöglich zu bedienen. Wenn alle Kundenaufträge frühestmöglich bedient werden, führt dies dazu, daß alle Kunden auch frühestmöglich wieder verlassen werden. Tritt auf dem Weg von i zu j ein Störereignis auf, welches die Fahrzeiten auf der gegebenen Route verlängert, wird j bei Gültigkeit des FIFO-Prinzips zwar später als geplant, aber dennoch frühestmöglich erreicht. Es soll nicht verschwiegen werden, daß sich in Einzelfällen die Festlegung des Abreisezeitpunkts auf den frühestmöglichen Zeitpunkt auch nachteilig auswirken kann. Angenommen, ein Fahrzeug fährt von i zu j über den Pfeil (m, n) . Während es den Pfeil (m, n) durchfährt, tritt eine Störung auf, die die Geschwindigkeit auf (m, n) deutlich

⁴vgl. auch Abschnitt 2.2.3.2

reduziert. Wäre der Kunde i zu einem späteren Zeitpunkt verlassen worden, wäre die Störung bereits bekannt gewesen und das Fahrzeug hätte einen alternativen Weg über (m', n') gewählt und den Kunden j früher erreicht. In diesem Fall hätte das Warten bei Kunde i vorteilhaft sein können. Da allerdings die Störungen nicht im Vorhinein bekannt sind und der Pfeil (m', n') genauso durch eine Störung betroffen sein kann, lässt sich mit dieser Begründung nicht rechtfertigen bei einem Kunden nicht frühestmöglich mit der Belieferung zu beginnen und ihn wieder frühestmöglich zu verlassen.

Wenn mehrere Touren mit gleicher Gesamtverspätung zu vergleichen sind, wird als sekundäres Vorteilhaftigkeitskriterium die Gesamtbediendauer minimiert. Aus der Menge der dann verbleibenden Lösungen, werden diejenigen Lösungen mit der geringsten Fahrzeit präferiert. Da sich die Bediendauer aus der Summe aus Gesamtfahrzeit, Wartezeit und (konstanter, somit nicht entscheidungsrelevanter) Servicezeit zusammensetzt, entfällt bei gegebener Bediendauer ein größerer Anteil der Bediendauer auf die Wartezeit. Somit besteht mehr Freiraum für Reaktionen auf Störereignisse. Aus der Menge der verbleibenden Touren werden diejenigen Touren bevorzugt, die eine möglichst geringe Gesamtentfernung aufweisen und daraus wiederum die Menge der Touren mit minimaler Anzahl verspäteter Kundenaufträge. Wird ein Kundenauftrag i verspätet bedient (gilt also $y_i > l_i$), ist $late_i = 1$, ansonsten ist $late_i = 0$. Die Anzahl der verspäteten Kundenaufträge ergibt sich somit zu $\sum_{i \in C_0} late_i$. Wie auch für dTP gilt bei verbleibenden Freiheitsgraden, daß aus der Menge der bei verbleibenden Touren immer diejenige Tour gewählt wird, bei der alle Kunden frühestmöglich verlassen werden.

Die lexikographische Zielsetzung lautet:

$$lexmin \text{ Kosten} = \left\{ \begin{array}{l} \sum_{i \in U_0} \max\{(y_i - l_i); 0\}; y_0; \sum_{i \in U \cup \{m\}} \sum_{j \in U_0} t_{ij}(\tau_i) \cdot x_{ij}; \\ \sum_{i \in U \cup \{m\}} \sum_{j \in U_0} d_{ij}(\tau_i) \cdot x_{ij}; \sum_{i \in C_0} late_i; \sum_{i \in U} y_i \end{array} \right\} \quad (3.1)$$

3.2.2 Problembeschreibung (deskriptives Modell)

Entscheidungsvariable

- $x_{ij} \in \{0; 1\}$: Die Fahrt von i zu j ist Teil der zurückzulegenden Tour, wenn $x_{ij} = 1$ ist.

Annahmen:

- Das Tourensteuerungsproblem ist auf einem bewerteten Graphen gegeben. Die Entfernungen und Fahrzeiten zwischen den Kunden und zwischen Kunden und dem Depot sind zeitabhängig. Die Fahrzeit zwischen zwei Knoten des Verkehrswegenetzwerks kann durch Störereignisse verändert werden. Zwischen zwei Kunden wird stets der Weg gewählt, der unter Berücksichtigung aller zum Reaktionszeitpunkt bekannten Störereignisse der schnellste Weg ist. Er wird mit Hilfe des Verfahrens aus Abschnitt 2.2.2.4 ermittelt.
- Mit einer verspäteten Belieferung eines Kunden, mit steigender Bediendauer und steigender Gesamtentfernung sind steigende Kosten verbunden. Es wird angenommen, daß alle entscheidungsrelevanten Kosten über die lexikographische Zielordnung (vgl. 3.2) ausreichend genau approximiert werden. Entscheidungsrelevant ist die Belieferungsreihenfolge der Menge der noch nicht bedienten Kunden U (Resttour), sowie die dabei zurückgelegte Route. Die Route bestimmt die Entfernungen $d_{ij}(\tau_i)$ und die Fahrzeiten $t_{ij}(\tau_i)$ im Verkehrswegenetzwerk.
- Die Tour ist erst beendet, wenn alle noch zu bedienenden Kunden und das Depot ($j \in U_0$) erreicht worden sind (3.3). Jeder Kunde aus U ist genau einmal zu bedienen und anschliessend zu verlassen (3.4). Auch der Knoten m , an dem sich das Lieferfahrzeug zum frühestmöglichen Reaktionszeitpunkt τ_m befindet, ist genau einmal zu verlassen.
- Die Nebenbedingung (3.5) stellt sicher, daß bei keinem Kunden $j \in U$ mit der Belieferung begonnen werden kann, bevor nicht das Lieferfahrzeug den Kunden erreicht hat. Die Belieferung eines Kunden darf nicht vor Beginn seines Zeitfensters beginnen (3.6). Beginnt die Belieferung eines Kunden erst nach dem Ende seines Zeitfensters, fallen Strafkosten für verspätete Belieferungen an. Jeder Kunde wird unmittelbar nach Beendigung der Bedienung $y_i + s_i$ (frühestmöglich) zum Zeitpunkt τ_i wieder verlassen (3.7).
- Die Bedingung (3.8) legt fest, daß direkt nacheinander bediente Kunden auch zeitlich aufeinanderfolgen müssen.

- Zur Ermittlung der Position des Lieferfahrzeugs im Verkehrswegenetzwerk, der erwarteten Ankunftszeit τ_m beim nächsten Knoten des Verkehrswegenetzwerks und der Fahrzeiten zwischen den Knoten, werden dem jeweils aktuellen Informationsstand entsprechende Geschwindigkeitsfunktionen zur Ermittlung der zeitlichen ($t_{ij}(\tau_i)$) und räumlichen ($d_{ij}(\tau_i)$) Distanzen zwischen den Kunden zugrunde gelegt. Wie beim dynamisch-deterministischen Tourenplanungsproblem werden die schnellsten Wege gewählt, sie werden mit dem Algorithmus von Dijkstra (1959) ermittelt.⁵ Gegebenenfalls ist dazu der Planungshorizont über l_0 hinaus zu verlängern, damit die Tour beendet werden kann. Ein Störereignis kann dazu führen, daß die betrachtete Tour nicht vor l_0 beendet werden kann.

Zielfunktion

$$\text{lexmin Kosten} = \left\{ \begin{array}{l} \sum_{i \in U_0} \max\{(y_i - l_i); 0\}; y_0; \sum_{i \in U \cup \{m\}} \sum_{j \in U_0} t_{ij}(\tau_i) \cdot x_{ij}; \\ \sum_{i \in U \cup \{m\}} \sum_{j \in U_0} d_{ij}(\tau_i) \cdot x_{ij}; \sum_{i \in U_0} \text{late}_i; \sum_{i \in U} y_i \end{array} \right\} \quad (3.2)$$

unter den Nebenbedingungen:

$$\sum_{i \in U \cup \{m\}} x_{ij} = 1 \quad \forall j \in U_0 \quad (3.3)$$

$$\sum_{j \in U_0} x_{ij} = 1 \quad \forall i \in U \cup \{m\} \quad (3.4)$$

$$\tau_m + t_{mj}(\tau_m) \leq y_j \quad \forall j \in U_0 \quad (3.5)$$

$$e_i \leq y_i \quad \forall i \in U \quad (3.6)$$

$$y_i + s_i = \tau_i \quad \forall i \in U \quad (3.7)$$

$$y_i + s_i + t_{ij}(\tau_i) - M \cdot (1 - x_{ij}) \leq y_j \quad \forall i \in U, \forall j \in U_0 \quad (3.8)$$

$$x_{ij} \in \{0; 1\} \quad \forall i, j \in U_0 \quad (3.9)$$

⁵vgl. Dijkstra (1959), S. 269-271

4 Ausgewählte Heuristiken für Tourenplanungs- und -steuerungsprobleme

4.1 Exkurs: Zulässigkeitsbedingungen nach Ahn und Shin (1991)

Die Überprüfung der Zulässigkeit eines Tourenplans ist sehr zeitaufwendig, insbesondere aufgrund der aufwendigen Berechnung der Wege im dynamischen Verkehrswegenetzwerk. Im folgenden werden Bedingungen vorgestellt, welche die Algorithmen teilweise erheblich beschleunigen können, da sie häufig wiederholte Zulässigkeitsprüfungen beschleunigen. Im Zuge der Anwendung von Verbesserungsverfahren wird darüber hinaus nicht immer eine komplette Lösung auf Zulässigkeit überprüft. Vielmehr werden nur diejenigen Touren überprüft, die durch einen Tauschvorgang im Rahmen der lokalen Suche überhaupt betroffen sind. Alle übrigen Touren bleiben unverändert und benötigen daher keine Zulässigkeitsüberprüfung.

Neben der Überprüfung der Kapazitätsrestriktionen ((2.9) auf Seite 42) muss bei jedem Einfüge- oder Tauschvorgang im Rahmen der Lösungskonstruktion und -verbesserung überprüft werden, ob die Zeitfensterrestriktionen (2.7) und (2.11) eingehalten werden. Eine Zulässigkeitsprüfung kann grundsätzlich als „Toursimulation“ oder als „Restriktionsprüfung vor Modifikation“ durchgeführt werden.¹ Die „Toursimulation“ erzeugt zunächst eine neue Tour, unter der Annahme sie sei zulässig. Anschließend wird die Tour Kunde für Kunde nachvollzogen und dabei überprüft, ob tatsächlich alle Restriktionen eingehalten werden. Diese Vorgehensweise ist allerdings sehr zeitaufwendig, da auch Lösungen überprüft werden, die bei Kenntnis der Problemeigenschaften sehr schnell als unzulässig verworfen werden können. Sinnvoll ist daher die „Restriktionsprüfung vor Modifikation“: Zunächst wird vor der Modifikation überprüft, ob einzelne Restriktionen durch eine bestimmte Änderung

¹vgl. Homberger (2000), S. 96

der Belieferungsreihenfolge verletzt werden, um anschließend nur diejenigen Lösungen komplett durchzurechnen, die zulässig sein können.

Für die Zulässigkeitsprüfung eines Nachbarschaftstauschvorgangs ist nicht ausreichend, alleine die betroffenen Knoten des zu tauschenden Abschnitts zu prüfen, sondern es müssen alle im Anschluss an den Abschnitt zu bedienenden Kundenaufträge dahingehend untersucht werden, ob sie noch rechtzeitig erreicht werden können. Dabei erleichtert die Gültigkeit des FIFO-Prinzips (vgl. Abschnitt 2.2.2.1) die Untersuchung. h bezeichnet den direkten Vorgänger (Depot oder Kundenauftrag) des Kundenauftrags i auf der aktuellen Tour, j ist der direkte Nachfolger von i . faz_i ist der frühestmögliche Beginn der Bedienung des Kunden i auf der aktuellen Tour, saz_i ist der spätestmögliche Beginn. Für das Depot gilt immer $faz_0 := e_0$ und $saz_0 := l_0$.

Unter der Annahme, daß das FIFO-Prinzip gilt, lassen sich zu allen Fahrzeitfunktionen $t_{ij}(\tau)$ von i zu j , mit $i, j \in C_0$ bei Abreise im Zeitpunkt τ auch die Funktion $t_{ij}^{-1}(\tau)$ ermitteln.² Sie geben die benötigte Fahrzeit zwischen i und j an, wenn das Fahrzeug zum Zeitpunkt τ bei j ankommen soll. Für jede Tour des aktuellen Tourenplans können rekursiv die faz und saz für alle Kunden ermittelt werden.

$$\begin{aligned} faz_i &:= \max \{ faz_h + s_h + t_{hi}(faz_h + s_h); e_i \} & \forall i \in C \\ saz_i &:= \min \{ saz_j - t_{ij}^{-1}(saz_j) - s_i; l_i \} & \forall i \in C \end{aligned}$$

Ahn und Shin (1991) formulieren drei Eigenschaften von zulässigen Touren, welche die Zulässigkeitsprüfung bei dynamisch-deterministischen Tourenplanungsproblemen erleichtern können.³

Zulässigkeitsbedingung 1⁴: wird ein (bisher unzugeordneter) Kunde u zwischen zwei in einer zulässigen Tour direkt aufeinanderfolgende Kunden i und j eingefügt, bleibt die Tour zulässig, wenn $faz_u \leq saz_u$ gilt und die Kapazität ausreicht. faz_u bezeichnet den frühestmöglichen Beginn der Bedienung des Kunden u auf der aktuellen Tour und saz_u bezeichnet den spätestmöglichen Beginn der Bedienung des Kunden u auf der aktuellen Tour. Es gilt $faz_u := \max\{e_u, faz_i + s_i + t_{iu}(faz_i + s_i)\}$ und $saz_u := \min\{l_u, saz_j - t_{uj}^{-1}(saz_j) - s_u\}$. Ein Kunden u kann nur dann zwischen i und j eingefügt werden, wenn der frühestmögliche Beginn der Bedienung von u (faz_u) vor dem spätestmöglichen Beginn der Bedienung des Kunden u (saz_u) liegt.

Durch die Ermittlung der faz und saz und deren Berücksichtigung bei Einfüge- und Tauschvorgängen sinkt der Rechenaufwand für Zulässigkeitsprüfungen sehr deutlich.⁵ Bei einer Einfügeoperation muss nicht mehr die komplette (Teil-)Tour auf

²vgl. Ahn und Shin (1991), S. 394

³vgl. Ahn und Shin (1991), S. 395f.

⁴vgl. Ahn und Shin (1991), S. 395

⁵vgl. Ahn und Shin (1991), S. 398f.

Überschreitung der Zeitfenster überprüft werden, sondern nur noch der einzufügende Knoten. Gerade bei Problemen mit langen Touren (viele Kunden je Tour) ist diese damit verbundene Rechenzeiterparnis besonders spürbar.

Zulässigkeitsbedingung 2⁶: werden zwei Touren T_{z1} und T_{z2} so aneinandergelängt, daß der letzte Knoten i der Tour T_{z1} vor dem ersten Knoten j der Tour T_{z2} bedient werden soll, muss gelten: $faz_i + s_i + t_{ij}(faz_i + s_i) \leq saz_j$. Es muss vor dem spätestzulässigen Beginn der Belieferung des Kunden j (saz_j) möglich sein, den Kunden i zu bedienen und anschließend zu j zu fahren. Diese Zulässigkeitsbedingung ist von Ahn und Shin (1991) für die Anwendung des Savings-Verfahrens formuliert worden.⁷

Im Rahmen der Experimente in dieser Arbeit wird sie bei der sequentiellen Konstruktion von neuen Touren des dynACO-Algorithmus (vgl. Abschnitt 4.5) verwendet: für alle potentiellen Nachfolger $u \in U$ des aktuellen Knotens i werden die $saz_u := \min\{l_u, l_0 - t_{u0}^{-1}(l_0) - s_u\}$ bestimmt. Sind die potentiellen Nachfolger u vor dem Zeitpunkt saz_u erreichbar und wird die Kapazitätsrestriktion (2.9) nicht verletzt, ist das Anhängen des Kunden u , $u \in U'$, an die aktuelle Tour zulässig. Eine Rechenzeiterparnis ist erzielbar, da die saz_u für alle $u \in C$ nur einmal bestimmt werden müssen. saz_u ist unabhängig von der aktuellen Tour.

Zulässigkeitsbedingung 3⁸: Häufig werden für Verbesserungsverfahren Verschiebe- oder Tauschvorgänge von Abschnitten benötigt.⁹ Dabei werden ein oder mehrere Abschnitte aus einer oder mehreren Touren vertauscht oder verschoben. Um die Zulässigkeit eines Tausch- oder Verschiebevorgangs überprüfen zu können, wird der zu verschiebende Abschnitt von einschließlich x'_1 bis einschließlich y_1 näher untersucht. Der Knoten x_1 bezeichnet den letzten Knoten ($x_1 \in C_0$), der vor zu dem zu tauschenden Abschnitt in einer Tour T_{z1} angeordnet ist. x'_1 ist der direkt auf x_1 folgende Knoten ($x'_1 \in C_0$) auf der Tour T_{z1} . y'_1 bezeichnet den Knoten der Tour T_{z1} , der als erstes nach dem zu tauschenden Abschnitt folgt ($y'_1 \in C_0$), y_1 ist der Vorgängerknoten von y'_1 auf der Tour T_{z1} ($y_1 \in C_0$). Dieser Abschnitt wird im folgenden mit V bezeichnet. Für Abschnitt V wird ein Abschnittszeitfenster $[e_V; l_V]$ ermittelt.¹⁰ Die Zulässigkeitsbedingung 3 besagt, daß damit der Abschnitt V zwischen zwei aufeinanderfolgend bediente Kunden i und j eingefügt werden kann, $faz_i + t_{i,x'_1}(faz_i + s_i) \leq e_V$ und $l_V + s_{y_1} + t_{y_1,j}^{-1}(l_V + s_{x_1}) \leq saz_j$ gelten muß. Damit ein Abschnitt einer Tour an eine andere Stelle verschoben werden kann, reicht die Überprüfung der Zulässigkeitsbedingung 3 allerdings nicht aus - sie ist laut Ahn und Shin (1991) notwendige, aber nicht hinreichende Bedingung für einen zulässigen Verschiebevorgang.¹¹ Bevor ein Abschnitt tatsächlich verschoben werden

⁶ vgl. Ahn und Shin (1991), S. 395

⁷ vgl. zum Savings-Verfahren Abschnitt 4.2.1

⁸ vgl. Ahn und Shin (1991), S. 395f.

⁹ vgl. beispielsweise Cross-Exchange in Abschnitt 4.3.2

¹⁰ vgl. zur genauen Ermittlung Ahn und Shin (1991), S. 395

¹¹ vgl. Ahn und Shin (1991), S. 397

soll, muss eine exakte Prüfung der Zulässigkeit des Verschiebevorgangs erfolgen.

Allerdings wird von Ahn und Shin (1991) übersehen, daß Zulässigkeitsbedingung 3 auch nur dann tatsächlich eine notwendige Bedingung für die Zulässigkeit eines Tausch- oder Verschiebevorgangs ist, wenn der zu verschiebende Abschnitt aus einer Tour in eine andere Tour verschoben wird. Findet der Verschiebevorgang innerhalb einer Tour statt oder werden zwei Abschnitte innerhalb einer Tour vertauscht, ändern sich möglicherweise auch die faz und saz von Kunden die nicht verschoben werden. Betroffen sind alle Kunden zwischen der Position, an der der Abschnitt eingefügt werden soll und der Position, von der der Abschnitt entfernt wird. Dazu sei ein Beispiel gegeben (vgl. Abbildung 4.1). Gegeben sei eine Tour $(\dots, x_1, x_1', \dots, y_1, y_1', \dots, x_2, x_2', \dots)$.¹² Der Abschnitt von x_1' bis y_1 wird zwischen die Aufträge x_2 und x_2' verschoben. Es ergibt sich die neue Tour $(\dots, x_1, y_1', \dots, x_2, x_1', \dots, y_1, x_2', \dots)$. Dabei ändern sich nicht nur die faz und saz der Kundenaufträge die zu dem verschobenen Abschnitt gehören, sondern auch die faz und saz der Kundenaufträge zwischen y_1' bis x_2 .

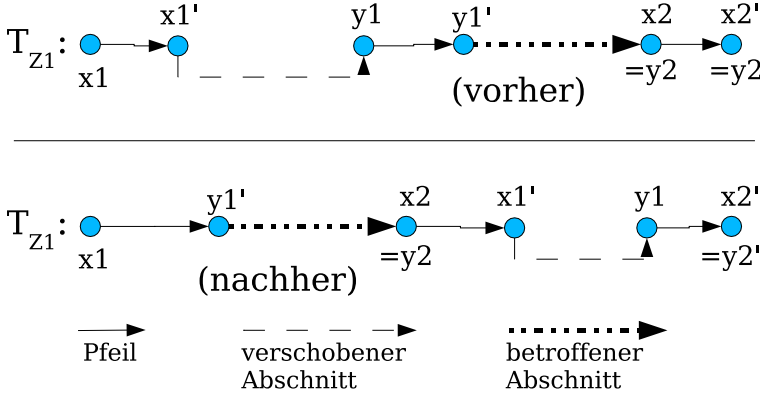


Abbildung 4.1: Zu Zulässigkeitsbedingung 3 nach Ahn und Shin (1991)

Wird die Zulässigkeitsbedingung 3 von Ahn und Shin (1991) oder die Approximation auf CX-Tauschvorgänge von zwei Abschnitten innerhalb von einer Tour angewandt, kann es passieren, daß zulässige Lösungen verworfen werden. Wenn zwei Abschnitte vertauscht werden, sind die faz und saz der Kundenaufträge zwischen den zu tausenden Abschnitten nicht geeignet, um die Zulässigkeit eines Tauschvorgangs zu überprüfen.

In dieser Arbeit wird die Zulässigkeitsbedingung 3 nicht verwendet, da die Ermittlung von Abschnittszeitfenstern sehr aufwendig ist und zudem nicht ausreicht um die Zulässigkeit eines Verschiebe- oder Tauschvorgangs sicherzustellen. Stattdessen wird

¹²Die Notation für einen Abschnitt entspricht der Notation, die für die Beschreibung von CX in Abschnitt 4.3.2 verwendet werden wird.

für einen Abschnitt überprüft, ob der erste Kunde des Abschnitts noch rechtzeitig erreicht werden kann, wenn er direkt im Anschluß an i bedient wird ($faz_i + s_i + t_{i,x'_1}(faz_i + s_i) \leq l_{x'_1}$) und - unabhängig davon - ob im Anschluß an die Belieferung des letzten Kunden des Abschnitts noch der Kunde j rechtzeitig erreicht werden kann ($e_{y_1} + s_{y_1} + t_{y_1,j}(e_{y_1} + s_{y_1}) \leq saz_j$). Diese Approximation ist zwar weniger genau als Zulässigkeitsbedingung 3, allerdings entfällt auch die Rechenzeit, die für die Ermittlung des Abschnittszeitfensters benötigt wird.

Für Tauschvorgänge von zwei Abschnitten innerhalb einer Tour werden daher nur für Kunden vor dem ersten Abschnitt die jeweiligen faz und nur für Kunden nach dem zweiten Abschnitt die jeweiligen saz bestimmt. Für die zu vertauschenden Abschnitte und alle Kundenaufträge zwischen den Abschnitten erfolgt eine Toursimulation zur Überprüfung der Zulässigkeit eines Tauschvorgangs.

Die Zulässigkeitsbedingungen 1 und 2 sowie die Approximation zu Zulässigkeitsbedingung 3 sind von großer Bedeutung für eine effiziente Implementation der im folgenden vorgestellten Algorithmen. Eine Vielzahl unzulässiger Tauschvorgänge lässt sich ohne aufwendige Toursimulation aussortieren und es lässt sich viel Rechenzeit einsparen.

4.2 Konstruktionsverfahren für Tourenplanungsprobleme

Konstruktionsverfahren führen eine Folge von Operationen durch, bis ein zulässiger Tourenplan resultiert. Die in den Abschnitten 4.2.1, 4.2.2 und 4.2.3, vorgestellten Verfahren sind Greedy-Heuristiken. In jedem Konstruktionsschritt wird diejenige Operation ausgeführt, die die kleinste Verschlechterung des Zielfunktionswerts mit sich bringt.

4.2.1 Savings-Heuristik

Das Savings-Verfahren ist ein paralleles Konstruktionsverfahren.¹³ Als Endkunden werden im folgenden die als erstes und als letztes in einer Tour bedienten Kunden bezeichnet. Das Savings-Verfahren basiert auf dem sukzessiven Aneinanderhängen von jeweils zwei unterschiedlichen Touren, indem zwei Endkunden je einer Tour miteinander verbunden werden.

Zunächst werden zu allen Kunden $i \in C$ Pendeltouren gebildet. Eine Pendeltour $(0, i, 0)$ bedient genau einen Kunden i . Anschließend werden die Savingswerte $sav_{ij} :=$

¹³vgl. hierzu und im folgenden Solomon (1987), S. 256

$d_{i0} + d_{0j} - d_{ij}$ für alle Kundenpaare $i, j \in C$ ermittelt. Der Savingswert misst die Wegstreckensparnis, die durch das Aneinanderhängen von zwei Touren an ihren Endknoten erzielt werden kann.

Alle Savingswerte werden der Größe nach in nicht-aufsteigender Reihenfolge sortiert und in einer Liste gespeichert. Solange die Liste der Savingswerte nicht leer ist, wird der jeweils erste Savingswert sav_{ij} aus der Liste entfernt. Sind die Kunden i und j Endkunden von zwei unterschiedlichen Touren und können die zu i und j gehörigen Touren derart verbunden werden, daß die resultierende Tour zulässig ist, werden diese beiden Touren zu einer neuen Tour zusammengefügt. Die Wegstrecke zum Depot nach der Belieferung des Endkunden der ersten Tour (d_{i0}) und die Wegstrecke vom Depot zum Endkunden der zweiten Tour (d_{0j}) wird eingespart, zusätzlich ist die Strecke von Kunde i zu Kunde j (d_{ij}) zurückzulegen. Voraussetzung für die Zulässigkeit ist, daß i und j Endkunden unterschiedlicher Touren sind und keine Kapazitäts- oder Zeitfensterrestriktionen durch das Aneinanderhängen der beiden Touren verletzt werden. Ist der ausgewählte Endkunde i einer Tour T_1 der erste auf der Rundreise zu beliefernde Kunde, wird die Belieferungsreihenfolge der Kunden der Tour T_1 vor dem Aneinanderhängen umgekehrt. Das Verfahren endet sobald die Liste der Savingswerte leer ist.

Solomon (1987) schlägt eine Anpassung des Savings-Verfahrens von Clarke und Wright (1964) für statisch-deterministische Tourenplanungsprobleme mit Zeitfensterrestriktionen vor.¹⁴ Ahn und Shin (1991) passen das Savingsverfahren an dynamisch-deterministische Tourenplanungsprobleme mit variablen Fahrzeiten zwischen den Kunden an.¹⁵

Während das Verfahren für Tourenplanungsprobleme ohne Zeitfensterrestriktionen relativ gut geeignet ist, sind die Ergebnisse für Probleme mit Zeitfensterrestriktionen relativ schlecht.¹⁶ Das ist darauf zurückzuführen, daß zwar die Wegstrecken (bzw. Fahrzeiten) verkürzt werden, dabei aber aufgrund der Zeitfensterrestriktionen bei den Kunden teilweise enorme Wartezeiten in Kauf zu nehmen sind. Die Fahrzeuge werden nicht gut ausgelastet und es werden möglicherweise mehr Touren geplant als eigentlich erforderlich wären.

Die Savingswerte weisen außerdem, sofern sie entfernungsbasiert ermittelt werden, wenig Aussagekraft bezüglich der potentiellen Verbesserung der Lösung auf. Selbst wenn die Entfernung d_{ij} durch die Fahrzeit t_{ij} ersetzt wird, darf bezweifelt werden, ob der Savings-Wert ein sinnvoller Indikator für die Vorteilhaftigkeit des Zusammenfügens von zwei Touren ist: neben der Fahrzeit t_{ij} können auch Wartezeiten bei Ankunft vor

¹⁴vgl. Solomon (1987), S. 256

¹⁵vgl. Ahn und Shin (1991), S. 397

¹⁶vgl. Solomon (1987), S. 263; vgl. Gietz (1994), S. 220

dem Beginn des Zeitfensters auftreten. Bei der Existenz von Zeitfensterrestriktionen ist auch problematisch, daß bei der Hälfte aller Endkunden die Reihenfolge einer Tour umgekehrt wird. Dies wird zu zahlreichen unzulässigen Kombinationsversuchen führen.

Bei dynamisch-deterministischen Tourenplanungsproblem (dTP) kommt ein weiteres Problem hinzu: Basiert die Ermittlung der Savingswerte auf zeitlichen Abständen zwischen den Kunden, lassen sie sich nicht allgemein gültig bestimmen. Die Savingswerte sind vielmehr nach jedem Iterationsschritt neu zu ermitteln, da sie in Abhängigkeit vom Abreisezeitpunkt τ variieren. Der Vorteil der einmaligen Ermittlung und der daraus resultierenden niedrigen Rechenzeit entfällt. Das Savings-Verfahren wird daher nicht näher betrachtet.

4.2.2 Nearest-Neighbor-Heuristik

Solomon (1987) schlägt ein sequentielles Verfahren des besten Nachfolgers (Nearest-Neighbor-Heuristik, NN) für statisch-deterministische Tourenplanungsprobleme mit Zeitfensterrestriktionen vor.¹⁷ Formal ist die allgemeine Funktionsweise von Nearest-Neighbor-Heuristiken in Algorithmus 4 dargestellt. Dabei sind x_{ij}^z die Entscheidungsvariablen aus dem Modell aus Abschnitt 2.2.4, sie werden mit den zugehörigen Routen im Tourenplan X zusammengefasst. Der Tourenplan X wird im folgenden auch als Lösung X bezeichnet.

Das Verfahren des besten Nachfolgers wird mit einer „leeren“ aktuellen Lösung X initialisiert, die weder Fahrten enthält, noch Fahrzeuge einsetzt (1a). Die Menge der noch zuzuordnenden Kundenaufträge U entspricht der Menge aller Kundenaufträge C (1b). Der Ausgangsknoten der ersten zu konstruierenden Tour ist das Depot (1c).

Zunächst wird eine neue Tour begonnen (2). Ausgangsknoten für die Suche des besten Nachfolgers ist immer der zuletzt besuchte Knoten j (3). Es wird zunächst die Menge $U'(i)$ der zulässigen Nachfolgerkunden von i ermittelt (4). Ein Kunde u ist ein zulässiger Nachfolger von i , wenn die Fahrzeugkapazität zur Belieferung ausreicht, der Kunde u pünktlich (vor Ende seines Zeitfensters l_u) erreicht werden kann und die an die Bedienung anschließende Rückkehr ins Depot noch rechtzeitig erfolgen kann. Zur Überprüfung der Zulässigkeit kann die Zulässigkeitsbedingung 2 von Ahn und Shin (1991) verwendet werden. Ist die Menge der zulässigen Nachfolger leer (5), wird die aktuelle Tour durch die Rückkehr ins Depot beendet (9).

Ist $U'(i)$ nicht leer, wird zunächst derjenige Kunde j ermittelt, der den geringsten Abstandswert $Abstand_{ij}$ zum zuletzt eingefügten Kunden i aufweist (6) und anschließend an die aktuelle Tour angehängt (7). Wenn mehrere

¹⁷vgl. Solomon (1987), S. 256

Algorithmus 4 Nearest Neighbor Heuristik

1. Initialisiere:
 - a) $x_{ij}^z := 0 \forall i, j \in C_0, z \in Z; \zeta_z := 0 \forall z \in Z, z := 0$
 - b) $U := C$
 - c) $j := 0$
 2. $z := z + 1, \zeta_z := 1$ (neue Tour)
 3. $i := j$
 4. bestimme $U'(i)$
 5. wenn $U'(i) = \emptyset$ gehe zu (9)
 6. $j \in \operatorname{argmin} \{ \text{Abstand}_{iu} \mid u \in U'(i) \}$
 7. $x_{ij}^z := 1, U := U \setminus \{j\}$
 8. gehe zu (3)
 9. $j := 0, x_{ij}^z := 1$
 10. wenn $U \neq \emptyset$ gehe zu (2)
 11. Ergebnis: Tourenplan X
-

Kunden den gleichen Abstandswert aufweisen, wird ein beliebiger Kunde mit minimalem Abstandswert ausgewählt. Der Abstandswert Abstand_{ij} wird für statisch-deterministische Tourenplanungsprobleme ohne Zeitfensterrestriktionen in der Regel mit der geographischen Distanz zwischen den einzelnen Kunden sowie den Kunden und dem Depot gleichgesetzt. Für Tourenplanungsprobleme mit Zeitfensterrestriktionen ist es sinnvoll neben den Entfernungen auch zeitliche Beziehungen zwischen den Kundenaufträgen zu berücksichtigen.

Die Suche unter allen jeweils zulässigen Nachfolgern $U' \in C$ wird solange weitergeführt (8), bis sich kein Kunde mehr in zulässiger Weise einfügen lässt ($U' = \emptyset$). Die Lösung X wird solange ergänzt, bis alle Kundenaufträge einer Tour zugeordnet worden sind (10). Als Ergebnis erhält man einen zulässigen Tourenplan X (11).

Solomon (1987) schlägt folgende Vorschrift zur Berechnung des Abstandswerts Abstand_{ij} für statisch-deterministische Tourenplanungsprobleme vor (vgl. Algorithmus 5).¹⁸

¹⁸vgl. Solomon (1987), S. 257

Algorithmus 5 Abstandswert nach Solomon (1987) für VRPTW

1. $faz_u := \max\{e_u, y_i + s_i + d_{iu}\}$
2. $err_{iu} := faz_u - (y_i + s_i)$
3. $dri_{iu} := l_u - (y_i + s_i + d_{iu})$
4. $\kappa_1 + \kappa_2 + \kappa_3 = 1$ und $\kappa_1 \geq 0, \kappa_2 \geq 0, \kappa_3 \geq 0$
5. Ergebnis: $Abstand_{iu} := \kappa_1 \cdot d_{iu} + \kappa_2 \cdot err_{iu} + \kappa_3 \cdot dri_{iu}$

Zunächst ist der Zeitpunkt faz_u zu bestimmen, zu dem frühestens mit der Bedienung des Kunden u begonnen werden kann (1). Solomon (1987) addiert dabei Größen mit unterschiedlichen Einheiten (Zeiten y_i, s_i und Entfernungen d_{iu}). Da Entfernungen und Fahrzeiten trotz unterschiedlicher Dimensionen gleichgesetzt werden, wird durch Solomon (1987) implizit angenommen, daß ein Zeiteinheit Fahrzeit einer Entfernungseinheit der Entfernung d_{iu} entspricht und somit die Größen addiert werden können. err_{iu} ist die zusätzliche Fahr- und Wartezeit, die anfällt, wenn u direkt im Anschluss an i bedient wird (2). dri_{iu} ist die zeitliche Differenz zwischen Erreichen des Kunden u und dem spätestzulässigen Zeitpunkt, zu dem u erreicht werden muss (3). Sobald dri_{iu} negativ wird, lässt sich u nicht mehr an die aktuelle Tour anhängen. κ_1, κ_2 und κ_3 sind die Gewichtungsfaktoren für die Ermittlung des Abstandswerts (4). Der Abstandswert $Abstand_{iu}$ ist die gewichtete Summe aus der Entfernung zwischen i und u , der zusätzlichen Fahr- und Wartezeit bei u err_{iu} und der zeitlichen Nähe zum Zeitfensterende dri_{iu} (5). Dieser Abstandswert ist für dynamisch-deterministische Tourenplanungsprobleme ungeeignet.

Donati et al. (2003a) beschreiben keine Experimente zum Verfahren des besten Nachfolgers. Der Ameisenkoloniealgorithmus von Donati et al. (2003a) konstruiert allerdings mit Hilfe eines Verfahrens des besten Nachfolgers neue Touren (vgl. Abschnitt 4.4.3). Für die experimentellen Untersuchungen des Verfahrens des besten Nachfolgers in Abschnitt 5 wird der Abstandswert zugrunde gelegt, den Donati et al. (2003a) für dTP vorgeschlagen haben (vgl. Algorithmus 6).¹⁹

Algorithmus 6 Abstandswert nach Donati et al. (2003a) für dTP

1. $w_u := \max\{e_u - t_{iu}(\tau_i) - \tau_i; 0\}$
2. $Abstand_{iu} := (t_{iu}(\tau_i) + w_u) \cdot (l_u - t_{iu}(\tau_i) - \tau_i)$

w_u bezeichnet die Wartezeit, die bei Kunde u anfällt, wenn er direkt im Anschluss

¹⁹vgl. Donati et al. (2003a), S. 13

an i bedient wird (1). Der Abstandswert wird als das Produkt aus erwarteter Fahr- und Wartezeit ($t_{iu}(\tau_i) + w_u$) und zeitlichem Abstand zum Ende des Zeitfensters des Kunden u ($l_u - t_{iu}(\tau_i) - \tau_i$) ermittelt (2). Je geringer die Differenz aus Zeitfensterende des Kunden u und dem Zeitpunkt, zu dem der Kunde u erreicht werden kann, ist, desto geringer fällt der Abstandswert $Abstand_{iu}$ aus. Ebenso ist mit sinkender Summe aus Fahrzeit und Wartezeit ein geringerer Abstandswert zwischen i und u zu erwarten.

Der Nearest-Neighbor Ansatz liefert sehr schnell zulässige Lösungen für Tourenplanungsprobleme, ist leicht verständlich und lässt sich einfach implementieren. Allerdings lässt die Lösungsqualität ohne ergänzende Nachbarschaftssuche deutlich zu wünschen übrig.²⁰

4.2.3 Insertion-Heuristik

Neben dem Verfahren des besten Nachfolgers schlägt Solomon (1987) mehrere sequentielle Einfügeheuristiken vor, wobei die als „I1“ bezeichnete Heuristik für die in seiner Arbeit untersuchten Problemstellungen die besten Ergebnisse liefert.²¹ Solomon (1987) stellt das Verfahren nur verbal dar, die formale Darstellung in Algorithmus 7 basiert auf der Beschreibung der Einfügeheuristik von Solomon (1987) durch Ahn und Shin (1991).²²

U bezeichnet die Menge der noch nicht zugeordneten Kundenaufträge. Die „Einfügekosten“ $c(i, u, j)$ approximieren die Auswirkung auf den Zielfunktionswert, wenn der Kunde u in der aktuellen Tour zwischen den Kunden oder das Depot i und den Kunden oder das Depot j eingefügt wird.

Es werden sequentiell neue Touren konstruiert.²³ Die Einfügeheuristik wird mit einer „leeren“ aktuellen Lösung X initialisiert. Weder sind Fahrten enthalten, noch wird irgendein Fahrzeug eingesetzt (1a). Die Menge der noch zuzuordnenden Kundenaufträge U entspricht zu Beginn des Verfahrens der Menge aller Kundenaufträge C (1b).

Die Konstruktion von Touren beginnt mit der Auswahl eines Startkunden u .²⁴ Derjenige Kunde u aus der Menge der noch nicht zugeordneten Knoten U wird ausgewählt, der die größte Entfernung zum Depot besitzt: $u \in \operatorname{argmax} \{d_{0u} \mid u' \in U\}$ (3).²⁵ Anschließend wird der Tourenzähler z erhöht, der Kunde u aus der Menge U entfernt und eine

²⁰vgl. Solomon (1987), S. 262; siehe auch Kapitel 5

²¹vgl. Solomon (1987), S. 260

²²vgl. Ahn und Shin (1991), S. 396f.

²³vgl. hierzu und im folgenden Solomon (1987), S. 257

²⁴vgl. hierzu und im folgenden Solomon (1987), S. 259

²⁵Solomon (1987) schlägt als mögliche Alternative vor, denjenigen Kunden u als Startkunden für eine neue Tour zu wählen, der das früheste Zeitfensterende l_u besitzt: $u \in \operatorname{argmin} \{l_{u'} \mid u' \in U\}$.

Algorithmus 7 Insertion-Heuristik

1. Initialisiere:
 - a) $x_{ij}^z := 0 \forall x_{ij}^z \in X; \zeta_z := 0 \forall z \in Z; z := 0$
 - b) $U := C$
2. Iteration: wenn $U = \emptyset$ gehe zu (10)
3. wähle $u \in \operatorname{argmax} \{d_{0u'} \mid u' \in U\}$
4. $z := z + 1, \zeta_z := 1, x_{0u}^z := 1, x_{u0}^z := 1, U := U \setminus \{u\}, c^* := 0$
5. wenn $c^* = +\infty$ gehe zu (2)
6. $c^* := +\infty$
7. prüfe für alle Kunden $u \in U$ und alle Fahrten $(i, j) \in T_z$ der aktuellen Tour
 - a) wenn einfügen von u zwischen i und j unzulässig, überspringe (7b)
 - b) bestimme $c(i, u, j)$, wenn $c(i, u, j) < c^*$ dann $i^* := i, j^* := j, u^* := u, c^* := c(i, u, j)$
8. wenn $c^* \neq +\infty$ dann $U := U \setminus \{u^*\}, x_{i^*j^*}^z := 0, x_{i^*u^*}^z := 1, x_{u^*j^*}^z := 1$
9. gehe zu (5)
10. Ergebnis: Tourenplan X

Pendeltour $(0, u, 0)$ zu dem Startkunden u erzeugt (4). Die aktuelle Tour wird solange durch Kundenaufträge ergänzt, bis kein Kundenauftrag mehr in die aktuelle Tour eingefügt werden kann (5) bis (9).

Zunächst werden die kleinsten bekannten Einfügekosten c^* unendlich hoch angesetzt. Es ist noch keine Position bekannt, an der ein unzugeordneter Kundenauftrag $u \in U$ eingefügt werden kann (6). Es wird für jeden unzugeordneten Kundenauftrag u und jede Position zwischen zwei direkt nacheinander zu erfüllenden Kundenaufträgen i und j in der aktuellen Tour untersucht (7), ob der betrachtete Kundenauftrag zwischen i und j eingefügt werden kann, ohne daß der resultierende Tourenplan unzulässig wird (7a). Eine Position ist dann zulässig, wenn der Kundenauftrag u zwischen i und j bedient werden kann und die dadurch entstehende neue Tour zulässig ist. Zur Beschleunigung der Zulässigkeitsüberprüfung kann an dieser Stelle die Zulässigkeitsbedingung 1 von Ahn und Shin (1991) verwendet werden.²⁶ Anschließend werden die Einfügekosten $c(i, u, j)$ ermittelt (7b). Die Ermittlung der Einfügekosten für dynamisch-deterministische Tourenplanungsprobleme wird in Algorithmus 8

²⁶vgl. Abschnitt 4.1

beschrieben. Sind die Einfügekosten $c(i, u, j)$ kleiner als die kleinsten bekannten Einfügekosten c^* , liegt eine neue beste Kombination von Kundenauftrag und Position vor (7b). Sobald alle Kombinationen von Kundenaufträgen und Positionen überprüft worden sind und sofern es einen zulässigen Kundenauftrag u^* gibt ($c^* \neq +\infty$), wird der Kundenauftrag u^* an die Position zwischen i^* und j^* in die aktuelle Tour eingefügt (8). Diese Einfügevorgänge werden solange wiederholt (9), wie sich Knoten in die aktuell betrachtete Tour einfügen lassen. Gilt $c^* = +\infty$, wurde kein Auftrag $u \in U$ gefunden, der auf zulässige Weise an irgendeine Position in der aktuellen Tour eingefügt werden kann (5). In diesem Fall wird eine neue Tour begonnen, sofern nicht bereits alle Kunden zugeordnet worden sind (2). Sobald alle Kunden einer Tour zugeordnet worden sind ($U = \emptyset$), endet das Verfahren. X ist dann ein zulässiger Tourenplan (10).

Die Einfügekosten $c(i, u, j)$, wie sie von Solomon (1987) für das VRPTW vorgeschlagen werden, vermengen räumliche und zeitliche Distanzen.²⁷ Für das dynamisch-deterministische Tourenplanungsproblem wird daher auf die Modifikation der Einfügeheuristik von Gietz (1994) zurückgegriffen.²⁸ Die Verfahren von Solomon (1987) und Gietz (1994) unterscheiden sich nur bezüglich der Einfügekosten. Algorithmus 8 beschreibt die Ermittlung der Einfügekosten $c(i, u, j)$ für dTP nach Gietz (1994), wie sie auch in der experimentellen Analyse in Kapitel 5 verwendet werden.²⁹ Dabei ist $c_{11}(i, u, j)$ die Differenz aus zusätzlich anfallender Fahrzeit, wenn

Algorithmus 8 „Einfügekosten“ für dTP nach Gietz (1994)

1. $fez_u := \max(e_u, \tau_i + t_{iu}(\tau_i)) + s_u$
 2. $c_{11}(i, u, j) := t_{iu}(\tau_i) + t_{uj}(fez_u) - \kappa_1 \cdot t_{ij}(\tau_i)$
 3. $faz_j^{neu} := \max(e_j, fez_u + t_{uj}(fez_u))$
 4. $c_{12}(i, u, j) := \begin{cases} faz_j^{neu} - y_0^z & \text{wenn } j = 0 \\ faz_j^{neu} - y_j & \text{sonst} \end{cases}$
 5. $c(i, u, j) = -(\kappa_2 \cdot d_{0u} - \kappa_3 \cdot c_{11}(i, u, j) - \kappa_4 \cdot c_{12}(i, u, j))$
-

der Kundenauftrag u zwischen i und j eingefügt wird, und der mit κ_1 gewichteten Fahrzeiterparnis, die durch den Einfügevorgang erzielt werden kann (1). Die Fahrzeiten sind abhängig vom Zeitpunkt τ_i , zu dem der Kunde i wieder verlassen wird bzw. vom Zeitpunkt fez_u , zu dem der einzufügende Kunde u wieder verlassen wird. Der Zeitpunkt fez_u zu dem der Kunde u frühestens wieder verlassen werden kann, entspricht dem frühestmöglichen Beginn der Bedienung des Kunden u zuzüglich der Servicezeit s_u (2). Der Kunde oder das Depot j kann frühestens erreicht werden,

²⁷vgl. Solomon (1987), S. 257

²⁸vgl. Gietz (1994), S. 146 und 171

²⁹vgl. Gietz (1994), S. 146 und S. 171

nachdem der Kunde u bedient worden ist und das Lieferfahrzeug anschließend von u zu j gefahren ist (3). $c_{12}(i, u, j)$ entspricht der Zeitspanne, um die der Beginn der Bedienung des Kunden j (y_j) bzw. das Ende der Tour im Depot j (y_0^z) durch den Einfügevorgang verschoben wird (4). Die Einfügekosten setzen sich aus der mit κ_2 gewichteten Summe aus Entfernung des Kunden u vom Depot d_{0u} , der mit κ_3 gewichteten modifizierten Fahrzeitveränderung $c_{11}(i, u, j)$ und der mit κ_4 gewichteten Veränderung des Belieferungszeitpunkts des Kunden j oder Rückkehrzeitpunkts im Depot j $c_{12}(i, u, j)$ zusammen (5). Je weiter der Kunde u vom Depot entfernt ist und je geringer die zusätzlich anfallende Fahrzeit und die Veränderung des Belieferungszeitpunkts von j ausfällt, desto kleiner sind die Einfügekosten $c(i, u, j)$.

4.3 Verbesserungsverfahren für Tourenplanungsprobleme

Verbesserungsverfahren führen, ausgehend von einer zulässigen Lösung, eine Folge von Operationen durch, welche die gegebene Lösung iterativ verbessern, bis ein Abbruchkriterium erreicht wird. In den Abschnitten 4.3.1 und 4.3.2 werden einfache Verbesserungsverfahren für dTP vorgestellt.

4.3.1 2-opt Nachbarschaftssuche

Die 2-opt Nachbarschaftssuche ist bei Tourenplanungsproblemen auf die Betrachtung einer einzelnen Tour beschränkt.³⁰ Wird ein Tausch von zwei Fahrten zwischen unterschiedlichen Touren zugelassen, erhält man die 2-opt* Nachbarschaft.³¹ Die 2-opt* Nachbarschaft ist ein Spezialfall der Cross-Exchange Nachbarschaft (vgl. auch Abbildung 4.6 auf S. 73).

Der 2-opt Nachbarschaftsoperator wird wie folgt definiert: Eine Tour T_z ist dann mit einer Tour $T_{z'}$ benachbart, wenn $T_{z'}$ durch das Ersetzen zweier Fahrten von T_z erzeugt werden kann. Dabei wird der Abschnitt zwischen den beiden ersetzten Fahrten umgekehrt, d.h. alle Fahrten des Abschnitts zwischen den neu eingefügten Fahrten werden durch entgegengerichtete Fahrten ersetzt. In Algorithmus 9 wird der 2-opt Algorithmus für die Nachbarschaftssuche innerhalb einer Tour z formal dargestellt.

Das 2-opt Verbesserungsverfahren durchsucht ausgehend von einer Ausgangslösung systematisch dessen Nachbarschaft (2), (3). Zwei Fahrten werden aus der Tour entfernt (a.), der dazwischenliegende Abschnitt umgekehrt (b) und die Tour durch zwei neue

³⁰vgl. hierzu und im folgenden Wendt (1995) S. 21

³¹vgl. Potvin und Rousseau (1995), S. 1434-1436

Fahrten wieder ergänzt (c). Die neu entstandene Tour T_z^{neu} wird auf Zulässigkeit überprüft (d). In jeder Iteration wird die erste gefundene, zulässige Nachbarlösung, die einen besseren Zielfunktionswert aufweist als die aktuelle Lösung, ausgewählt (first-fit) (e) und von dieser neuen Lösung ausgehend erneut die Nachbarschaft systematisch durchsucht. Wird keine verbesserte Lösung gefunden oder ist die neu entstandene Tour unzulässig, wird die Suche ausgehend von der aktuell besten bekannten Lösung T_z^* fortgesetzt (f).

Das Verfahren endet, wenn die Nachbarschaft vollständig durchsucht wurde und keine Nachbarlösung gefunden wurde, die besser als die aktuelle Lösung ist (4). Die ermittelte Lösung wird als 2-optimal bezeichnet.

Algorithmus 9 2-opt innerhalb einer Tour

1. Setze $T_z^{neu} := T_z$ und $T_z^* := T_z$
 2. Für alle $pos_1 := 0(1) |T_z^{neu}| - 2$ {
 3. Für alle $pos_2 := pos_1 + 2(1) |T_z^{neu}|$ {
 - a) Entferne die Fahrten $(c_{pos_1}^z, c_{(pos_1+1)}^z)$ und $(c_{pos_2}^z, c_{(pos_2+1)}^z)$ aus der Tour T_z^{neu}
 - b) Kehre die Belieferungsreihenfolge des Abschnitts $(c_{pos_1+1}^z, \dots, c_{pos_2}^z)$ um, indem alle Fahrten des Abschnitts durch die entgegengerichteten Fahrten ersetzt werden.
 - c) Vervollständige die Tour T_z^{neu} durch das Einfügen der Fahrten $(c_{pos_1}^z, c_{pos_2}^z)$ und $(c_{(pos_1+1)}^z, c_{(pos_2+1)}^z)$
 - d) Überprüfe T_z^{neu} auf Zulässigkeit. Ist die Tour T_z^{neu} unzulässig, gehe zu (f.)
 - e) Ist T_z^{neu} besser als die beste bekannte Tour T_z^* , setze $T_z^* := T_z^{neu}$ und starte erneut bei (2)
 - f) $T_z^{neu} := T_z^*$
 - } (next pos_2)
 - } (next pos_1)
 4. Ergebnis: Die Tour T_z^* ist eine 2-optimale Tour.
-

Bei dynamisch-deterministischen Tourenplanungsproblemen mit Zeitfensterrestriktionen können durch das Umkehren der Belieferungsreihenfolge zahlreiche unzulässige Nachbarlösungen entstehen.³² Daher wird die 2-opt Nachbarschaftssuche nicht näher betrachtet.

³²vgl. Potvin und Rousseau (1995), S. 1434

4.3.2 Cross-Exchange Nachbarschaftssuche

Die Cross-Exchange Nachbarschaft wurde von Taillard et al. (1997) für statisch-deterministische Tourenplanungsprobleme eingeführt.³³ Eine Lösung X_2 ist dann Nachbar der Lösung X_1 , wenn X_2 aus X_1 erzeugt werden kann, indem zwei disjunkte Abschnitte von Kundenaufträgen vertauscht werden. Dabei wird auf beiden Abschnitten die Belieferungsreihenfolge der Kunden beibehalten (vgl. Abbildung 4.2).

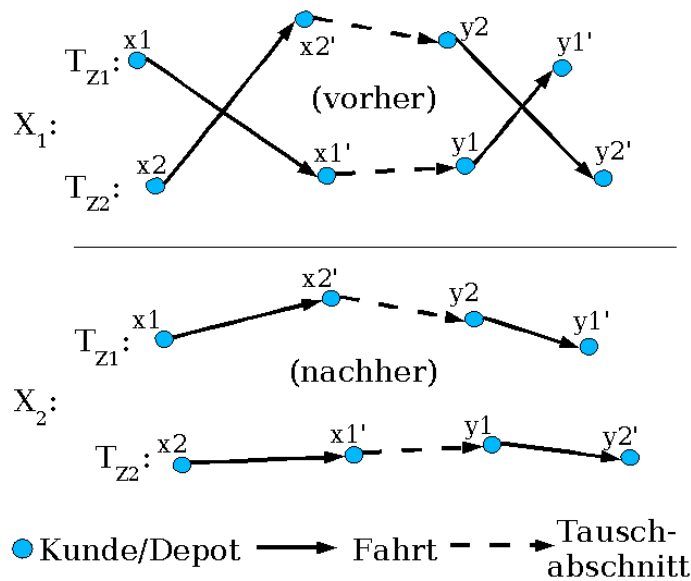


Abbildung 4.2: Cross-Exchange: Tausch von zwei Abschnitten zwischen zwei Touren (vgl. Taillard et al. (1997) S. 173)

Das Cross-Exchange Verbesserungsverfahren (im folgenden CX) durchsucht systematisch die Cross-Exchange Nachbarschaft. Eine formale Darstellung des CX-Verfahrens findet sich in Algorithmus 10 (first-fit Variante) und Algorithmus 11 (best-fit Variante).

Die zulässige Ausgangslösung, mit der die CX-Nachbarschaftssuche beginnt, wird mit $X^{aktuell}$ bezeichnet. Mit X^{best} wird die beste bekannte Lösung bezeichnet. Ausgehend von einer zulässigen Lösung $X^{aktuell}$ werden zwischen zwei (nicht notwendigerweise unterschiedlichen) Touren T_{z_1} und T_{z_2} des Tourenplans $X^{aktuell}$ zwei Abschnitte vertauscht. Der Knoten x_1 bezeichnet den letzten Knoten ($x_1 \in C_0$), der vor zu dem zu tauschenden Abschnitt in der Tour T_{z_1} angeordnet ist. x'_1 ist der direkt auf x_1 folgende Knoten ($x'_1 \in C_0$) auf der Tour T_{z_1} . y'_1 bezeichnet den Knoten der Tour T_{z_1} , der als erstes nach dem zu tauschenden Abschnitt folgt ($y'_1 \in C_0$), y_1 ist der Vorgängerknoten von y'_1 auf der Tour T_{z_1} ($y_1 \in C_0$). Wenn $x_1 = y_1$ entspricht, ist der zu tauschende Abschnitt der Tour T_{z_1} „leer“, d.h. es werden keine Kunden aus der Tour

³³vgl. hierzu und im folgenden Taillard et al. (1997), S. 173f.

T_{z_1} entfernt. Bei nicht-leeren Touren ist x'_1 der erste Kunde auf dem aus der Tour T_{z_1} zu entfernenden Abschnitt, y_1 der letzte Kunde des Abschnitts. Analog sind x_2 , x'_2 , y_2 und y'_2 die Bezeichner der Knoten, die den Abschnitt auf der Tour T_{z_2} spezifizieren.

Algorithmus 10 Cross-Exchange first-fit

1. $X^{best} := X^{aktuell}$
 2. für $z_1 := 1(1)anz(X^{aktuell})$ { (neue Iteration)
 3. für $pos_{11} := 0(1) |T_{z_1}|$ {
 4. für $pos_{21} := pos_{11}(1) |T_{z_1}|$ {
 5. für $z_2 := z_1(1)anz(X^{aktuell})$ {
 6. für $pos_{12} := 0(1) |T_{z_2}|$ {
 7. für $pos_{22} := pos_{12}(1) |T_{z_2}|$ {
 8. Tausch überprüfen
 - a) $x_1 := c_{pos_{11}}^{z_1}$; $x'_1 := c_{pos_{11}+1}^{z_1}$; $y_1 := c_{pos_{21}}^{z_1}$; $y'_1 := c_{pos_{21}+1}^{z_1}$;
 $x_2 := c_{pos_{12}}^{z_2}$; $x'_2 := c_{pos_{12}+1}^{z_2}$; $y_2 := c_{pos_{22}}^{z_2}$; $y'_2 := c_{pos_{22}+1}^{z_2}$
 - b) $X^{neu} := X^{aktuell} \cup \{(x_1, x'_1), (y_2, y'_1), (x_2, x'_1), (y_1, y'_2)\}$
 $\setminus \{(x_1, x'_1), (y_1, y'_1), (x_2, x'_2), (y_2, y'_2)\}$
 - c) wenn X^{neu} unzulässig, dann gehe zu (9)
 - d) wenn X^{neu} besser als X^{best} , dann $X^{best} := X^{neu}$,
nächste Iteration: gehe zu (2)
 9. } next pos_{22}
 10. } next pos_{12}
 11. } next z_2
 12. }next pos_{21}
 13. } next pos_{11}
 14. } next z_1
 15. Ergebnis: Die Lösung X^{best} ist eine CX-optimale Lösung.
-

Zunächst entspricht die beste bekannte Lösung der aktuellen Lösung (1). Alle Touren T_{z_1} und T_{z_2} der aktuellen Lösung werden systematisch untersucht (2)-(7). Je zwei Fahrten (x_1, x'_1) , (y_1, y'_1) aus T_{z_1} werden mit zwei Fahrten (x_2, x'_2) , (y_2, y'_2) aus T_{z_2} vertauscht (8). Dadurch wird der Abschnitt zwischen x_1 und y'_1 aus Tour T_{z_1} zwischen x_2 und y'_2 in Tour T_{z_2} eingefügt, gleichzeitig wird der Abschnitt, der ursprünglich

Algorithmus 11 Cross-Exchange best-fit

1. $X^{best} := X^{aktuell}$
 2. für $z_1 := 1(1)anz(X^{aktuell})$ { (neue Iteration)
 3. für $pos_{11} := 0(1) |T_{z_1}|$ {
 4. für $pos_{21} := pos_{11}(1) |T_{z_1}|$ {
 5. für $z_2 := z_1(1)anz(X^{aktuell})$ {
 6. für $pos_{12} := 0(1) |T_{z_2}|$ {
 7. für $pos_{22} := pos_{12}(1) |T_{z_2}|$ {
 8. Tausch überprüfen
 - a) $x_1 := c_{pos_{11}}^{z_1}; x'_1 := c_{pos_{11}+1}^{z_1}; y_1 := c_{pos_{21}}^{z_1}; y'_1 := c_{pos_{21}+1}^{z_1};$
 $x_2 := c_{pos_{12}}^{z_2}; x'_2 := c_{pos_{12}+1}^{z_2}; y_2 := c_{pos_{22}}^{z_2}; y'_2 := c_{pos_{22}+1}^{z_2}$
 - b) $X^{neu} := X^{aktuell} \cup \{(x_1, x'_2), (y_2, y'_1), (x_2, x'_1), (y_1, y'_2)\}$
 $\setminus \{(x_1, x'_1), (y_1, y'_1), (x_2, x'_2), (y_2, y'_2)\}$
 - c) wenn X^{neu} unzulässig, dann gehe zu (9)
 - d) wenn X^{neu} besser als X^{best} , dann $X^{best} := X^{neu}$
 9. } next pos_{22}
 10. } next pos_{12}
 11. } next z_2
 12. }next pos_{21}
 13. } next pos_{11}
 14. } next z_1
 15. wenn $X^{best} \neq X^{aktuell}$ setze $X^{aktuell} := X^{best}$, gehe zu (2)
 16. Ergebnis: Die Lösung X^{best} ist eine CX-optimale Lösung.
-

zwischen x_2 und y'_2 in Tour T_{z_2} bedient werden sollte, in Tour T_{z_1} zwischen x_1 und y'_1 eingefügt (8b) (vgl. Abbildung 4.2). Die durch den Tausch entstehende Lösung X^{neu} ist nicht notwendigerweise eine zulässige Lösung (8c). Zur Überprüfung der Zulässigkeit kann die modifizierte Zulässigkeitsbedingung 3 aus Abschnitt 4.1 herangezogen werden. Ist X^{neu} zulässig und gleichzeitig eine bessere Lösung als X^{best} , wird X^{best} durch X^{neu} ersetzt (8d).

An dieser Stelle besteht zwischen der first-fit und der best-fit Variante von CX ein Unterschied. Wenn die erste Verbesserung akzeptiert wird (CX-first-fit), dann wird sofort die nächste Iteration, ausgehend von X^{best} begonnen ((8d) in Algorithmus (10)). Ist X^{neu} nicht besser als X^{best} , wird die nächsten Kombinationen von Fahrten und Touren untersucht. Wird je Iteration die größte Verbesserung angestrebt (CX-best-fit), wird vorläufig die neue beste Lösung als X^{best} gespeichert und mit der systematischen Suche in der Nachbarschaft fortgefahren. Sind alle möglichen Tauschvorgänge von Abschnitten untersucht worden, liegt bei CX-first-fit mit X^{best} eine CX-optimale Lösung vor (Zeile 15 in Algorithmus 10). Bei CX-best-fit ist zunächst zu überprüfen, ob die aktuell beste bekannte Lösung X^{best} in der letzten Iteration ermittelt wurde ($X^{best} \neq X^{aktuell}$, Zeile 15 in Algorithmus 10) oder ob in der letzten Iteration alle Tauschvorgänge überprüft worden sind, ohne eine verbesserte Lösung zu finden. Nur wenn in der letzten Iteration keine Verbesserung mehr erzielt worden ist, liegt eine CX-optimale Lösung vor (16), ansonsten wird in der nächsten Iteration ausgehend von X^{best} versucht, eine verbesserte Lösung zu finden.

Abschnitte von Kundenaufträgen können zwischen Touren getauscht (vgl. Abbildung 4.2) und von einer Tour in eine andere Tour verschoben werden (vgl. Abbildung 4.3).

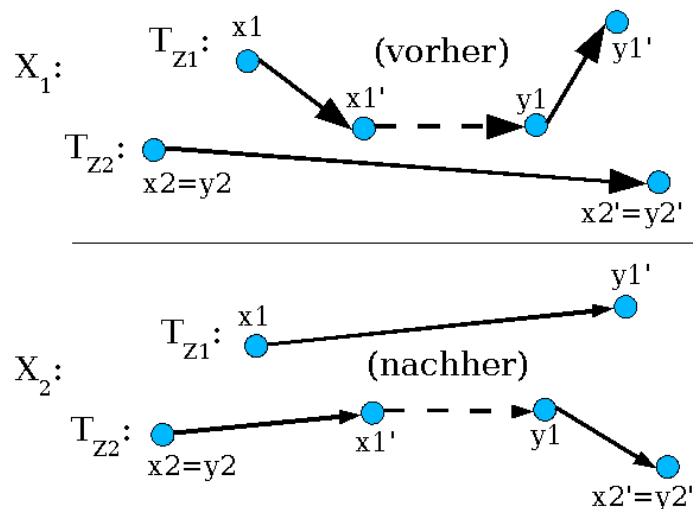


Abbildung 4.3: Or-Opt als Spezialfall von Cross-Exchange
 (Quelle: Taillard et al. (1997), S. 173)

Je nach Definition umfasst Cross-Exchange auch die Möglichkeit einen Abschnitt

innerhalb einer Tour zu verschieben oder zwei Abschnitte in einer Tour zu vertauschen (vgl. Abbildung 4.4).³⁴ Gilt $z_1 = z_2$, wird die Belieferungsreihenfolge der Kundenaufträge innerhalb einer Tour verändert.

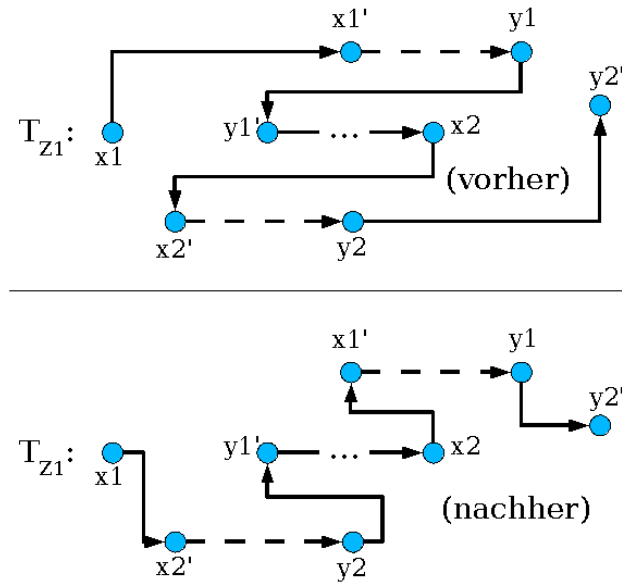


Abbildung 4.4: Cross-Exchange innerhalb einer Tour

Dabei müssen allerdings disjunkte Abschnitte vertauscht werden. Sind die zu vertauschenden Abschnitte nicht disjunkt, entstehen Kurzzyklen, die nicht mit dem Depot verbunden sind (vgl. Abbildung 4.5).

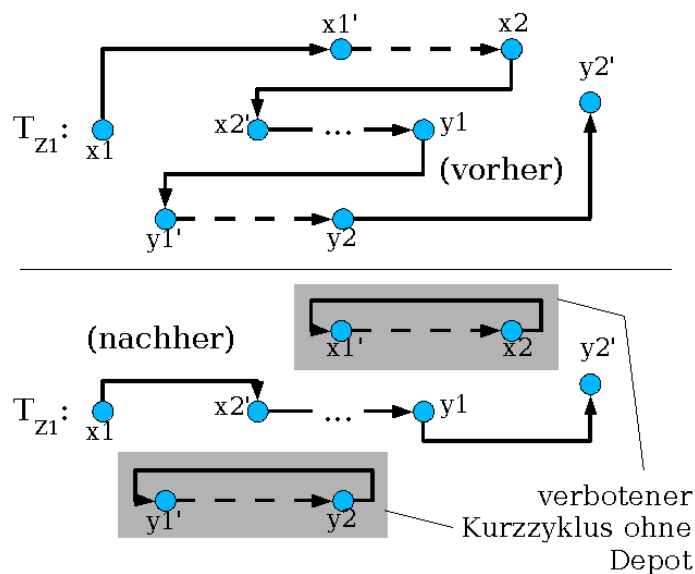


Abbildung 4.5: Unzulässige Cross-Exchange Operation

³⁴vgl. Taillard et al. (1997), S. 176 definiert CX auch für Tauschvorgänge innerhalb einer Tour, wobei nur zwei Fahrten vertauscht werden; vgl. Ichoua et al. (2003), S. 386 definiert CX nur für Tauschvorgänge zwischen unterschiedlichen Touren

Verbleibt beim Verschieben eines Abschnitts aus einer Tour in eine andere Tour eine „leere“ Tour (keine Kunden sind zu bedienen, $x_{00}^z = 1$), wird die leere Tour aus der Lösung entfernt ($\zeta_z := 0$). Die Nachbarlösung benötigt eine Tour weniger als die Ausgangslösung zur Belieferung aller Kunden.³⁵ Umgekehrt ist eine Erhöhung der Anzahl der benötigten Touren durch CX-Nachbarschaftsoperatoren nicht möglich, da „leere“ Touren nicht zum Tausch zur Verfügung stehen. Sind beide zu vertauschenden Abschnitte leer, findet kein Tausch statt. Die Cross-Exchange Nachbarschaft ist also nicht symmetrisch. Eine Lösung mit insgesamt geringeren Auslieferungskosten, die eine Tour mehr als die aktuelle Lösung erfordern würde, ist nicht erreichbar.³⁶

Die zu tauschenden Abschnitte können unterschiedlich lang oder auch leer sein. Sind beide Abschnitte leer, liegt kein Tauschvorgang vor. Taillard et al. (1997) beschränkt die CX-Nachbarschaft durch eine Vorgabe von $CX_{\text{abschnitt}}^{\max} = 7$ Kundenaufträgen die auf den zu vertauschenden Abschnitten liegen dürfen.³⁷ Dadurch gilt für (2)-(7) aus Algorithmus 10 und aus Algorithmus 11 nicht länger, daß „alle“ Abschnitte der jeweiligen Tour untersucht werden. Vielmehr werden nur die Abschnitte mit null bis sieben Kunden geprüft, formal gilt also zusätzlich $pos_{21} - pos_{11} \leq CX_{\text{abschnitt}}^{\max}$ und $pos_{22} - pos_{12} \leq CX_{\text{abschnitt}}^{\max}$. Im Rahmen der experimentellen Analyse in Kapitel 5 wird der Einfluss unterschiedlicher Vorgaben für die Abschnittslänge näher untersucht.

In der Literatur finden sich häufig Nachbarschaftsdefinitionen, die Spezialfälle von Cross-Exchange darstellen. Der von Bent und Van Hentenryck (2004) als Relocation bezeichnete Nachbarschaftsoperator generiert die Teilmenge der CX-Nachbarschaft, die sich ergibt, wenn x_1 und x_2 in unterschiedlichen Touren liegen, der Abschnitt zwischen x_1' und y_1 aus nur einem Kunden besteht und der zweite Abschnitt leer ist ($x_2 = y_2$).³⁸ Mit Exchange generieren Bent und Van Hentenryck (2004) den Teil der CX-Nachbarschaft, der sich ergibt, wenn x_1 und x_2 in unterschiedlichen Touren liegen, und der Abschnitt zwischen x_1' und y_1 sowie der Abschnitt zwischen x_2' und y_2 aus je nur einem Kunden besteht.³⁹ Werden Relocation und Exchange nach der Definition von de Backer et al. (2000) verwendet, dürfen x_1 und x_2 auch aus der gleichen Tour stammen.⁴⁰

Taillard et al. (1997) stellt dar, daß die Cross-Exchange Nachbarschaft die or-opt Nachbarschaft enthält (vgl. Abbildung 4.3). Die Menge der CX-Nachbarn, die resultieren, wenn $z_1 \neq z_2$ (x_1 und x_2 liegen in unterschiedlichen Touren) gilt und der

³⁵vgl. Taillard et al. (1997), S. 183

³⁶Wird die lexikographische Zielfunktion verwendet, die in Abschnitt 2.2.3.2 beschrieben wird, ist die primäre Zielsetzung die Minimierung der Tourenzahl. In diesem Fall stellt die Asymmetrie der Nachbarschaft kein Problem dar, da eine Lösung mit mehr Touren in jedem Fall eine Verschlechterung der Lösungsqualität bedeutet.

³⁷vgl. Taillard et al. (1997), S. 182

³⁸vgl. Bent und Van Hentenryck (2004), S. 518

³⁹vgl. Bent und Van Hentenryck (2004), S. 518

⁴⁰vgl. de Backer et al. (2000), S. 5f.

Abschnitt $(x2', y2)$ leer ist ($pos_{12} = pos_{22}$), entspricht der Menge or-opt Nachbarn.⁴¹ Donati et al. (2003a) bezeichnen die or-opt Nachbarschaft als Branch-Relocate.⁴²

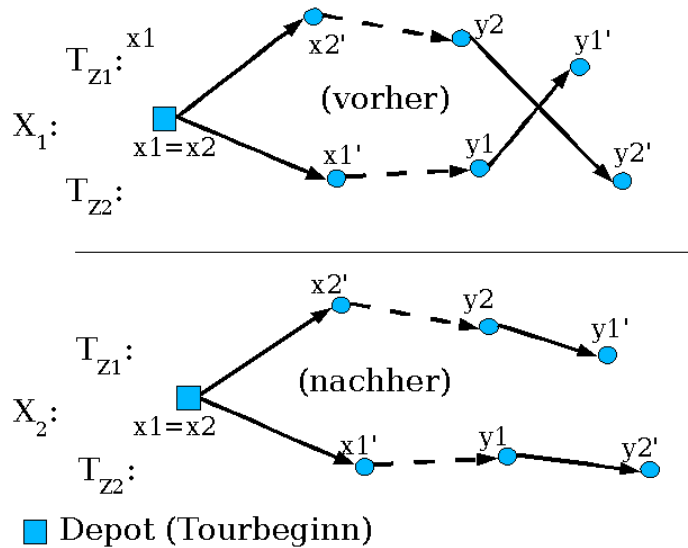


Abbildung 4.6: 2-opt* als Spezialfall von Cross-Exchange
(Quelle: Taillard et al. (1997), S. 173)

Die von Potvin und Rousseau (1995) und Taillard et al. (1997) als 2-opt*-Nachbarschaft und von Bent und Van Hentenryck (2004) als Crossover bezeichnete Nachbarschaft ist die Teilmenge der CX Nachbarschaft, die sich ergibt wenn $x1$ und $x2$ (alternativ $y1'$ und $y2'$) dem Depot entsprechen ($pos_{11} = pos_{12} = 0$) (alternativ $pos_{21} = pos_{22} = 0$). 2-opt* ist ein Tausch von zwei Fahrten aus zwei unterschiedlichen Touren.⁴³ Die Fahrt $(x1, x1')$ entspricht dann $(x1, x2')$ und die Fahrt $(y2, x2')$ ist äquivalent zu $(x2, x2')$ (alternativ $(y1, y1') \hat{=} (y1, y2')$ und $(y2, y2') \hat{=} (y2, y1')$), vgl. Abbildung (4.6). Im Unterschied zu 2-opt wird die Belieferungsreihenfolge auf den zu tauschenden Abschnitten nicht umgekehrt.

4.4 Ameisenkolonialgorithmen für Tourenplanungsprobleme

4.4.1 Die ACO-Metaheuristik

Ameisen sind einfache Insekten, die in Kolonien zusammenleben.⁴⁴ Durch Zusammenarbeit zeigen Ameisen erstaunliche Fähigkeiten: Sie finden beispielsweise

⁴¹vgl. Taillard et al. (1997), S. 173

⁴²vgl. Donati et al. (2003a), S. 18f.

⁴³vgl. Potvin und Rousseau (1995), S. 1434-1436; Taillard et al. (1997), S. 173; vgl. Bent und Van Hentenryck (2004), S. 518

⁴⁴vgl. hierzu und im folgenden Stützle (1998), S. 23f.

zuverlässig den kürzesten Weg zwischen ihrem Bau und einer Futterquelle. Ameisen tauschen Informationen über Pheromone aus, chemische Substanzen, die in unterschiedlichen Mengen abgelegt werden können, um beispielsweise einen Weg zu markieren. Einzelne Ameisen scheinen nahezu zufällig und ziellos umherzuirren. Entdecken sie eine Pheromonspur, folgen sie ihr zuverlässig zu einer Futterquelle oder in den Ameisenbau. Die Wahrscheinlichkeit, daß eine Ameise einer bestimmten Pheromonspur folgt, ist dabei umso größer, je mehr Pheromon abgelegt worden ist. Folgt eine Ameise einer Pheromonspur zu einer Futterquelle, kann sie dabei wiederum Pheromon ablegen und somit einen selbst verstärkenden Prozess anstoßen: Je mehr Ameisen in der Vergangenheit einen bestimmten Weg gewählt haben, desto attraktiver wird der Weg in der Zukunft für die Ameisen sein. Die Pheromone verflüchtigen sich im Zeitablauf. Abbildung 4.7 zeigt schematisch die Herausbildung eines vorteilhaften Wegs zwischen dem Ameisenbau und einer Futterquelle.

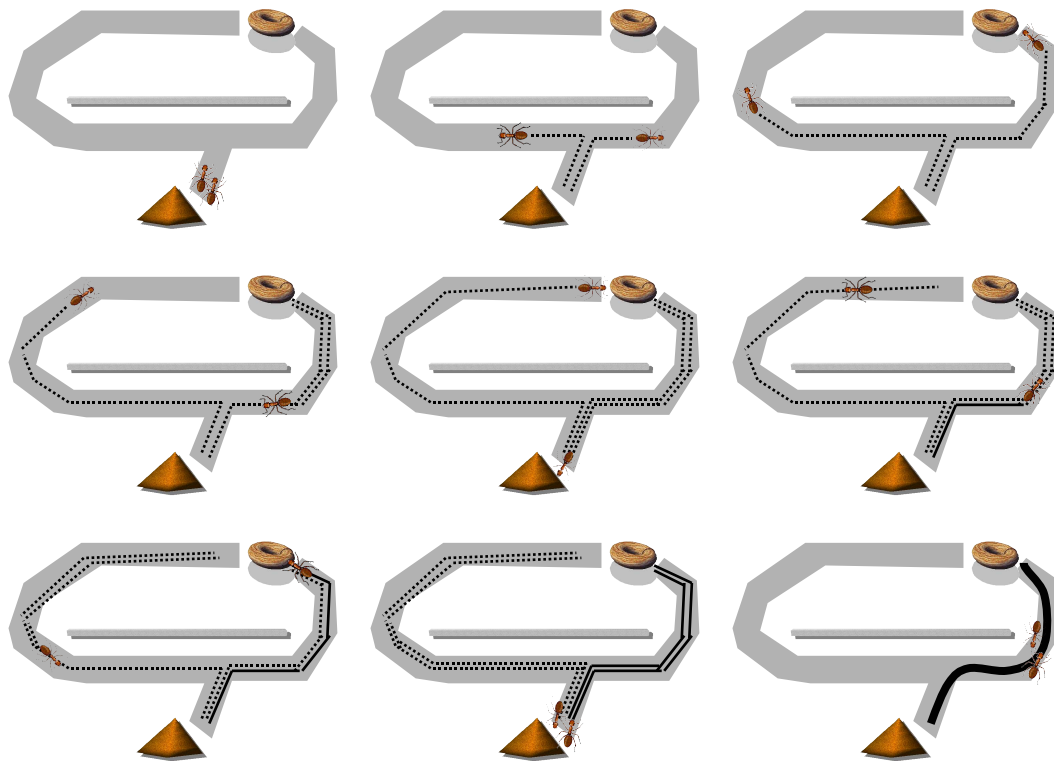


Abbildung 4.7: Prinzip der Ausbildung einer Ameisenstraße vom Bau zur Futterquelle

Beide Ameisen verlassen gleichzeitig den Bau (Pyramide in Abbildung 4.7) auf der Suche nach Futter (Kringel in Abbildung 4.7). Die Wahrscheinlichkeit den linken und den rechten Weg zu wählen ist zunächst gleich groß - es liegen keinerlei wegweisenden Pheromoninformationen vor. Beide Ameisen hinterlassen bei der Suche pro Zeiteinheit gleich viel Pheromon und bewegen sich mit der gleichen Geschwindigkeit. Die Ameise, die den rechten Weg wählt, erreicht die Futterquelle als erste und kehrt entlang der

eigenen Pheromonspur auf dem gleichen Weg zurück in den Bau. In dem Zeitpunkt, in dem die zweite Ameise die Futterquelle erreicht hat, ist die erste Ameise bereits wieder im Bau - auf dem rechten Weg liegt inzwischen doppelt so viel Pheromon wie auf dem linken Weg. Eine weitere Ameise verlässt den Bau. Die Wahrscheinlichkeit, daß sie den rechten Weg wählt ist deutlich größer als die Wahrscheinlichkeit den linken Weg zu wählen, da die Pheromonspur stärker ist. In dem Zeitpunkt, in dem die zweite Ameise wieder in den Bau zurückkehrt, ist die Pheromonspur für den rechten Weg erheblich stärker als die des linken Wegs. Die Wahrscheinlichkeit, bei einer weiteren Suche nach Futter der rechte Weg gewählt wird, ist inzwischen relativ groß. Dadurch daß Pheromone im Zeitablauf verdunsten, sinkt zudem die Attraktivität, den linken Weg zu wählen, so daß schließlich der rechte Weg mit Pheromonen markiert ist und die Ameisenstraße bildet.

Inspiziert durch das Verhalten realer Ameisen bei der Futtersuche wurden von Dorigo (1992) in seiner Dissertation Ameisenkolonialgorithmen als Lösungsprinzip für komplexe Optimierungsprobleme vorgestellt.⁴⁵ Seitdem ist eine Vielzahl von Veröffentlichungen erschienen, die sich mit Ameisenkolonialgorithmen (ACO-Algorithmen) beschäftigen und unterschiedlichste Problemstellungen mit dieser Metaheuristik lösen.⁴⁶

Bei Ameisenkolonialgorithmen handelt es sich um metaheuristische Verfahren, die ein algorithmisches Pendant zu den Pheromonspuren der Ameisen für den globalen Informationsaustausch zwischen einzelnen Agenten, die als künstliche Ameisen bezeichnet werden, verwenden. Für Tourenplanungsprobleme ist es zweckmäßig den Fahrten x_{ij}^z Pheromonwerte zuzuordnen, um gute Lösungsbestandteile verstärkt zur Konstruktion von neuen Lösungen heranzuziehen.

In jeder Iteration konstruiert eine Ameisenkolonie zulässige Lösungen zum Tourenplanungsproblem. Eine Ameisenkolonie besteht aus einer oder mehreren künstlichen Ameisen, die sukzessive nacheinander Rundreisen zur Belieferung aller Kunden konstruieren. Durch die wiederholte Konstruktion wird versucht, möglichst unterschiedliche und vielfältige Lösungen zu ermitteln. Die künstlichen Ameisen greifen dabei auf eine lokale, kurzsichtige Komponente („Visibility“) und globale, längerfristige Informationen („Pheromone“) zurück. Diejenigen Lösungsbestandteile werden bevorzugt ausgewählt, denen ein hoher Pheromonwert zugeordnet ist und die eine hoch bewertete lokale Komponente vorweisen. Die lokale Komponente approximiert dabei die Vorteilhaftigkeit des Einfügens eines bestimmten Knotens ans Ende der aktuell in Konstruktion befindlichen Tour. Häufig wird als lokale Komponente der Kehrwert des Abstandswerts, der für das Verfahren des besten Nachfolgers benutzt

⁴⁵vgl. Dorigo (1992)

⁴⁶vgl. für einen Überblick Dorigo und Di Caro (1999), S. 17; Dorigo und Gambardella (1999), S. 142; Dorigo und Stützle (2004), S. 39f.

wird, verwendet, da als Zielsetzung die Minimierung der Gesamtentfernung und/oder Gesamtfahrzeit angenommen wird. Die Pheromonwerte bilden die Vorteilhaftigkeit von Teillösungen ab, Bestandteile von guten Lösungen aus vergangenen Iterationen werden nach wenigen Iterationen einen höheren Pheromonwert aufweisen als Bestandteile von schlechten Lösungen.

Im Anschluss an die Lösungskonstruktion wird in der Regel ein lokales Suchverfahren eingesetzt, um die konstruierte Lösung zu verbessern. Schließlich wird die Pheromonmatrix aktualisiert, um die erneute Pheromonablage und das Verflüchtigen der bereits vorhandenen Pheromone nachzubilden: Zunächst wird das Pheromon für alle aktuellen und potentiellen Lösungsbestandteile um einen konstanten Faktor gesenkt, um die Pheromonverdunstung abzubilden. Anschließend können Ameisen die Lösungsbestandteile (hier Fahrten $x_{ij}^z = 1$) der von ihr gefundenen Lösung verstärken, indem die korrespondierenden Pheromonwerte erhöht werden. Je besser die erzielte Lösung ist, desto mehr Pheromon wird abgelegt.

Der Informationsaustausch über vorteilhafte Teillösungen oder Lösungsbestandteile wird also nicht direkt zwischen den Ameisen vorgenommen. Die Kommunikation erfolgt vielmehr über Pheromonwerte, die während und nach der Konstruktion angepasst werden können (lokales und globales Pheromonupdate). Nach wiederholter Durchführung konvergieren geeignete Ameisenkolonialgorithmen gegen gute Lösungen. Dabei sollte eine frühzeitige Konvergenz gegen lokale Optima vermieden werden, andererseits ist eine gewisse Konvergenz notwendig, damit das Verfahren überhaupt in vorteilhafte Bereiche des Lösungsraums vordringen kann.

Stützle und Hoos (1997) haben in Experimenten zum Travelling-Salesman-Problem herausgefunden, daß bei der Konstruktion von neuen Lösungen durch einen Ameisenkolonialgorithmus nach einigen Iterationen ausgehend von einem bestimmten Knoten häufig nur ein einziger Knoten als Nachfolger mit sehr hoher Wahrscheinlichkeit in Frage kommt, während andere Nachfolger nur einer sehr geringe Auswahlwahrscheinlichkeit aufweisen.⁴⁷ Die Suche stagniert bzw. konvergiert dadurch vorzeitig auf einem relativ schlechten Niveau. Daher schlagen Stützle und Hoos (1997) eine Beschränkung des Wertebereichs der Pheromonwerte vor. Eine Beschränkung nach unten erhöht die Auswahlwahrscheinlichkeit von vermeintlich „schlechteren“ Lösungsbestandteilen, die Beschränkung der Pheromonwerte nach oben senkt gleichzeitig die Wahrscheinlichkeit der Auswahl vermeintlich guter Bestandteile aus vorhergehenden Lösungen. Bestandteile von guten Lösungen werden dabei immer noch häufiger ausgewählt als die übrigen Bestandteile, die Tendenz zu einer vorzeitigen Konvergenz wird allerdings deutlich verringert.

⁴⁷vgl. hierzu und im folgenden Stützle und Hoos (1997), S. 310

4.4.2 Der Ansatz von Gambardella et al. (1999) für statische Tourenplanungsprobleme mit Zeitfensterrestriktionen

4.4.2.1 Zielsetzung und Lösungsabbildung

Gambardella et al. (1999) bezeichnen ihren Ameisenkolonialgorithmus als MACS-VRPTW (multiple ant colony system for the vehicle routing problem with time windows).⁴⁸ Zwei voneinander unabhängige Ameisenkolonien (ACS-TIME und ACS-VEI) konstruieren gleichzeitig neue Lösungen zum statisch-deterministischen Tourenplanungsproblem mit Zeitfensterrestriktionen. Primäre Zielsetzung des Ameisenkolonialgorithmus ist die Minimierung der Anzahl der Touren, die zur Belieferung aller Kundenaufträge erforderlich sind.⁴⁹ Aus der Menge der Lösungen mit minimaler Tourenzahl ist diejenige Lösung auszuwählen, welche die Gesamtfahrzeit minimiert. Die Kolonie ACS-VEI versucht die Anzahl der Touren zu minimieren, die Kolonie ACS-TIME verfolgt das Ziel, die Gesamtfahrzeit zu minimieren.

Jede der beiden Kolonien besitzt eine eigene, unabhängige Pheromonmatrix. Ein Element ϕ_{ij} der Pheromonmatrix (im folgenden als Pheromonwert bezeichnet) bildet dabei die Vorteilhaftigkeit ab, direkt im Anschluss an den Knoten i den Knoten j zu besuchen. Die aktuell beste bekannte Lösung wird im folgenden mit X^{best} bezeichnet. Die Kolonie ACS-VEI versucht zulässige Lösungen mit $(anz(X^{best}) - 1)$ Touren zu konstruieren.⁵⁰ Die Kolonie ACS-TIME konstruiert Lösungen mit $anz(X^{best})$ Touren mit möglichst geringer Gesamtfahrzeit.

Gambardella et al. (1999) überführen das statisch-deterministische Tourenplanungsproblem mit Zeitfensterrestriktionen (VRPTW) in ein Handlungsreisendenproblem (TSP) mit Zeitfenster- und Kapazitätsrestriktionen, indem das Depot \mathcal{Z} -fach „dupliziert“ wird.⁵¹ \mathcal{Z} ist dabei eine Vorgabe, die die Anzahl der Touren, die maximal zur Verfügung stehen, festlegt. \mathcal{Z} Knoten mit den gleichen Zeitfenstern und Koordinaten wie das Depot werden dem Problem hinzugefügt. Die Menge der \mathcal{Z} Knoten und das Depot werden im folgenden als Depotknoten bezeichnet. Es ist zu beachten, daß jeder Knoten (auch jeder Depotknoten) nur genau einmal besucht werden darf. Lösungen werden als Rundreise abgebildet. Eine Lösung ist zulässig, wenn alle Kunden- und Depotknoten genau einmal unter Berücksichtigung aller Zeitfenster- und Kapazitätsrestriktionen besucht werden.

Da sich die Vorgabe der Tourenzahl zwischen den beiden Kolonien unterscheidet sind somit auch die zu der jeweiligen Kolonie gehörenden Pheromonmatrizen

⁴⁸vgl. Gambardella et al. (1999), S. 63

⁴⁹vgl. Gambardella et al. (1999), S. 63

⁵⁰vgl. hierzu und im folgenden Gambardella et al. (1999), S. 69f.

⁵¹vgl. hierzu und im folgenden Gambardella et al. (1999), S. 71

unterschiedlich groß sind, kann der Informationsaustausch nicht über eine gemeinsame Pheromonmatrix erfolgen. Der Informationsaustausch zwischen den beiden Kolonien erfolgt über den Austausch der jeweils aktuell besten bekannten Lösung. In Algorithmus 12 ist die Funktionsweise des Koordinationsmechanismus MACS dargestellt.

Algorithmus 12 Funktionsweise MACS-VRPTW

1. Initialisierung
 - a) ermittle X^{init} mit Verfahren des besten Nachfolgers
 - b) setze $X^{best} := X^{init}$
2. Wiederhole {
3. starte parallelen Prozess ACS-VEI mit der Vorgabe $\mathcal{Z} = anz(X^{best}) - 1$
4. starte parallelen Prozess ACS-TIME mit der Vorgabe $\mathcal{Z} = anz(X^{best})$
5. solange ACS-VEI und ACS-TIME laufen: prüfe, ob neue beste Lösung ermittelt wurde{
6. wenn Lösung X^{best} verbessert wurde, stelle neue beste bekannte Lösung X^{best} ACS-VEI und ACS-TIME zur Verfügung
7. wenn die Anzahl der benötigten Touren verringert wurde ($anz(X^{best})$ ist gesunken), beende Prozesse ACS-VEI und ACS-TIME, gehe zu (3)
8. wenn Abbruchkriterium für MACS erreicht wurde, beende ACS-VEI und ACS-TIME, gehe zu (10)
9. }
10. } Suche beendet, Ergebnis: X^{best}

(vgl. Gambardella et al. (1999), S. 68)

Zunächst wird zur Initialisierung des Ameisenkolonialgorithmus MACS eine zulässige Ausgangslösung für das VRPTW mit Hilfe des Verfahrens des besten Nachfolgers ermittelt (1).⁵² Sie wird im folgenden als X^{init} bezeichnet. Die aktuell beste bekannte Lösung X^{best} wird zunächst gleich X^{init} gesetzt.

Anschließend werden wiederholt die beiden Ameisenkolonien ACS-VEI und ACS-TIME aufgerufen, die parallel (gleichzeitig) nach neuen Lösungen zum VRPTW suchen ((3) und (4)). Eine detaillierte Beschreibung der Kolonien ACS-VEI und ACS-TIME findet sich in Algorithmus 13 und Algorithmus 14.

Das Hauptprogramm MACS (die aufrufende Instanz) überprüft anschliessend in regelmäßigen Abständen, ob eine neue beste Lösung von einer der beiden Kolonien

⁵²vgl. hierzu und im folgenden Gambardella et al. (1999), S. 68

ermittelt wurde (5). Sobald eine der beiden Kolonien eine neue beste Lösung findet, wird diese neue beste Lösung X^{best} an die aufrufende Instanz (MACS) zurückgegeben und unverzüglich beiden Kolonien zur Verfügung gestellt. Die beste bekannte Lösung steht sofort beiden Kolonien zur Verfügung (6).

Wird durch eine der beiden Kolonien eine neue zulässige Lösung ermittelt, die eine geringere Gesamtfahrzeit als $travel(X^{best})$ aufweist, wird X^{best} durch die neu ermittelte Lösung ersetzt, die beiden Kolonien setzen die Suche fort. ACS-VEI und ACS-TIME werden in allen Folgeiterationen das Pheromonupdate mit der neuen besten Lösung durchführen (und nur dadurch werden Informationen zwischen den Kolonien ausgetauscht). Wird durch die neue beste Lösung eine Tour eingespart, werden beide Kolonien beendet (7).⁵³ Für beide Kolonien sind in diesem Fall neue Pheromonmatrizen zu erzeugen und die Kolonien neu zu initialisieren. Die Pheromonmatrizen sind zu verkleinern, da die Anzahl der Depotkopien durch $anz(X^{best})$ determiniert wird. Wird ein (von Gambardella et al. (1999) nicht näher spezifiziertes) Abbruchkriterium erreicht, werden die beiden Kolonien beendet (8).

4.4.2.2 Kolonie ACS-VEI

In Algorithmus 13 wird die Funktionsweise von ACS-VEI skizziert. ACS-VEI beginnt die Suche nach neuen, besseren Lösungen mit der Initialisierung der Pheromonmatrix und der Ermittlung einer Teillösung $X^{best'}$ (1). $X^{best'}$ bezeichnet die Teillösung, bei der die meisten Kundenaufträge bei gegebener Tourenzahl eingeplant werden. Sie wird mit einem Verfahren des besten Nachfolgers für das TSP mit Zeitfenster- und Kapazitätsrestriktionen initialisiert, wobei die Konstruktion der Lösung abgebrochen wird, sobald alle Depotknoten in die Lösung aufgenommen wurden. Die Lösung $X^{best'}$ ist daher normalerweise unzulässig, da nicht alle Kundenaufträge bedient werden. ϕ_{init} entspricht dem Pheromonwert, der bei der Initialisierung der Pheromonmatrix jedem Element der Matrix zugewiesen wird.⁵⁴

$$\phi_{init} := \frac{1}{((|C_0| + \mathcal{Z}) \cdot travel(X^{init}))}$$

ϕ_{init} ist der Kehrwert aus der Anzahl der Kunden- und Depotknoten, multipliziert mit der Gesamtfahrzeit $travel(X^{init})$ der Startlösung.

In jeder Iteration konstruieren nacheinander alle $anz_{ameisen}$ Ameisen der Ameisenkolonie neue Teillösungen zum VRPTW (3). $anz_{ameisen}$ ist dabei ein Parameter, der festlegt, wieviele Suchvorgänge innerhalb einer Iteration durchgeführt

⁵³d.h. neue Tourenzahl-Vorgabe, neue, leere sofortiger Pheromonmatrizen, Abbruch der aktuellen Konstruktionsphase derjenigen Kolonie, die nicht X^{best} gefunden hat.

⁵⁴vgl. Gambardella et al. (1999), S. 67

Algorithmus 13 Funktionsweise ACS-VEI

1. Initialisiere Pheromonmatrix, $X^{best'}$ und IN_u
2. Iteration: {
 3. für alle $anz_{Ameisen}$ Ameisen {
 4. konstruiere neue (Teil-)Lösung $X^{aktuell}$
 5. ergänze $X^{aktuell}$ durch Einfügeverfahren
 6. setze $IN_u := IN_u + 1$ für alle nicht belieferten Kunden $u \notin X^{aktuell}$
 7. ermittle Anzahl der in aktueller Teillösung belieferten Kunden
 - a) wenn in $X^{aktuell}$ mehr Kunden zugeordnet wurden als in $X^{best'}$, setze $X^{best'} := X^{aktuell}$ und $IN_u := 0 \forall U \in C$
 - b) wenn $X^{aktuell}$ eine zulässige Lösung ist, setze $X^{best} := X^{aktuell}$ und gebe X^{best} an MACS zurück
 8. } next Ameise
 9. führe Pheromonupdate mit $X^{best'}$ durch
 10. führe Pheromonupdate mit X^{best} durch
 11. nächste Iteration - gehe zu (3)
 12. } Ende Iteration

(vgl. Gambardella et al. (1999), S. 70)

werden sollen (wieviele Ameisen suchen sollen), bevor ein globales Pheromonupdate erfolgt. Jede Ameise beginnt, ausgehend von einem zufällig ausgewählten Depotknoten, sequentiell eine Rundreise durch alle Kunden- und Depotknoten zu konstruieren. Dabei wird in jedem Konstruktionsschritt ein Knoten an die letzte Position der Rundreise angehängt („besucht“). Mit jedem Besuch eines Depotknotens wird eine Tour des Tourenplans beendet.

Auswahl des nächsten zu besuchenden Knotens:

Mit U' wird im folgenden die Menge der zulässigen Nachfolgerknoten bezeichnet. Ein Nachfolgerknoten ist genau dann zulässig, wenn die Kapazitätsrestriktion eingehalten wird, der Knoten noch innerhalb seines Zeitfensters erreicht werden kann und anschließend an die Bedienung des Depot vor dem Ende des Planungshorizonts erreicht werden kann. Ausgehend vom zuletzt besuchten Knoten i wird für alle zulässigen Nachfolgerknoten j ein Wahrscheinlichkeitswert p_j ermittelt:

$$p_j := \frac{\phi_{ij} \cdot \eta_{ij}^\beta}{\sum_{j \in U'} \phi_{ij} \cdot \eta_{ij}^\beta}$$

Der Wahrscheinlichkeitswert p_j wird durch die lokale Komponente η_{ij} und durch den Pheromonwert ϕ_{ij} beeinflusst. β ist der Gewichtungsfaktor für die lokale Komponente η_{ij} .⁵⁵ Je höher β ist, desto mehr Einfluss hat die lokale Komponente η_{ij} auf die Wahl des nächsten zu besuchenden Kunden. Mit einer Wahrscheinlichkeit von q_0 (Ausbeutungsparameter) wird derjenige Knoten j ausgewählt, für den der Wahrscheinlichkeitswert p_j maximal ist, mit einer Wahrscheinlichkeit von $(1 - q_0)$ erfolgt die Auswahl des Nachfolgers j zufallsgesteuert, wobei dann p_j der Auswahlwahrscheinlichkeit für den Knoten j entspricht.⁵⁶ Auf das Pheromonupdate wird in Abschnitt 4.4.2.4 und auf die lokale Komponente in Abschnitt 4.4.2.5 näher eingegangen.

Die Konstruktionsphase endet, sobald alle Depotknoten in die Rundreise aufgenommen worden sind (4).⁵⁷ Sind nicht alle Knoten in der Rundreise enthalten, ist die konstruierte Teillösung X^{aktuell} unzulässig, da nicht alle Kunden besucht werden. Enthält X^{aktuell} im Anschluß an die Konstruktionsphase nicht alle Kundenaufträge, wird versucht die fehlenden Kundenaufträge durch ein dem INS-Verfahren von Solomon (1987) ähnlichen Verfahren in die aktuelle Tour einzufügen (5).⁵⁸

⁵⁵vgl. hierzu und im folgenden Gambardella et al. (1999), S. 66

⁵⁶vgl. hierzu und im folgenden Gambardella et al. (1999), S. 66

⁵⁷vgl. hierzu und im folgenden Gambardella et al. (1999), S. 66

⁵⁸vgl. hierzu und im folgenden Gambardella et al. (1999), S. 72f.

Es ist zu überprüfen, ob in $X^{aktuell}$ mehr Kundenaufträge zugeordnet werden konnten, als in $X^{best'}$.⁵⁹ IN_j wird ausschließlich von der Kolonie ACS-VEI verwendet und gibt an, seit wie vielen Iterationen der Kundenauftrag j nicht Bestandteil einer Teillösung war. Nachdem eine Ameise aus ACS-VEI eine unvollständige Teillösung konstruiert hat, werden für alle Knoten u , die nicht in der Teillösung $X^{aktuell}$ enthalten sind, die zugehörigen Werte IN_u um eins erhöht (6). Wird ein bestimmter Kundenauftrag häufig nicht berücksichtigt, steigt die Attraktivität dieses Kunden während der Konstruktion neuer Lösungen an.

$X^{best'}$ wird durch $X^{aktuell}$ ersetzt, wenn in $X^{aktuell}$ eine größere Anzahl Kunden als in $X^{best'}$ zugeordnet wurden (7). Findet ACS-VEI eine neue Teillösung $X^{aktuell}$ mit weniger unzugeordneten Kundenaufträgen als $X^{best'}$, wird IN_u für alle Kundenaufträge u auf 0 zurückgesetzt und die neue Lösung $X^{best'} := X^{aktuell}$ gespeichert (7a). Wird eine neue, zulässige Lösung ermittelt (d.h. alle Kunden wurden in die Rundreise aufgenommen, kein Kundenauftrag bleibt unzugeordnet), wird X^{best} an die aufrufende Instanz MACS zurückgegeben (7b). In diesem Fall liegt in jedem Fall eine neue beste Lösung X^{best} vor, da die Anzahl der Touren im Vergleich zur bisher besten bekannten zulässigen Lösung reduziert wird.

Sobald alle Ameisen der aktuellen Iteration die Lösungsfindung abgeschlossen haben, erfolgt ein globales Pheromonupdate mit allen Fahrten aus der Lösung $X^{best'}$ (9) und aus X^{best} (10), vgl. dazu Abschnitt 4.4.2.4. Die Kolonie ACS-VEI endet nicht selbstständig, sondern wird durch den Abbruch der Suche durch die aufrufende Instanz MACS beendet.

⁵⁹vgl. hierzu und im folgenden Gambardella et al. (1999), S. 69

4.4.2.3 Kolonie ACS-TIME

Algorithmus 14 Funktionsweise ACS-TIME

1. Initialisiere Pheromonmatrix
2. Iteration: {
3. für alle $anz_{Ameisen}$ Ameisen {
4. konstruiere neue (Teil-)Lösung $X^{aktuell}$
5. ergänze $X^{aktuell}$ durch Einfügeverfahren
6. wenn $X^{aktuell}$ nicht zulässig, verwirfe $X^{aktuell}$ und gehe zu (9)
7. führe CX-Verbesserungsverfahren auf $X^{aktuell}$ durch
8. wenn $X^{aktuell}$ besser als X^{best} ist, setze $X^{best} := X^{aktuell}$ und gebe X^{best} an MACS zurück
9. } next Ameise
10. führe Pheromonupdate mit X^{best} durch
11. nächste Iteration - gehe zu (3)
12. } Ende Iteration

(vgl. Gambardella et al. (1999), S. 69)

Die Ermittlung neuer Lösungen funktioniert bei ACS-TIME im Wesentlichen analog zu ACS-VEI, sie wird in Algorithmus 14 dargestellt. Jede Ameise konstruiert sequentiell (Teil-)Lösungen zum VRPTW (4), die anschließend durch ein Einfügeverfahren ergänzt werden (5). Der Vektor IN_u und die unzulässige Teillösung $X^{best'}$ werden bei der Lösungsfindung nicht verwendet, ACS-TIME verwirft im Unterschied zu ACS-VEI unzulässige Teillösungen $X^{aktuell}$ (6). Wird von einer Ameise eine zulässige Lösung ermittelt (d.h. alle Kunden sind der Rundreise zugeordnet worden), wird mit Hilfe des CX-Verfahrens versucht die Lösung $X^{aktuell}$ zu verbessern (7). Erst danach wird überprüft, ob eine neue beste Lösung X^{best} ermittelt wurde (8). Sobald alle Ameisen einer Kolonie die Konstruktions- und ggf. die Verbesserungsphase beendet haben, folgt ein globales Pheromonupdate mit der besten bekannten Lösung X^{best} (10). Wie ACS-VEI wird auch ACS-TIME nicht selbstständig, sondern durch die aufrufende Instanz MACS beendet.

4.4.2.4 Pheromonupdate

Während eine Ameise eine neue Lösung konstruiert, erfolgt ein lokales Pheromonupdate. ρ ist die Verdunstungsrate für das Pheromon, ϕ_{init} der Wert, mit dem die Pheromonmatrix initialisiert wird. Der Pheromonwert der zuletzt in die Lösung $X^{aktuell}$ aufgenommenen Fahrt von i zu j wird nach jedem Konstruktionsschritt angepasst ($\phi_{ij} := (1 - \rho) \cdot \phi_{ij} + \phi_{init} \cdot \rho$), damit für andere Ameisen der gleichen Kolonie die zuletzt eingefügte Fahrt weniger attraktiv ist und möglichst unterschiedliche Lösungen konstruiert werden.

Im Anschluss an die Konstruktionsphase erfolgt bei ACS-VEI und ACS-TIME ein globales Pheromonupdate mit der besten bekannten zulässigen Lösung X^{best} .⁶⁰ Für alle Fahrten (i, j) , gilt

$$\phi_{ij} := \begin{cases} (1 - \rho) \cdot \phi_{ij} + \frac{\rho}{travel(X^{best})} & \text{wenn } (i, j) \text{ in } X^{best} \text{ enthalten ist} \\ (1 - \rho) \cdot \phi_{ij} & \text{sonst} \end{cases}$$

Der Pheromonwert ϕ_{ij} wird reduziert, um die Verdunstung abzubilden. Gleichzeitig werden für alle Fahrten der Lösung X^{best} die zugehörigen Pheromonwerte erhöht, wobei umso mehr Pheromon abgelegt wird, je niedriger die Gesamtfahrzeit $travel(X^{best})$ der besten bekannten Lösung ausfällt. Für alle Fahrten (i, j) , die nicht in X^{best} enthalten sind, wird der zugehörige Pheromonwert nur reduziert: $\phi_{ij} := (1 - \rho) \cdot \phi_{ij}$.

Die Kolonie ACS-VEI führt ein weiteres globales Pheromonupdate mit den Fahrten aus der Lösung $X^{best'}$ durch.⁶¹ $X^{best'}$ ist die Teillösung, die bei gegebener Tourenzahl die meisten Kundenaufträge einplant. Das Pheromonupdate erfolgt analog zu dem Pheromonupdate auf Basis von X^{best} . Der Zweck dieses Pheromonupdates ist, Fahrten zu verstärken, welche die Belieferung von möglichst vielen Kunden auf einer Rundreise ermöglichen.

4.4.2.5 Lokale Komponente

Die lokale Komponente η_{ij} soll die Auswahlwahrscheinlichkeit von guten Lösungsbestandteilen erhöhen und von schlechten Lösungsbestandteilen reduzieren, unabhängig davon ob sie in vergangenen Iterationen bereits in die Lösung aufgenommen worden sind. Aufgrund von Zeitfensterrestriktionen und daraus resultierenden asymmetrischen Vorteilhaftigkeitsbeziehungen zwischen einzelnen Kundenaufträgen ist es nicht zweckmäßig, wie bei Travelling-Salesman-Problemen üblich, die Inverse der euklidischen Distanz als lokale Komponente zu wählen. Stattdessen wird η_{ij}

⁶⁰vgl. hierzu und im folgenden Gambardella et al. (1999), S. 69 (ACS-TIME) und S. 70 (ACS-VEI)

⁶¹vgl. hierzu und im folgenden Gambardella et al. (1999), S. 70

folgendermaßen bestimmt:⁶²

$$\eta_{ij} := \frac{1}{\max(1; (\max(\tau + t_{ij}; e_j) - \tau) \cdot (l_j - \tau) - IN_j)}$$

$\max(\tau + t_{ij}; e_j)$ ist der frühestmögliche Zeitpunkt, zu dem die Belieferung des Kundenauftrags j beginnen kann. $(\max(\tau + t_{ij}; e_j) - \tau)$ ergibt die Summe aus zusätzlicher Fahr- und Wartezeit, wenn der Kundenauftrag j als nächstes bedient wird. Ein Kundenauftrag der sowohl zeitlich (im Sinne von der frühestmöglichen Bedienung) als auch geographisch (im Sinne von Fahrzeit) nahe bei i liegt, wird einen niedrigen Wert von $(\max(\tau + t_{ij}; e_j) - \tau)$ aufweisen. $(l_j - \tau)$ ist ein Indikator für die „Dringlichkeit“ des nachfolgenden Kundenauftrags: Je kleiner $(l_j - \tau)$ ist, desto weniger Zeit steht noch für die Einplanung des Kundenauftrags zur Verfügung. Wird $(l_j - \tau)$ negativ, lässt sich j nicht mehr in zulässiger Weise einplanen.

$((\max(\tau + t_{ij}; e_j) - \tau) \cdot (l_j - \tau) - IN_j)$ nimmt mit steigendem „zeitlichen Abstand“ zwischen i und j zu und sinkt mit steigender Dringlichkeit des Kundenauftrags j . Für ACS-VEI sinkt der Abstandswert zudem mit der Zahl von Konstruktionsschritten, in denen eine Teillösung konstruiert wurde, in der der Auftrag j nicht enthalten war. Für ACS-TIME gilt immer $IN_j = 0$. Damit keine undefinierten Werte für η_{ij} resultieren können, wird 1 als Mindestwert für den Nenner festgelegt. Mit steigendem $(\max(1; \max(\tau + t_{ij}; e_j) - \tau) \cdot (l_j - \tau) - IN_j)$ sinkt die Vorteilhaftigkeit, den Knoten j nach dem Knoten i zu besuchen.

4.4.3 Der modifizierte Ansatz von Donati et al. (2003a) für dynamisch-deterministische Tourenplanungsprobleme

4.4.3.1 Zielsetzung

Donati et al. (2003a) modifizieren den Ameisenkolonialgorithmus von Gambardella et al. (1999) um dTP mit variablen Fahrzeiten lösen zu können.⁶³ Die primäre Zielsetzung ist die Minimierung der Anzahl der benötigten Touren. Aus der Menge der Lösungen mit minimaler Tourenzahl ist diejenige auszuwählen, welche die Summe der Fahrzeiten minimiert.⁶⁴ Die Modellierung der dynamischen Fahrzeiten wird von Ichoua et al. (2003) übernommen.⁶⁵

Donati et al. (2003a) untersuchen die Vorteilhaftigkeit ihres Ansatzes auf Basis eines realen Problems mit 30 Kundenaufträgen ohne Zeitfenster.⁶⁶ Die in dem zugrunde

⁶²vgl. Gambardella et al. (1999), S. 72

⁶³vgl. Donati et al. (2003a), S. 2

⁶⁴vgl. Donati et al. (2003a), S. 11

⁶⁵vgl. Abschnitt 2.2.2.3

⁶⁶vgl. Donati et al. (2003a), S. 24-26

liegenden Verkehrswegenetzwerk zu wählenden Wege werden vor Beginn der Planung mit Hilfe eines kürzeste-Wege Algorithmus ermittelt und liegen anschließend für alle Fahrten im Planungszeitraum fest.⁶⁷ Das der Tourenplanung zugrunde liegende Verkehrswegenetzwerk umfasst 31 Knoten (30 Kunden + 1 Depot).

Ein Unterschied zwischen dem Ameisenkolonialgorithmus von Gambardella et al. (1999) und dem von Donati et al. (2003a) besteht in der Lösungsabbildung. Das Depot wird von Donati et al. (2003a) nicht dupliziert. Vielmehr darf das Depot mehrfach besucht werden. Weiterhin unterscheidet sich die Pheromonmatrix und die lokale Komponente.

4.4.3.2 Pheromonmatrix

Anstelle einer Pheromonmatrix je Kolonie eine Menge von Pheromonmatrizen verwendet um während der Konstruktion dynamische Fahrzeiten besser zu berücksichtigen.⁶⁸ Im Abhängigkeit vom Zeitpunkt zu dem die Entscheidung getroffen werden muss, welcher Kunde als nächstes zu beliefern ist, variieren die Fahrzeiten zwischen den Kunden. Je nach Zeitpunkt des Verlassens von i können unterschiedliche Kunden vorteilhaft sein.⁶⁹ Es wird für jedes Intervall θ („time-subspace“, „time-slice“) im Planungszeitraum, für das im Zeitablauf unterschiedliche Geschwindigkeiten definiert sind, eine eigene Pheromonmatrix definiert.⁷⁰ Für eine grafische Darstellung dieses Konzepts vgl. Abbildung 4.8.⁷¹ Die mit „time-slice“ beschrifteten Ebenen bilden je ein Intervall ab, für das eine eigene Pheromonmatrix zugrundegelegt wird.

Die Beschreibung der Beziehung der Pheromonmatrizen untereinander erfolgt bei Donati et al. (2003a) nur unscharf und nicht eindeutig, auf Anpassungen beim Pheromonupdate wird nicht eingegangen. Für die experimentelle Analyse in Kapitel 5 wird daher angenommen, daß die zum aktuellen Intervall θ zugehörige Pheromonmatrix durch den Abreisezeitpunkt in i festgelegt wird und bei allen Pheromonupdates nur die Pheromonwerte derjenigen Pheromonmatrix angepasst werden, in deren Intervall die Fahrt von i zu j beginnt.

4.4.3.3 Lokale Komponente

Als lokale Komponente verwenden Donati et al. (2003a) $\eta_{ij}(\tau)$.

$$\eta_{ij}(\tau) := \frac{1}{\max(1; (t_{ij}(\tau) + w_j) \cdot (l_j - (\tau + t_{ij}(\tau))) - IN_j)}$$

⁶⁷vgl. Donati et al. (2003b) S. 24f.

⁶⁸vgl. Donati et al. (2003a), S. 12

⁶⁹Dies gilt allerdings auch für VRPTW, aufgrund von Zeitfensterrestriktionen.

⁷⁰vgl. hierzu und im folgenden Donati et al. (2003a), S. 12f.

⁷¹Quelle: Donati et al. (2003a), S. 12

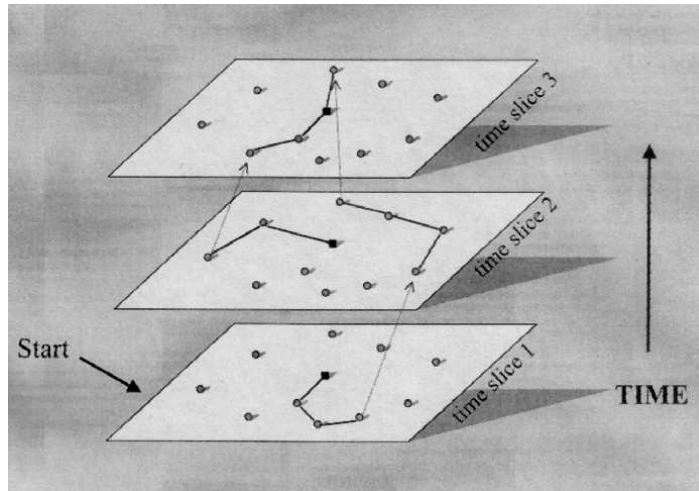


Abbildung 4.8: Aufteilung der Pheromonmatrix in mehrere Intervalle nach Donati et al. (2003a)

Dabei ist $w_j := \max(0; e_j - (t_{ij}(\tau) + \tau))$.⁷² w_j ist die Zeitspanne, die das aktuell betrachtete Fahrzeug beim Kunden j warten muss, bis es mit der Bedienung beginnen kann. $(t_{ij}(\tau) + w_j)$ approximiert die zusätzliche Bediendauer $sched(X)$, die durch die Belieferung des Kunden j direkt nach i zu erwarten ist, $(l_j - (\tau + t_{ij}(\tau)))$ ist die Zeit, die zwischen Zeitfensterende des Kunden j und der Ankunft bei j liegt. IN_j wird wie bei Gambardella et al. (1999) nur im Rahmen der Kolonie ACS-VEI verwendet um die Auswahlwahrscheinlichkeit von Knoten zu erhöhen, die in zahlreichen Iterationen nicht mehr in einer Lösung enthalten waren. Für ACS-TIME gilt $IN_j = 0$ für alle j .

Es ist besonders attraktiv den Kunden j als nächstes zu besuchen, wenn der Nenner von $\eta_{ij}(\tau)$ möglichst klein ausfällt. Dies ist der Fall, wenn die Belieferung des Kunden j möglichst zeitnah erfolgen kann (geringe Fahrzeit und geringe Wartezeit bei j) und der Kunde j dringlich zu bedienen ist, da die Ankunft bei j nahe am Ende seines Zeitfensters liegt. Für ACS-VEI wird die Attraktivität eines Kunden j zusätzlich gesteigert, wenn er seit vielen Iterationen nicht Bestandteil einer Teillösung war.

4.4.3.4 Verbesserungsverfahren

Donati et al. (2003a) schlagen zur Lösungsverbesserung durch ACS-TIME ein mehrstufiges, lokales Suchverfahren vor.⁷³

- 1) einzelne Kunden von einer Tour in eine andere Tour zu verschieben
- 2) je zwei Kunden zwischen Touren zu tauschen
- 3) innerhalb einer Tour einen 2-opt Tausch vorzunehmen

⁷²vgl. Ichoua et al. (2003), S. 13

⁷³vgl. Donati et al. (2003a), S. 18f.

4) einen Abschnitt aus mehreren Kunden von einer Tour in eine andere Tour zu verschieben

5) je zwei Abschnitte aus mehreren Kunden zwischen zwei Touren zu tauschen

Um dabei eine unterschiedliche Suchreihenfolge in jeder Iteration zu ermöglichen, wird die Reihenfolge der Touren in der Lösung zufällig vertauscht.⁷⁴ Diese Änderung der Reihenfolge ist nur dann sinnvoll, wenn angenommen wird, daß nach dem first-fit-Prinzip getauscht wird. Die Nachbarschaft dieses Suchverfahrens wird durch das Cross-Exchange Verfahren, ergänzt durch 2-opt, abgedeckt. Daher wird für die experimentelle Analyse in Kapitel 5 anstelle des mehrstufigen Verfahrens eine CX-Nachbarschaftssuche verwendet.

4.4.3.5 Konstruktion von neuen Lösungen

Die Auswahlwahrscheinlichkeit für die Belieferung des Kunden j direkt im Anschluss an die Belieferung von i , wenn i zum Zeitpunkt τ verlassen wird, ergibt sich als:

$$p_j := \frac{\phi_{ij\theta} \cdot \eta_{ij}(\tau)}{\sum_{j \in U'} \phi_{ij\theta} \cdot \eta_{ij}(\tau)}$$

Analog zu Gambardella et al. (1999) wird mit einer Wahrscheinlichkeit von q_0 derjenige Kunde j als nächstes bedient, für den p_j maximal ist. Mit einer Wahrscheinlichkeit von $(1 - q_0)$ wird der Kunde j , der als nächstes zu bedienen ist zufällig aus U' ausgewählt, wobei die Auswahlwahrscheinlichkeit für einen Kunden j p_j entspricht. Im Unterschied zu Gambardella et al. (1999) verwenden Donati et al. (2003a) keinen Parameter β für die Gewichtung der lokalen Komponente im Verhältnis zum Pheromonwert.

Die Beschreibung des Ameisenalgorithmus von Donati et al. (2003a) weist auf Seite 7 einen Fehler auf: im Gegensatz zum Ansatz für das statisch-deterministische VRPTW von Gambardella et al. (1999), soll nach der Beschreibung von Donati et al. (2003a) das globale Pheromonupdate erst erfolgen, wenn die beiden Kolonien zurückgesetzt worden sind. Folgt man dieser Vorschrift, nehmen die Pheromonwerte bei der Konstruktion von neuen Lösungen kontinuierlich ab, bedingt durch das lokale Pheromonupdate. Solange allerdings keine neue, beste Lösung gefunden wird, wird kein neues Pheromon abgelegt. Nach der Konstruktion einiger schlechter Lösungen wird schließlich nahezu kein Pheromon mehr in der Pheromonmatrix übrig bleiben und die Suche ausschließlich durch die lokale Komponente gesteuert. Das Verfahren wird in einer endlosen Schleife von Iterationen kreisen, ohne bessere Lösungen finden zu können, bis irgendwann durch ein Abbruchkriterium das gesamte Verfahren beendet wird. Wird hingegen,

⁷⁴vgl. Donati et al. (2003a), S. 19

wie von Gambardella et al. (1999) vorgeschlagen, nach jeder Konstruktion neuer Lösungen durch alle Ameisen ein globales Pheromonupdate durchgeführt, wird die Suche durch die bestbekannte Lösung gelenkt werden. Eine Anfrage an die Autoren zu der Richtigkeit dieser Darstellung ist bedauerlicherweise unbeantwortet geblieben. Für die experimentelle Analyse in Kapitel 5.5 wird die Vorgehensweise von Gambardella et al. (1999) für das Pheromonupdate gewählt.

4.4.3.6 Weitere Unterschiede zu Gambardella et al. (1999)

Donati et al. (2003a) schlagen eine Anpassung der Startzeitpunkte im Depot vor.⁷⁵ Dadurch lässt sich möglicherweise die Gesamtfahrzeit senken. Wird allerdings als Zielsetzung die Minimierung der Gesamtbediendauer verfolgt, ist diese Vorgehensweise bei Gültigkeit des FIFO Prinzips nicht von Vorteil, denn jede Verzögerung des Abfahrtszeitpunkts vom Depot führt zu einer steigenden Gesamtbediendauer. Auf diese Anpassung wird daher im Rahmen der experimentellen Analyse in dieser Arbeit verzichtet.

Um die erforderlichen Rechenzeiten zu verkürzen, führen Donati et al. (2003a) einen Vorteilhaftigkeitsparameter γ ein. Nicht jede Lösung $X^{aktuell}$ der Kolonie ACS-TIME führt das Verbesserungsverfahren durch.⁷⁶ Vielmehr werden nur Lösungen verbessert, die vielversprechend erscheinen. Eine Lösung wird als vielversprechend angesehen, wenn nicht mehr als die $(1 + \gamma)$ -fache Anzahl an Fahrzeugen von X^{best} benötigt wird und bei gleicher Tourenzahl wie X^{best} die benötigte Gesamtbediendauer $sched(X^{aktuell})$ nicht das $(1 + \gamma)$ -fache der Gesamtbediendauer $sched(X^{best})$ übersteigt. Jede Lösung $X^{aktuell}$, die besser als die beste bekannte Lösung X^{best} ist, gilt ebenfalls als vielversprechend.

Weder Gambardella et al. (1999) noch Donati et al. (2003a) gehen darauf ein, wie ein zweckmäßiges Abbruchkriterium für ihren Ameisenkolonialgorithmus aussieht. Für die experimentelle Analyse in Kapitel 5 wird der Ameisenkolonialgorithmus nach einer vorgegebenen Anzahl Iterationen, ohne daß eine neue beste Lösung X^{best} gefunden wurde, beendet.

⁷⁵vgl. Donati et al. (2003a), S. 13f.

⁷⁶vgl. hierzu und im folgenden Donati et al. (2003a) S. 6

4.5 Ein neuer Ameisenkolonialgorithmus für dTP (dynACO)

4.5.1 Aufbau

Im folgenden wird ein neuer, als dynACO bezeichneter Ameisenkolonialgorithmus zur Lösung von dTP vorgestellt. Eine einzelne Ameise konstruiert durch sequentielles Anfügen von zufallsgesteuert ausgewählten Kundenaufträgen eine Tour nach der anderen. Wird das Depot an die aktuelle Tour angefügt, ist die aktuelle Tour beendet und wird dem Tourenplan $X^{aktuell}$ hinzugefügt. Sobald alle Kundenaufträge $u \in C$ zugeordnet sind, ist die Konstruktionsphase beendet. Im Anschluss an die Konstruktionsphase werden vielversprechende, nicht CX-optimale Lösungen durch maximal CX_{iter}^{max} Iterationen CX-Nachbarschaftssuche verbessert.

Schließlich erfolgt ein globales Pheromonupdate durch die bis dahin beste bekannte Lösung X^{best} . Ein lokales Pheromonupdate erfolgt nicht. Die Ameise wiederholt die Lösungskonstruktion, bis ein Abbruchkriterium erreicht wird. Als Abbruchkriterium wird die maximale Anzahl Iterationen ohne Verbesserung anz_{iter}^{max} verwendet. In Algorithmus 15 wird formal die Vorgehensweise des Ameisenalgorithmus dynACO beschrieben. Lösungen zum dTP werden als Binärvariablen gespeichert. Gilt $x_{ij}^z = 1$, dann ist in der aktuellen Lösung der Kunde j direkt im Anschluss an den Kunden i durch Fahrzeug z zu beliefern, die Fahrt (i, j) ist Teil des Tourenplans X . Die Konstruktion erfolgt sequentiell, wobei Touren nacheinander konstruiert werden, indem an die jeweils letzte Stelle der aktuell zu konstruierenden Tour ein Kundenauftrag oder das Depot angehängt wird. Für die Zulässigkeitsprüfung wird auf die Zulässigkeitsbedingung 2 aus Abschnitt 4.1 zurückgegriffen.

4.5.2 Initialisierung

Zur Beschleunigung der Lösungsfindung durch die Ameisen und für die Festlegung der minimalen und maximalen Pheromonwerte ϕ^{min} und ϕ^{max} wird vor der ersten Ameisen-Iteration eine Startlösung X^{init} mit Hilfe der Einfügeheuristik INS aus Abschnitt 4.2.3 gebildet (1a). Sie wird als vorläufig beste bekannte Lösung X^{best} verwendet. Die Startwerte und der Wertebereich der Pheromonmatrix werden anschließend initialisiert und der Iterationszähler anz_{inter} auf null gesetzt (1b). Jeder Fahrt (i, j) , $i, j \in C_0$ wird ein Pheromonwert ϕ_{ij} zugeordnet. Der Pheromonwert ist eine positive, reelle Zahl und wird im Laufe des Verfahrens ständig angepasst.

Bei ersten Vorab-Experimenten mit dynACO ließ sich beobachten, daß dynACO ohne eine Beschränkung des Wertebereichs für die Pheromonwerte dazu tendiert

Algorithmus 15 Ameisenkolonialgorithmus dynACO

1. Initialisierung:
 - a) konstruiere Ausgangslösung X^{init} mit Einfügeheuristik, $X^{best} := X^{init}$
 - b) $\phi_{ij} := sched(X^{init}) \forall i, j \in C_0$, $\phi^{min} := \max \left\{ 1; \frac{sched(X^{init})}{|C|} \right\}$, $\phi^{max} := sched(X^{init}) \cdot |C|$, $anz_{iter} := 0$
 2. Iteration: Wiederhole {
 3. neuer Tourenplan $X^{aktuell}$ ($x_{ij}^z := 0 \forall i, j \in C_0$, $\zeta_z := 0 \forall z \in Z$)
 4. Konstruktionsphase (Ameisen-Iteration) {
 5. $z := z + 1$, $\zeta_z := 1$, $i := 0$, $\tau_i := e_0$
 6. sequentielle Tourkonstruktion:
 - a) ermittle U' (Menge aller zulässigen Nachfolger von i)
 - b) wähle j aus U' mit Wahrscheinlichkeit p_j , $p_j = \frac{\phi_{ij} \cdot \eta_{ij}^\pi(\tau)}{\sum_{j \in U'} \phi_{ij} \cdot \eta_{ij}^\pi(\tau)}$
 - c) $x_{ij}^z := 1$, $\tau_j := \max \{e_j; \tau_i + t_{ij}(\tau_i)\} + s_j$
 - d) wenn $j = 0$ gehe zu (5)
 - e) wenn $j \neq 0$ dann $U := U \setminus \{j\}$
 - f) wenn $U = \emptyset$ dann $x_{j0}^z = 1$ (beende Tour), gehe zu (7)
 - g) $i := j$, gehe zu (6a)
 7. } Ende Konstruktionsphase
 8. Verbesserungsphase: {
 9. wenn ($anz(X^{aktuell}) \leq (1 + \gamma) \cdot anz(X^{best})$ und $sched(X^{aktuell}) \leq (1 + \gamma) \cdot sched(X^{best})$) oder ($X^{aktuell}$ besser als X^{best}), dann verbessere X^{neu} mit höchstens CX_{iter}^{max} Iterationen Cross Exchange
 10. wenn $X^{aktuell}$ besser als X^{best} ist, dann $X^{best} := X^{aktuell}$, $anz_{iter} := 0$
 11. wenn $X^{aktuell}$ nicht besser als X^{best} ist, $anz_{iter} := anz_{iter} + 1$
 12. } Ende Verbesserungsphase
 13. Pheromonupdate: {

$$\phi_{ij}^{extra} := \begin{cases} 2 \cdot \frac{sched(X^{init})}{sched(X^{best})} & \text{wenn } (i, j) \in X^{best} \\ 1 & \text{sonst} \end{cases}$$

$$\phi_{ij} := \max \{ \rho \cdot \phi_{ij}; \phi^{min} \} \forall i, j \in C_0, \phi_{ij} := \min \{ \phi_{ij} \cdot \phi_{ij}^{extra}; \phi^{max} \} \forall i, j \in C_0$$
 14. } Ende Pheromonupdate
 15. wenn $anz_{iter} < anz_{iter}^{max}$ gehe zu (3)
 16. } Ende Iteration; Ergebnis: Tourenplan X^{best}
-

vorzeitig auf relativ schlechtem Niveau zu konvergieren. In Anlehnung an das MIN-MAX-Ant-System von Stützle und Hoos (1997) werden daher auch bei dynACO Mindest- und Höchstwerte für den Pheromonwert definiert.⁷⁷ Sie werden aus Vereinfachungsgründen bei dynACO konstant für alle Iterationen des Verfahrens, in Abhängigkeit von der Ausgangslösung X^{init} festgelegt. Diese relativ einfache Beschränkung hat sich als ausreichend erwiesen um ein vorzeitiges Konvergieren des Algorithmus zu verhindern. Die Minimal- und Maximalwerte für die Pheromonmatrix ϕ_{min} und ϕ_{max} werden mit $\phi^{min} := \max \left\{ 1; \frac{sched(X^{init})}{|C|} \right\}$ und $\phi^{max} := sched(X^{init}) \cdot |C|$ festgelegt (1b). Dabei ist $sched(X^{init})$ die Gesamtbediendauer der Startlösung X^{init} . Alle Werte der Pheromonmatrix werden mit $\phi_{ij} := sched(X^{init})$ initialisiert.

4.5.3 Ameisen-Iteration

4.5.3.1 Konstruktionsphase

Mit U wird im folgenden die Menge der noch unzugeordneten Kundenaufträge bezeichnet, mit U' die Menge der zulässigen Nachfolger in der aktuellen Tour. U' enthält immer mindestens das Depot.

Eine einzelne Ameise konstruiert eine neue Lösung (3). Die Ameise konstruiert nacheinander neue Touren ((4) bis (7)). Zunächst beginnt jede Tour im Depot (5). Ausgehend vom aktuellen Kunden wird die Menge der zulässigen Nachfolger U' ermittelt (6a.). Der jeweils nächste zu bedienende Kundenauftrag oder das Depot j wird zufallsgesteuert mit einer Wahrscheinlichkeit von p_j aus U' ausgewählt (6b). Die konstruierende Ameise begibt sich zu dem ausgewählten Knoten j , der neue Abreisezeitpunkt τ_j wird ermittelt (6c). Wenn j dem Depot entspricht, wird eine neue Tour begonnen (6d). Ist j ein Kundenauftrag, wird er aus der Menge der zuzuordnenden Kundenaufträgen U entfernt (6e). Sind alle zuzuordnenden Kundenaufträge in der aktuellen Lösung enthalten, wird die Tour und die Konstruktionsphase beendet (6f). Sind noch Knoten zuzuordnen, ist j der neue Ausgangsknoten (6g).

Während der Konstruktionsphase erfolgt kein lokales Pheromonupdate. Im Unterschied zu den Ameisenkolonialgorithmen von Gambardella et al. (1999) und Donati et al. (2003a) aus Abschnitt (4.4.2) und (4.4.3) wird von jeder Ameise eine zulässige Lösung erzeugt.

Auswahlwahrscheinlichkeit p_j : Die Wahrscheinlichkeit p_j , daß ein zulässiger Kundenauftrag oder das Depot j ($j \in U'$) als nächstes gewählt wird, ist immer positiv und kleiner gleich 1.⁷⁸ Die Auswahlwahrscheinlichkeit hängt dabei sowohl

⁷⁷vgl. Stützle und Hoos (1997), S. 310

⁷⁸vgl. Stützle (1998) S. 66

von der lokalen Komponente $\eta_{ij}(\tau)$ als auch von dem Pheromonwert ϕ_{ij} ab, wobei Pheromonwert und Wert der lokalen Komponente multiplikativ verknüpft sind (6b). π ist der Gewichtungsfaktor für die lokale Komponente, ($\pi \geq 0$). Je höher π angesetzt ist, desto stärker wird die Auswahlentscheidung der Ameise durch die lokale Komponente geprägt. Wird π niedrig gewählt, spielen vorteilhafte Lösungsbestandteile aus vergangenen Iterationen eine relativ größere Rolle bei der Auswahl des nächsten zu bedienenden Kunden.

Lokale Komponente $\eta_{ij}(\tau)$: Die lokale Komponente $\eta_{ij}(\tau)$ ist eine Größe, die die individuelle Vorteilhaftigkeit approximieren soll, den Kundenauftrag j direkt im Anschluss an den Kundenauftrag i einzuplanen, wenn der Kunde i zum Zeitpunkt τ verlassen wird. Die dynamischen Fahrzeit-Informationen, Kundenzeitfensterrestriktionen, Distanzen und die Dringlichkeit eines Kundenauftrags werden durch $\eta_{ij}(\tau)$ in der Konstruktionsphase berücksichtigt.

$$\begin{aligned} \eta_{ij}(\tau) := 1/ & \quad (\quad \lambda_1 \cdot d_{ij}(\tau) & (4.1) \\ & + \lambda_2 \cdot t_{ij}(\tau) \\ & + \lambda_3 \cdot \max \{e_j - (\tau + t_{ij}(\tau)); 0\} \\ & + \lambda_4 \cdot (l_j - (\tau + t_{ij}(\tau))) \end{aligned}$$

Die lokale Komponente enthält die gewichtete Summe aus Entfernung $d_{ij}(\tau)$, Fahrzeit $t_{ij}(\tau)$, Wartezeit bei j $\max \{e_j - \tau - t_{ij}(\tau); 0\}$ und den zeitlichen Abstand $(l_j - t_{ij}(\tau) - \tau)$ bis zum Ende des Belieferungsintervalls von j . Dabei sind λ_1 , λ_2 , λ_3 und λ_4 Gewichtungparameter für die lokale Komponente. λ_1 ist der Gewichtungsfaktor für die Entfernung zwischen i und j . λ_2 gewichtet die Fahrzeit, die von i zu j bei Abreise bei i zum Zeitpunkt τ anfällt. λ_3 gewichtet die erwartete Wartezeit vor Beginn der Bedienung von Kunde j . λ_4 gewichtet den zeitlichen Abstand bis zum Ende des Belieferungsintervalls von j . Je größer die Entfernung, die Fahrzeit, die Wartezeit oder der zeitliche Abstand bis zum Ende des Belieferungsintervalls von j ausfallen, desto weniger attraktiv ist es, den Kunden j direkt im Anschluss an i zu beliefern und desto kleiner fällt $\eta_{ij}(\tau)$ aus. Es gilt offensichtlich immer $\eta_{ij}(\tau) > 0$, da der Summand $\lambda_4 \cdot (l_j - (\tau + t_{ij}(\tau)))$ für alle $j \in U'$ positiv ist. Gilt $l_j < (\tau + t_{ij}(\tau))$ handelt es sich bei j um einen Knoten, der nicht in U' , der Menge der zulässigen Nachfolger in der aktuellen Tour, enthalten sein kann. Somit ist sichergestellt, daß die Auswahlwahrscheinlichkeit p_j (positive Pheromonwerte vorausgesetzt) für zulässige Nachfolger $j \in U'$ nicht negativ wird.

4.5.3.2 Verbesserungsphase

Im Anschluss an die Konstruktionsphase ((4) bis (7)) wird versucht, die gefundene Lösung $X^{aktuell}$, sofern sie vielversprechend erscheint, mit Hilfe der CX-Nachbarschaftssuche zu verbessern ((8) bis (12)). Eine Lösung $X^{aktuell}$ wird als vielversprechend angesehen, wenn entweder die von $X^{aktuell}$ benötigte Tourenzahl das $(1 + \gamma)$ -fache der Tourenzahl der besten bekannten Lösung X^{best} nicht übersteigt ($anz(X^{aktuell}) \leq (1 + \gamma) \cdot anz(X^{best})$) und auch die Bediendauer das $(1 + \gamma)$ -fache der Bediendauer der besten bekannten Lösung nicht überschreitet, $sched(X^{aktuell}) \leq (1 + \gamma) \cdot sched(X^{best})$ (9). Ist die neue ermittelte Lösung besser als die beste bekannte Lösung, ist sie in jedem Fall vielversprechend.⁷⁹ Je kleiner γ ist, desto geringer ist die Wahrscheinlichkeit, daß überhaupt eine Lösung $X^{aktuell}$ verbessert wird. Damit sinken zwar die Rechenzeiten, gleichzeitig besteht bei zu kleinem γ die Gefahr, daß der Algorithmus ein schlechtes lokales Optimum nicht mehr verlassen kann.

Um die Suche zeitlich im Rahmen zu halten, wird die CX-Nachbarschaftssuche auf maximal CX_{iter}^{max} Iterationen je Ameise beschränkt (9). Dabei besteht die Möglichkeit die erste gefundene Verbesserung zu akzeptieren (first-fit) oder die beste Lösung der CX-Nachbarschaft zu wählen (best-fit). Um die Rechenzeiten zu beschränken, werden zudem nicht beliebig lange Kunden-Abschnitte im Rahmen der CX-Nachbarschaftssuche vertauscht, sondern maximal $CX_{abschnitt}^{max}$ lange Abschnitte untersucht. In Kapitel 5 werden unterschiedliche Parameter für die Abschnittslänge mit first-fit und best-fit CX untersucht.

Ist die neue Lösung $X^{aktuell}$ besser als die beste bekannte Lösung X^{best} , wird X^{best} durch die neue Lösung $X^{aktuell}$ ersetzt (10). Wenn die Ameise hingegen keine neue beste Lösung finden konnte, wird der Iterationszähler anz_{iter} um eins erhöht (11). anz_{iter} gibt an, wieviele Ameisen-Iterationen durchgeführt wurden, ohne daß eine neue beste Lösung gefunden wurde und wird für das Abbruchkriterium verwendet (14).

4.5.3.3 Pheromonupdate

Ist die Verbesserungsphase abgeschlossen, wird die Pheromonmatrix angepasst ((13) und (14)). Der Parameter ρ legt den Anteil der Pheromone fest, der nach Abschluß einer Ameiseniteration übrigbleibt (im englischen mit „trail-persistence-rate“ bezeichnet, im folgenden Pheromonerhaltungsrate). Durch die Festlegung von ϕ^{min} ist gewährleistet, daß theoretisch alle Fahrten als Lösungsbestandteile in Frage kommen. Sinkt der Pheromonwert für eine bestimmte Fahrt unter ein vorgegebenes Minimum, wird er

⁷⁹Eine bessere als die bislang beste bekannte Lösung erfüllt nicht automatisch beide Ungleichungen: Es ist denkbar, daß eine Lösung $X^{aktuell}$ eine Tour weniger benötigt als X^{best} ($anz(X^{aktuell}) < anz(X^{best})$) und gleichzeitig $sched(X^{aktuell}) > (1 + \gamma) \cdot sched(X^{best})$ gilt. Eine solche Lösung ist auch als vielversprechend anzusehen.

auf den Minimalwert ϕ^{min} festgelegt. Für alle in der aktuell besten bekannten Lösung X^{best} enthaltenen Fahrten werden die zugehörigen Pheromonwerte um das ϕ_{ij}^{extra} -fache erhöht. ϕ_{ij}^{extra} ist dabei für alle Fahrten der Lösung X^{best} umso größer, je besser die aktuell beste bekannte Lösung X^{best} im Vergleich zur Ausgangslösung X^{init} ist. Für alle Fahrten (i, j) die nicht Teil der besten bekannten Lösung X^{best} sind, ist $\phi_{ij}^{extra} = 1$. Damit vorteilhafte Fahrten nicht zu stark bevorzugt werden und dadurch die Suche vorzeitig in einem lokalen Optimum konvergiert, wird der Pheromonwert nach oben durch den Maximalwert ϕ^{max} beschränkt (13).

Zu beachten ist, daß im Unterschied zu der in der Literatur üblichen Vorgehensweise bei Ameisenkolonialgorithmen hier anstelle eines additiven Pheromonupdates ein multiplikatives Pheromonupdate vorgenommen wird. Einige, der experimentellen Analyse vorausgegangene Untersuchungen haben gezeigt, daß ein multiplikatives Pheromonupdate für dynamisch-deterministische Tourenplanungsprobleme einem additiven Pheromonupdate überlegen ist. Ist die beste bekannte Lösung X^{best} besonders vorteilhaft, wird der Nenner relativ groß, es wird daher stärker die Pheromonspur verstärkt.

Der Ameisenkolonialalgorithmus dynACO wird beendet, sobald eine vorgegebene Anzahl anz_{max}^{iter} von Lösungen durch die Ameisen ermittelt wird, ohne daß eine neue, beste Lösung X^{best} gefunden wurde (15). Die dann vorliegende Lösung X^{best} ist der resultierende Tourenplan (16).

4.6 Störungsreaktionsverfahren zu dynamisch-stochastischen Tourensteuerungsproblemen

Zu einer ausführlichen Darstellung des Tourensteuerungsproblems (dTS) vgl. Kapitel 3. Es bestehen unterschiedliche Möglichkeiten mit den im Zeitablauf neu eintreffenden Informationen umzugehen. Im Folgenden werden vier unterschiedliche Handlungsalternativen für dTS dargestellt. Dabei handelt es sich um drei Störungsreaktionsverfahren und die Handlungsalternative „keine Reaktion“. Sie werden in Abschnitt 5.6 experimentell auf ihre Eignung zur Minimierung der Gesamtverspätung, unter Berücksichtigung von stochastischen Fahrzeiten, untersucht.

4.6.1 Handlungsalternative keine Reaktion (ALT)

Die einfachste Handlungsalternative für dTS ist nichts zu verändern. Eine vorläufige Tour T und die dazugehörige Route, die zum dynamisch-deterministischen

Tourenplanungsproblem (dTP) ermittelt wurden und ausschließlich auf den Informationen basieren, die zum Zeitpunkt der Planung verfügbar waren, werden für das dynamisch-stochastische Tourensteuerungsproblem (dTS) ohne Veränderung übernommen. Sowohl die geplante Route als auch die geplante Tour T wird im dynamisch-stochastischen Kontext bewertet. Jede Änderung der geplanten Wege oder der Belieferungsreihenfolge der Kunden ist ausgeschlossen. Die Geschwindigkeitsfunktionen werden durch eintretende Störereignisse verändert, genauso wie die resultierenden Fahrzeiten im Verkehrswegenetzwerk, die Ankunfts- und die Bedienzeitpunkte bei den Kunden.

Die Handlungsalternative, keinerlei Plananpassung vorzunehmen, ist wichtig, um die Wirkung der anderen Störungsreaktionsverfahren beurteilen zu können und um den Unterschied zwischen Planung und Realisierung darzustellen. Diese Handlungsalternative wird im folgenden mit ALT bezeichnet.

4.6.2 Anpassung der Routen im Verkehrswegenetzwerk (ROUTE)

Wird eine einmal geplante Tour T als unveränderlich gegeben angesehen, kann gegebenenfalls eine Anpassung der zugehörige Route als Reaktion auf Störereignisse erfolgen. Eine Änderung der Belieferungsreihenfolge der Kundenaufträge ist beispielsweise dann nicht möglich, wenn die Belieferungsreihenfolge der Kunden durch die Beladungsreihenfolge des Lieferfahrzeugs determiniert ist. Bei einer gegebenen Belieferungsreihenfolge von Kunden werden neue Routen für die Wege im Verkehrswegenetzwerk zwischen den einzelnen Kunden sowie zwischen den Kunden und dem Depot bestimmt. In jedem Reaktionszeitpunkt τ_{reakt} werden neue Routen zu allen Kunden, die noch nicht bedient worden sind, ermittelt. Die neu ermittelten Routen ersetzen die vorläufigen Routen und behalten bis zum nächsten Reaktionszeitpunkt ihre Gültigkeit. Die Fahrzeiten zwischen den einzelnen Kunden und zwischen dem letzten Kunden der Tour und dem Depot werden unter Berücksichtigung aller zum Zeitpunkt τ_{reakt} bekannten Störereignisse ermittelt, wobei die neuen schnellsten Wege mit Hilfe des Verfahrens aus Abschnitt (2.2.2.4) bestimmt werden. Dieses Reaktionsverfahren wird mit ROUTE bezeichnet.

4.6.3 Lokale Verbesserungsverfahren

Neben einer Anpassung der Route bei einer vorgegebenen Tour T besteht gegebenenfalls die Möglichkeit im Rahmen der Tourensteuerung die Belieferungsreihenfolge der Kundenaufträge, d.h. sowohl die Tour als auch

die dazugehörige Route, anzupassen. Gegeben ist eine vorläufige Tour T . Die Belieferungsreihenfolge aller zum Reaktionszeitpunkt τ_m noch nicht belieferten Kundenaufträge kann verändert werden.

Je weiter die Auslieferung fortgeschritten ist, desto weniger Kundenaufträge aus T sind entscheidungsrelevant, da die Belieferungsreihenfolge der bereits besuchten Kunden nicht mehr verändert werden kann. Im Reaktionszeitpunkt τ_{reakt} wird nicht eine komplett neue Plan-Tour auf Basis der neuen Geschwindigkeitsfunktionen ermittelt, sondern die gegebene Tour T durch lokale Suchverfahren angepasst und die dazugehörigen Routen neu ermittelt. Die Route entspricht dabei dem jeweils schnellsten Weg im Verkehrswegenetzwerk zwischen den zu beliefenden Kunden und zwischen Kunden und dem Depot, auf Basis der jeweils aktuell vorliegenden Geschwindigkeitsfunktionen. Die schnellsten Wege werden mit Hilfe des Verfahrens aus Abschnitt (2.2.2.4) bestimmt, unter Berücksichtigung aller bekannten Störereignisse σ (mit $b_\sigma \leq \tau_{reakt}$).

Im folgenden werden beispielhaft zwei einfache lokale Suchverfahren vorgestellt, die für die Tourensteuerung eingesetzt werden können. Die Plan-Belieferungsreihenfolge wird im folgenden in Vektordarstellung beschrieben: $V := (m, v_{[1]}, \dots, v_{[|U|]}, 0)$. $v_{[1]}$ ist der vorläufig nächste, $v_{[2]}$ der vorläufig übernächste, usw. zu beliefende Kunde. Folgender Zusammenhang besteht zwischen den Entscheidungsvariablen x_{ij} des dTS und $V = (m, \dots, v_{[n]}, v_{[n+1]}, \dots, 0)$: Der Kundenauftrag an der $[n + 1]$ -ten Position des Vektors V wird direkt im Anschluss an den Kundenauftrag an der $[n]$ -ten Position beliefert, $x_{[n],[n+1]} = 1$ (für $n = 1(1)|U| - 1$). Das erste Element von V ist immer derjenige Knoten m des Verkehrswegenetzwerks, der als nächstes erreicht wird ($x_{m,[1]} = 1$). Das letzte Element von V muss das Depot sein, da jede Tour im Depot enden muss ($x_{[|U|],0} = 1$). Die Länge des Vektors V entspricht der Mächtigkeit der Menge der noch nicht belieferten Kunden $U + 2$.

4.6.3.1 Paarweiser Tausch (PT)

Die Nachbarschaft PT basiert auf dem paarweisen Vertauschen von zwei direkt nacheinander belieferten Kundenaufträgen (vgl. Abbildung 4.9). Eine Lösung V_2 ist Nachbar der Lösung V_1 , wenn sich V_2 aus V_1 erzeugen lässt, indem die Position zweier direkt aufeinanderfolgender Kundenaufträge vertauscht wird. Aus $V_1 := (\dots, A, B, C, D, \dots)$ wird $V_2 := (\dots, A, C, B, D, \dots)$, wenn die Positionen der Aufträge B und C vertauscht werden. Das erste und das letzte Element der Lösung dürfen nicht vertauscht werden. Weder ist es möglich, daß der nächste erreichbare Knoten $m \in N$ verändert wird, noch darf ein anderer Knoten als das Depot das letzte Element einer Tour werden, da jede Tour im Depot enden muss.

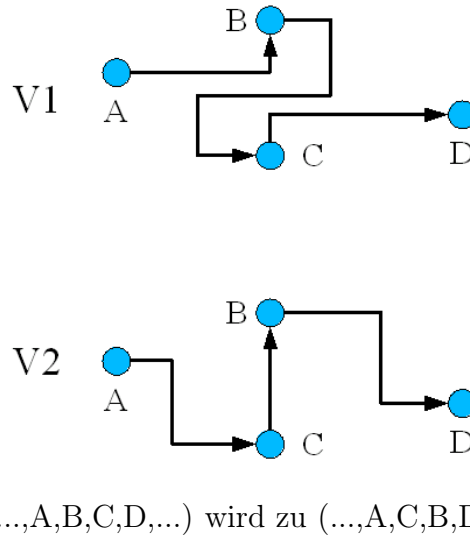


Abbildung 4.9: Störungsreaktionsverfahren PT

Formal wird das PT-Verbesserungsverfahren in Algorithmus 16 dargestellt. Es sei zunächst die Lösung V gegeben, welche die aktuell verbleibende Resttour abbildet. Für alle direkt nacheinander belieferten Kundenauftragspaare i , $(i + 1)$ wird ein Nachbar V_{neu} erzeugt, indem die Kundenaufträge an der i -ten und der $(i + 1)$ -ten Position miteinander vertauscht werden (3). Zu jeder neu ermittelten Resttour V_{neu} wird eine neue Route auf Basis der schnellsten Wege mit Hilfe des Verfahrens aus Abschnitt (2.2.2.4), unter Berücksichtigung aller bekannten Störereignisse σ , bestimmt. Anschließend wird überprüft, ob der Tauschvorgang vorteilhaft ist, indem für die neu entstandene Tour V_{neu} der Zielfunktionswert ermittelt und mit dem der aktuellen Tour V verglichen wird. Ist V_{neu} besser als V , wird der Tausch durchgeführt und der paarweise Tausch neu gestartet (4). Ansonsten wird V_{neu} verworfen und das nächste Paar von Kundenaufträgen untersucht (5). In jeder Iteration wird dabei die erste Verbesserung akzeptiert (first-fit). Das PT-Verbesserungsverfahren endet, wenn durch keinen paarweisen Tausch mehr eine Verbesserung erzielt werden kann.

Algorithmus 16 Störungsreaktionsverfahren PT

1. Für $i := 1(1) |U| - 1$ {
 2. $V_{neu} := V$
 3. vertausche die Kundenaufträge an den Positionen $v_{[i]}$ und $v_{[i+1]}$ von V_{neu}
 4. wenn V_{neu} besser als V , dann $V := V_{neu}$ und gehe zu (1)
 5. } next i
 6. Ergebnis: V ist eine PT-optimale Resttour
-

Die resultierende Resttour V legt die Belieferungsreihenfolge vorläufig fest, bis zu einem späteren Reaktionszeitpunkt neue Informationen eintreffen und berücksichtigt werden (6). Für jeden Kunden der in der Zwischenzeit beliefert wurde ist, liegt die Belieferungsreihenfolge endgültig fest.

4.6.3.2 Tausch der Positionen von zwei Kundenaufträgen (EX)

Die Nachbarschaft EX basiert auf dem paarweisen Vertauschen der Positionen von je zwei Kundenaufträgen, die noch zu beliefern sind (vgl. Abbildung 4.9).

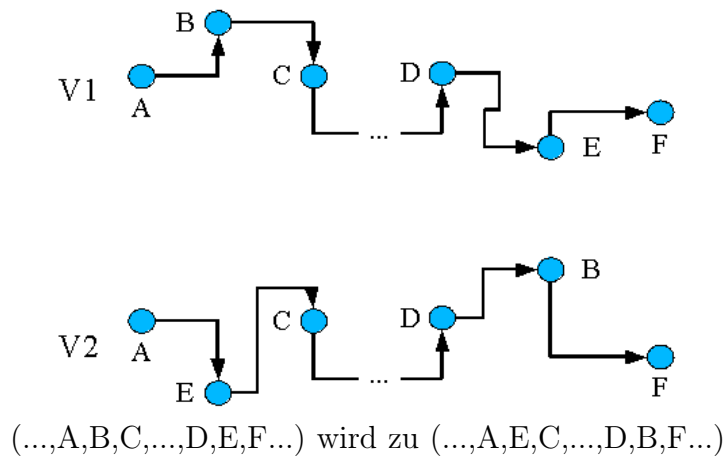


Abbildung 4.10: Störungsreaktionsverfahren EX

Eine Lösung V_2 ist Nachbar der Lösung V_1 , wenn sich V_2 aus V_1 erzeugen lässt, indem die Position von zwei beliebigen Kundenaufträgen vertauscht wird. Aus $V_1 := (... , A, B, C, ..., D, E, F, ...)$ wird $V_2 := (... , A, E, C, ..., D, B, F, ...)$, wenn die Positionen der Aufträge B und E vertauscht werden.

Algorithmus 17 Störungsreaktionsverfahren EX

1. Für $i := 1(1) |U| - 1$ {
 2. Für $j := i + 1(1) |U|$
 3. $V_{neu} := V$
 4. vertausche die Kundenaufträge an den Positionen $v_{[i]}$ und $v_{[j]}$ von V_{neu}
 5. wenn V_{neu} besser als V , dann $V := V_{neu}$ und gehe zu (1)
 6. } next j
 7. } next i
 8. Ergebnis: V ist eine EX-optimale Resttour
-

Formal wird das EX-Verbesserungsverfahren in Algorithmus 17 dargestellt. Gegeben ist die aktuell geplante Resttour V . Für alle Kombinationen von Kundenaufträgen i (1) und j (2) wird ein Nachbar V_{neu} erzeugt, indem die Kundenaufträge an der i -ten und der j -ten Position von V miteinander vertauscht werden (4). Zu jeder neu ermittelten Belieferungsreihenfolge wird eine neue Route bestimmt und überprüft, ob der Tausch vorteilhaft ist (5). Ist V_{neu} besser als V , d.h. ist der Zielfunktionswert von V_{neu} geringer als der Zielfunktionswert von V , wird der Tausch durchgeführt und das Tauschverfahren neu gestartet (5). Ansonsten wird V_{neu} verworfen und das nächste Paar von Kundenaufträgen untersucht ((6) und (7)). In jeder Iteration wird der erste Vektor, der zu einer Verbesserung des Zielfunktionswerts führt, akzeptiert (first-fit). Das EX-Verbesserungsverfahren endet, sobald kein Nachbar mit einem besseren Zielfunktionswert existiert (8).

5 Experimentelle Analyse

5.1 Ziel der experimentellen Analyse zu dynamisch-deterministischen Tourenplanungsproblemen (dTP)

Das Ziel der experimentellen Analyse in dieser Arbeit ist ein Vergleich der Lösungsgüte von ausgewählten Algorithmen zur Lösung von dynamisch-deterministischen Tourenplanungsproblemen (dTP). Die Lösungsgüte des Verfahrens des besten Nachfolgers (im folgenden NN, vgl. Abschnitt 4.2.2) und einer Einfügeheuristik (im folgenden INS, vgl. Abschnitt 4.2.3) für dynamisch-deterministische Tourenplanungsprobleme, jeweils ergänzt durch eine lokale CX-Nachbarschaftssuche (vgl. Abschnitt 4.3.2) wird mit zwei unterschiedlichen Ameisenkolonialgorithmen (MACS, vgl. Abschnitt 4.4.3 und dynACO, vgl. Abschnitt 4.5) verglichen. Darüber hinaus werden zur Lösung von dynamisch-stochastischen Tourensteuerungsproblemen (dTS) drei Störungsreaktionsverfahren (ROUTE, PT und EX) und die Handlungsalternative ALT untersucht (vgl. Abschnitt 4.6).

Obwohl der Mangel an standardisierten Modellen für dynamisch-stochastische Tourenplanungsprobleme bereits von Dror und Powell (1993) kritisiert wird, sind bis heute keine Standardproblemstellungen für dynamisch-deterministische Tourenplanungsprobleme (dTP) in der Literatur vorgeschlagen worden.¹ Für die Untersuchung von dTP werden daher neue Benchmarkprobleme erzeugt. Die Probleminstanzen basieren dabei auf Netzwerken mit den wesentlichen Eigenschaften von realen Verkehrswegenetzwerken. Die konkreten Problemausprägungen stehen im Internet unter <http://www.boomgaarden.org/dvrp> zum Download zur Verfügung.

Zunächst wird in Abschnitt 5.2 dargestellt, wie Verkehrswegenetzwerke generiert werden, auf denen die neuen Probleminstanzen zu dTP aufbauen. Eine ausführliche Beschreibung der neuen Probleminstanzen zu dTP folgt in Abschnitt 5.3.

In Abschnitt 5.4 werden die untersuchten Parameterausprägungen für die Verfahren INS, NN, MACS und dynACO dargestellt. Die Ergebnisse der Versuche zu den

¹vgl. Dror und Powell (1993), S. 11-14

neuen Probleminstanzen zu dTP werden in Abschnitt 5.5 ausführlich dargestellt und interpretiert.

In Abschnitt 5.6 folgt die experimentelle Analyse von dynamisch-stochastischen Tourensteuerungsproblemen. Zunächst werden in 5.6.1 die „Störpläne“ beschrieben, die aus einer gegebenen Lösung zu einem dTP mit Beginn der Auslieferung eine Menge von dTS machen. Ein Störplan ist die Menge aller Störereignisse, die die Fahrzeiten im Verkehrswegenetzwerk beeinflussen. Schließlich folgt die Darstellung und Interpretation der Ergebnisse der darauf aufbauenden Versuche zu dTS in Abschnitt 5.6.2.

5.2 Netzwerkgenerator

5.2.1 Eigenschaften von realen Verkehrswegenetzwerken

Für die Untersuchung von dynamisch-deterministischen Tourenplanungsproblemen (dTP) werden Probleminstanzen auf Netzwerkbasis benötigt, damit im Zeitablauf alternative Wege zwischen einzelnen Kunden und zwischen Kunden und Depot abgebildet werden können. Damit diese Netzwerke die wesentlichen Eigenschaften von realen Verkehrswegenetzwerken aufweisen, wurde ein neuer Netzwerkgenerator entwickelt. Zunächst wurden fünf ausgewählte Regionen aus der Bundesrepublik Deutschland in Netzwerkform erhoben: Passau und Umland, Frankfurt a. M. und Umland, das Autobahnnetz des Ruhrgebiets, Dresden und Umgebung sowie die Region zwischen Kiel und Flensburg. Basierend auf realen Straßenkarten sind jede größere Ortschaft als Knoten, die dazwischen liegenden Autobahnen, Bundesstraßen und Landstraßen als Verbindungen des Netzwerks abgebildet worden. Darüber hinaus wurden zusätzliche Knoten berücksichtigt, die zur Abbildung von Kreuzungen ausserhalb von Ortschaften benötigt wurden.

Eine Untersuchung der Gemeinsamkeiten der Verkehrswegenetzwerke hat folgende charakteristischen Eigenschaften ergeben: Das Verhältnis von Pfeilen zu Knoten ($\frac{\text{Anzahl Pfeile}}{\text{Anzahl Knoten}}$) liegt bei etwa 1,4 bis 1,9 für die untersuchten Karten. Es bestehen in der Regel nur direkte Verbindungen zwischen Knoten, die räumlich nicht weit voneinander entfernt liegen. Zu einem Knoten führen mindestens eine und höchstens sieben Straßen, wobei die meisten Knoten mit drei bis fünf anderen Knoten direkt verbunden sind. Abbildung 5.1 zeigt links einen der untersuchten Ausschnitte der Karten als Netzwerk abgebildet. Werden alle Pfeile zwischen den Knoten entfernt und durch den im folgenden beschriebenen Netzwerkgenerator systematisch neue Pfeile generiert, resultiert das Verkehrswegenetzwerk, welches rechts in Abbildung 5.1 dargestellt ist. Ein optischer Vergleich zeigt, daß der Netzwerkgenerator Strukturen erzeugt, die erstaunliche Ähnlichkeiten mit dem realen Verkehrswegenetzwerk aufweisen. Die

Vorschriften zur Generierung neuer Verkehrswegenetze werden ausführlich im folgenden Abschnitt beschrieben.

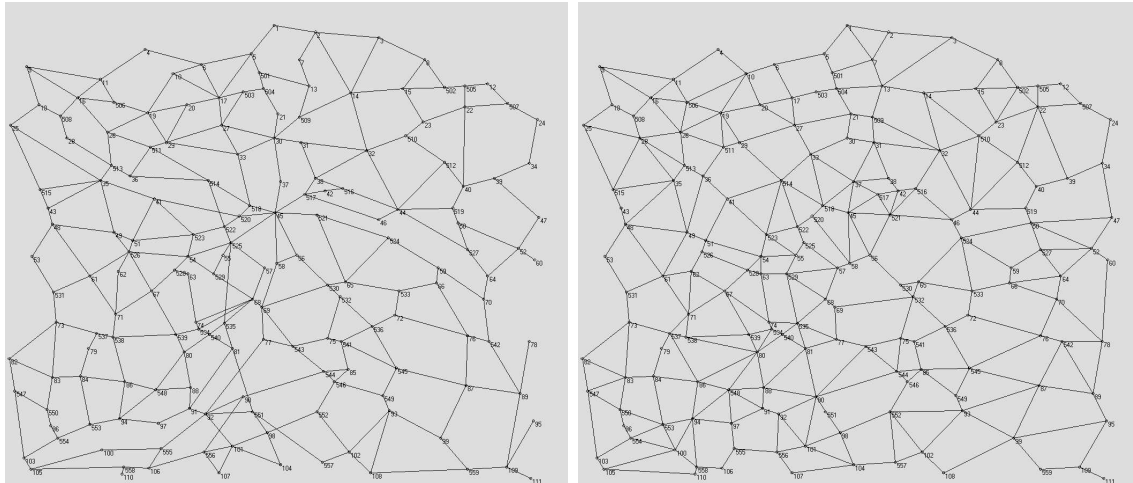


Abbildung 5.1: Links: reales Verkehrswegenetzwerk Passau und Umgebung; Rechts: Ergebnis des Netzwerkgenerators

5.2.2 Netzwerkgenerator: Struktur

Ein neues Verkehrswegenetzwerk wird in zwei Schritten generiert. Zunächst wird eine Netzwerkstruktur erzeugt (vgl. Algorithmus 18), anschließend erfolgt die Bewertung der Pfeile des Netzwerks. W bezeichnet einen Graphen (das in Konstruktion befindliche Verkehrswegenetzwerk), der vorläufig aus der Knotenmenge N und der Kantenmenge K besteht. $|K|$ ist die Anzahl an Kanten, die zum Graphen W gehört. K_{max} ist die Anzahl der Kanten, die der fertige Graph W besitzen soll. d_{mn}^W ist die Länge des kürzesten Wegs zwischen m und n im Graphen W . Mit x_m^{koord} wird die X -Koordinate des Knoten m und mit y_n^{koord} die Y -Koordinate des Knoten n bezeichnet. d_{mn}^{euklid} ist die euklidische Distanz zwischen den Knoten m und n , wobei $d_{mn}^{euklid} := \sqrt{(x_m^{koord} - x_n^{koord})^2 + (y_m^{koord} - y_n^{koord})^2}$ gilt.

Algorithmus 18 Netzwerkgenerator - Struktur

1. Koordinaten für $|N| - 1$ Knoten (Kunden, Kreuzungen) zufällig in der Ebene $[0; 500]^2$ bestimmen.
 2. Depot in der Mitte der Ebene (Position (250; 250)) hinzufügen.
 3. Auf Basis der euklidischen Entfernungen zwischen den Knoten $m, n \in N$ wird ein minimal spannender Baum gebildet (Verfahren von Kruskal (1956)).
 4. Wiederhole {
 5. Bestimme die jeweils kürzesten Wege zwischen allen Knoten $m, n \in N$ im Graphen W (Verfahren von Floyd (1962)).
 6. Füge Kante $[m, n]$ in W ein, für die $\frac{d_{mn}^W}{d_{mn}^{euklid}}$ maximal ist und die zu keiner Überschneidung führt.
 7. Wenn keine Kante $[m, n]$ ohne Überschneidung eingefügt werden kann, gehe zu (9).
 8. } bis $|K| \geq K_{max}$
 9. Ergebnis: Graph W .
-

In einem ersten Schritt wird eine Menge von $|N| - 1$ Knoten zufällig in der Ebene $[0; 500]^2$ verteilt generiert (1). Die X - und Y -Koordinaten eines jeden Knoten werden dazu unabhängig voneinander, gleichverteilt aus dem Intervall $[0; 500]$ gezogen. Diese Knoten bilden die geographische Lage der Kreuzungen und der zu beliefernden Kunden ab. Ein Depotknoten wird in die Mitte der Ebene (Koordinaten (250; 250)) eingefügt (2). Die Distanzen zwischen allen Knoten $m, n \in N$ werden mit euklidischen Distanzen d_{mn}^{euklid} in der Ebene bewertet. Zwischen allen Knoten (Depot, Kunden, Kreuzungen) wird zunächst mit dem Verfahren von Kruskal ein minimal spannender Baum bestimmt (3).² Nach diesem Schritt liegt ein erster, verbundener Graph mit $|N| - 1$ Kanten vor.

Der Graph W wird sukzessive durch weitere Kanten ergänzt ((4) bis (8)). Es werden die kürzesten Wege zwischen allen Knoten des Graphen W mit Hilfe des Verfahrens von Floyd (1962) ermittelt (5).³ Für alle nicht direkt verbundenen Knotenpaare m, n wird die „relative Streckenersparnis“ $\frac{d_{mn}^W}{d_{mn}^{euklid}}$ ermittelt, die dem Verhältnis von kürzestem Weg im Graphen zwischen m und n zu der euklidischen Entfernung zwischen den Knoten m und n entspricht. Je größer die relative Wegersparnis ist, desto vorteilhafter ist es, die Knoten m und n direkt zu verbinden. Dem Graphen wird diejenige Kante $[m, n]$ hinzugefügt, für die die relative Streckenersparnis maximal ist und die beim Einfügen in den Graphen zu keiner Überschneidung mit vorhandenen Kanten führt (6).

²vgl. Kruskal (1956), S. 48-50

³vgl. Floyd (1962), S. 345

Durch die Koordinaten der beiden Knoten die eine Kante bestimmen, läßt sich eine Gerade legen. Um zu überprüfen, ob sich zwei Kanten überschneiden, wird der Schnittpunkt s der beiden durch die Kanten verlaufenden Geraden ermittelt (vgl. Abbildung 5.2). Damit sich die Kanten $[A, B]$ und $[C, D]$ schneiden, muß der

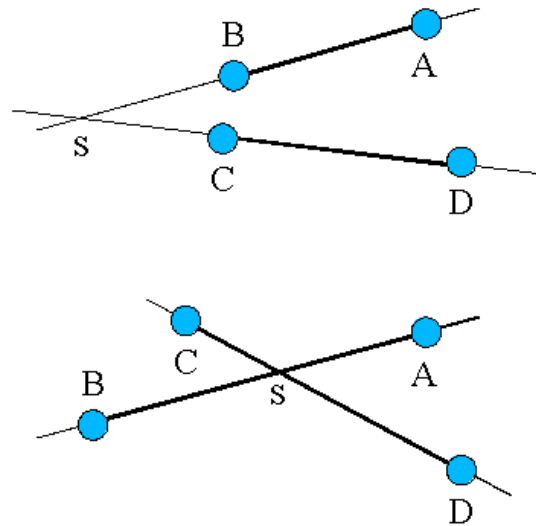


Abbildung 5.2: Ermittlung des Schnittpunkts zweier Kanten

Schnittpunkt s , beschrieben durch seine Koordinaten x_s^{koord} und y_s^{koord} auf beiden Kanten liegen. Wenn folgende Bedingungen erfüllt sind, liegt eine Überschneidung vor: x_s^{koord} muß zwischen x_A^{koord} und x_B^{koord} sowie zwischen x_C^{koord} und x_D^{koord} liegen, gleichzeitig muß y_s^{koord} zwischen y_A^{koord} und y_B^{koord} sowie zwischen y_C^{koord} und y_D^{koord} liegen. Ist eine der Bedingungen nicht erfüllt, überschneiden sich die Kanten $[A, B]$ und $[C, D]$ nicht. Überschneidungen sind deshalb auszuschliessen, weil es nicht plausibel erscheint zwei Straßen zu kreuzen, ohne daß eine Möglichkeit besteht, von einer Straße auf die andere zu wechseln. Damit abgebildet werden kann, daß von einer Straße auf eine andere gewechselt werden kann, ist aber ein Knoten erforderlich. In realen Verkehrswegenetzwerken treten vereinzelt Überschneidungen auf, wenn beispielsweise eine Autobahn von einer Straße unterquert wird, ohne daß eine Auf- und Abfahrt an der Stelle vorgesehen ist. Dies wird hier nicht berücksichtigt.

Die Ergänzung des Graphen W wird solange fortgesetzt, bis keine Kante mehr existiert, die ohne Überschneidung eingefügt werden kann (7) oder bis eine vorher festgelegte Anzahl an Kanten K_{max} erreicht wird (8).

Im Rahmen der experimentellen Analyse werden Probleminstanzen mit 50 und 65 Kundenaufträgen betrachtet. Es werden vier unterschiedlich dichte Verkehrswegenetzwerke generiert: 50a, 50b, 65a und 65b. Die Netzwerke 50a und 50b basieren auf einem Depot, 50 Kundenknoten und 150 weiteren Knoten. Das Netzwerk 50a erhält insgesamt 298 Kanten, das Netzwerk 50b 398 Kanten. Alle Kanten aus 50a

sind auch in 50b enthalten. Die Netzwerke 65a und 65b besitzen neben dem Depot 65 Kundenknoten und 140 weitere Knoten. 65a erhält insgesamt 307 Kanten und 65b erhält 409 Kanten. 65b enthält wiederum alle Kanten, die in 65a enthalten sind.

In Abbildung (5.3) sind die Netzwerke 50a und 50b, in Abbildung (5.4) die Netzwerke 65a und 65b dargestellt. Die Kundenstandorte und das Depot werden durch dickere Punkte hervorgehoben dargestellt.

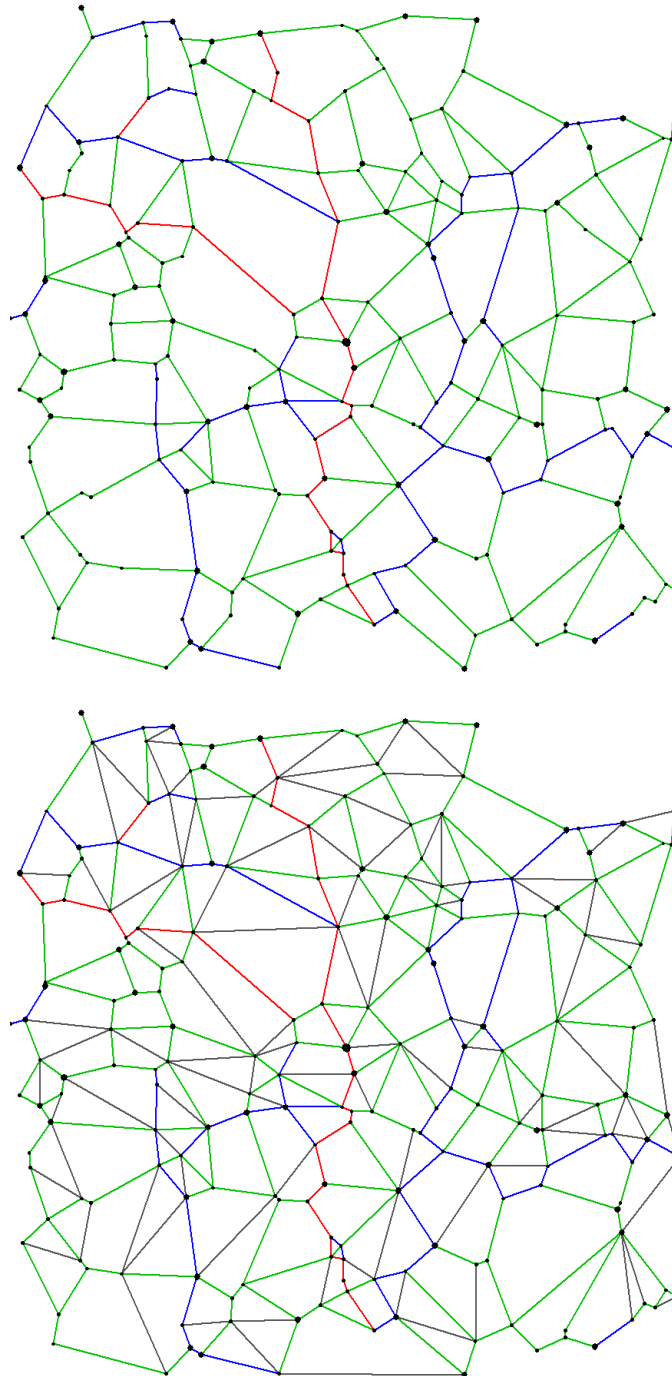


Abbildung 5.3: Verkehrswegenetzwerke: oben 50a, unten 50b

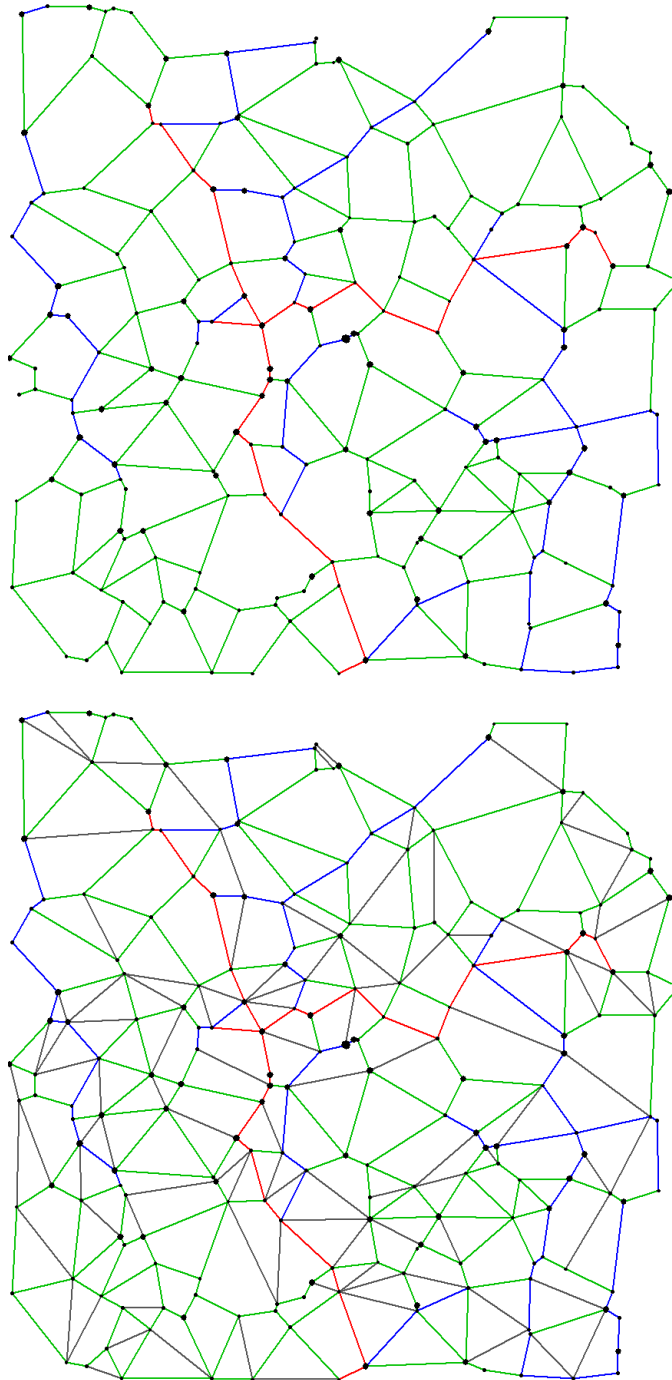


Abbildung 5.4: Verkehrswegenetzwerke: oben 65a, unten 65b

5.2.3 Netzwerkgenerator: Bewertung der Pfeile des Verkehrswegenetzwerks

5.2.3.1 Distanzbewertung

Sobald die Struktur des Graphen W festliegt, wird aus dem Graphen ein Netzwerk, indem jede Kante $[m, n]$ durch zwei entgegengesetzte Pfeile (m, n) und (n, m) ersetzt und anschließend bewertet wird.⁴ Der Übergang von einem ungerichteten zu einem gerichteten Graphen ist erforderlich um asymmetrische Beziehungen im Verkehrswegenetzwerk (beispielsweise unterschiedliche Geschwindigkeiten in entgegengesetzten Richtungen einer Verbindung) abbilden zu können. Jeder Pfeil erhält sowohl Entfernungen als auch Geschwindigkeitsfunktionen als Bewertung. Das Netzwerk W erhält für alle Pfeile die euklidische Distanz zwischen den Knoten m und n als Distanzbewertung d_{mn} und d_{nm} zugewiesen. Die Entfernungen sind im Zeitablauf konstant. Die Geschwindigkeiten sind dynamischer Natur, sie sind im Zeitablauf variabel.

5.2.3.2 Geschwindigkeitsfunktionen für dynamisch-deterministische Tourenplanungsprobleme (dTP)

Die abschnittsweise definierten Geschwindigkeitsfunktionen eines Pfeils (m, n) , die zur Ermittlung der variablen Fahrzeiten dienen, werden als Produkt aus Kategorie- und Typ-Multiplikator ($kategorie_{mn} \cdot typ_{mn}$) errechnet.

Zuweisung $kategorie_{mn}$: Durch die Zuordnung einer Kategorie $kategorie_{mn}$ zu einem Pfeil werden die tageszeitabhängigen Schwankungen der Geschwindigkeiten abgebildet. Jede Kategorie besitzt unterschiedliche, tageszeitabhängige Multiplikatoren. Die Kategorien unterscheiden sich dahingehend, wann sich die Geschwindigkeit wie stark ändert und wie stark die daraus resultierenden Fahrzeiten im Zeitablauf schwanken. Da nicht alle Probleminstanzen die gleichen Öffnungszeiten des Depots $[e_0; l_0]$ aufweisen werden, sind die Zeitangaben für die Gültigkeitsdauer eines Kategorie-Multiplikators im folgenden nicht in Zeiteinheiten sondern in Prozent vom Planungszeitraum (% v. PZR) angegeben.

In Tabelle (5.1) und (5.2) sind zwei unterschiedliche Ausprägungen von Kategorie-Multiplikatoren, die im Rahmen der experimentellen Analyse verwendet werden, angegeben. Die Tabelle (5.1) beschreibt dabei die weniger dynamischen Bewertungen des Netzwerks. Zu Beginn und zum Ende des Planungszeitraums sind die Geschwindigkeiten im Verkehrswegenetzwerk am höchsten. Die Kategorie 0

⁴Es ergeben sich: 50a: 596 Pfeile, 50b: 796 Pfeile, 65a: 614 Pfeile, 65b: 818 Pfeile

% v. PZR	Kat. 0	Kat.1	Kat. 2	Kat. 3
0-10	2,0	2,0	2,0	2,0
10-25	1,0	1,0	1,6	1,6
25-75	1,6	1,6	1,6	1,6
75-90	1,0	1,6	1,0	1,6
90-100	2,0	2,0	2,0	2,0

Tabelle 5.1: Kategorie-Multiplikatoren: geringere Dynamik

% v. PZR	Kat. 0	Kat. 1	Kat. 2	Kat. 3
0-10	3,0	3,0	3,0	3,0
10-30	1,4	2,0	2,0	1,4
30-45	2,0	2,5	2,0	2,0
45-60	1,4	2,0	1,4	2,0
60-80	1,0	1,5	1,0	1,7
80-85	2,0	2,5	2,0	2,5
85-100	3,0	3,0	3,0	3,0

Tabelle 5.2: Kategorie-Multiplikatoren: stärkere Dynamik

weist sowohl zu frühen als auch zu späten Zeiten Störungen auf. Diese Kategorie wird beispielsweise innerstädtischen Pfeilen zugewiesen, die sowohl morgens als auch am späten Nachmittag durch Feierabendverkehr beeinträchtigt werden. Pfeile der Kategorie 1 sind vor allem im ersten Teil des Planungszeitraums durch reduzierte Geschwindigkeiten gekennzeichnet (z.B. Verbindungen, die zu Ballungsräumen hinführen), während Pfeile der Kategorie 2 vor allem im letzten Teil des Planungszeitraums geringere Geschwindigkeiten besitzen (nur durch Feierabendverkehr betroffene Verbindungen). Pfeile der Kategorie 3 sind deutlich weniger stark durch Schwankungen der Fahrzeiten betroffen.

Die Tabelle (5.2) stellt stärker im Zeitablauf schwankende Multiplikatoren dar, die zudem in sieben statt fünf unterschiedliche Intervalle unterteilt werden. Die Kategorie 0 und 1 unterscheiden sich dahingehend, wie stark die Geschwindigkeiten tageszeitabhängigen Schwankungen unterliegen. Kategorie 0 weist stärkere Schwankungen auf als Kategorie 1. Pfeile der Kategorie 2 entsprechen im Wesentlichen denen der Kategorie 0, abgesehen davon, daß die Geschwindigkeit im Intervall zwischen 10-30% des Planungszeitraums weniger stark sinkt (Faktor 2,0 statt Faktor 1,4). Pfeile der Kategorie 3 unterscheiden sich von denen der Kategorie 0 im Intervall von 60-85% des Planungszeitraums, die Geschwindigkeit wird weniger stark reduziert (Faktor 1,7 statt 1,0 bzw. 2,5 statt 2,0).

Die Zuordnung einer Kategorie zu einem Pfeil erfolgt zufällig, wobei einem Pfeil mit einer Wahrscheinlichkeit von 25% die Kategorie 0, mit einer Wahrscheinlichkeit von 15% die Kategorie 1, mit einer Wahrscheinlichkeit von 30% die Kategorie 2 und mit

einer Wahrscheinlichkeit von 30% die Kategorie 3 zugeordnet wird.

Zuweisung typ_{mn} : Der Pfeil-Typ bildet vereinfachend die unterschiedlichen Durchschnittsgeschwindigkeiten auf Autobahnen, Schnellstraßen und Nebenstraßen (schnell/mittel/langsam) ab. Jedem Pfeil-Typ ist ein konstanter, tageszeitunabhängiger Multiplikator typ_{mn} für die Durchschnittsgeschwindigkeit auf dem Pfeil zugeordnet. Für alle Pfeile des Verkehrswegenetzwerks (m, n) $m, n \in N$ gilt $typ_{mn} = typ_{nm}$ (Symmetrie), da davon auszugehen ist, daß eine Autobahn in einer Richtung nicht einen Feldweg als Gegenrichtung besitzt. Typ 0 entspricht der schnellsten Verbindung (z.B. Autobahn), Typ 3 der langsamsten Verbindung (z.B. Nebenstraße).

Typ	Multipl.
0	2,5
1	1,9
2	1,0
3	0,7

Tabelle 5.3: Typ-Multiplikatoren

In Tabelle 5.3 sind die Ausprägungen der vier Typ-Multiplikatoren, die im Rahmen der experimentellen Analyse verwendet werden, aufgeführt.

Damit die Pfeile der Kategorie 0 und gegebenenfalls der Kategorie 1 (Autobahnen, Hauptstraßen) nicht nur zwischen einzelnen Knoten des Verkehrswegenetzwerks zufällig verteilt auftreten, sondern zusammenhängende Durchgangsstraßen im Netzwerk bilden, werden die Typ-Multiplikatoren typ_{mn} den Pfeilen des Netzwerks nach folgender Vorschrift zugewiesen (vgl. Algorithmus 19):

Mit der Variablen $anz_{zugeordnet}$ wird die Anzahl der Pfeile gezählt, denen bereits ein Typ zugewiesen wurde. anz_0 ist eine Vorgabe, die festlegt, wieviele Pfeile den Typen 0 zugewiesen bekommen, mit anz_1 wird die Anzahl der Pfeile bezeichnet, die den Typ 1 zugewiesen bekommen sollen.

Die Pfeile (m, n) des Netzwerks besitzen noch keine zugeordneten Typ-Multiplikatoren. Als erstes ist der Typ 0 (der schnellste Typ) einem Teil der Pfeile des Netzwerks zuzuordnen (1). Ein beliebiger Knoten m des Netzwerks wird zufällig als Ausgangsknoten gewählt (3). Dabei ist die Auswahlwahrscheinlichkeit für jeden Knoten gleich hoch. Dazu wird zufällig eine von vier Richtungen r (Nord, Süd, Ost, West) gewählt (4). Ausgehend vom aktuellen Knoten m werden alle mit m benachbarten Knoten n (es existiert ein Pfeil (m, n)) ermittelt und in der Menge R gespeichert (6). Alle Knoten, die entgegen der aktuell vorgegebenen Richtung r liegen, werden aus R wieder entfernt ((6a) bis (6d)). Ist die aktuelle Richtung r beispielsweise „Nord“, werden

Algorithmus 19 Netzwerkgenerator - Zuweisung von Typ-Multiplikatoren

1. setze $typ := 0$, $typ_{mn} := -1 \forall (m, n) \in W$, $anz_{zugeordnet} := 0$
 2. wiederhole {
 3. wähle beliebigen Knoten $m \in N$
 4. wähle zufällig Richtung $r \in \{Nord, Sued, Ost, West\}$
 5. wiederhole {
 6. ermittle Menge R aller Nachbarknoten von m
 - a) wenn $r = \{Nord\}$ dann entferne alle Knoten n aus R für die gilt:
 $(y_m^{koord} - y_n^{koord}) > 0$
 - b) wenn $r = \{Sued\}$ dann entferne alle Knoten n aus R für die gilt:
 $(y_m^{koord} - y_n^{koord}) < 0$
 - c) wenn $r = \{Ost\}$ dann entferne alle Knoten n aus R für die gilt:
 $(x_m^{koord} - x_n^{koord}) > 0$
 - d) wenn $r = \{West\}$ dann entferne alle Knoten n aus R für die gilt:
 $(x_m^{koord} - x_n^{koord}) < 0$
 7. entferne aus R alle Knoten n für die gilt $typ_{mn} \neq -1$
 8. wenn $R = \emptyset$ dann gehe zu (3)
 9. wenn $r \in \{Nord, Sued\}$ dann setze $r_{mn} := |x_m^{koord} - x_n^{koord}| \forall n \in R$
wenn $r \in \{Ost, West\}$ dann setze $r_{mn} := |y_m^{koord} - y_n^{koord}| \forall n \in R$
 10. wähle Knoten n mit minimalem Abstand r_{mn} ,
 11. setze $typ_{mn} := typ$; $typ_{nm} := typ$; $m := n$, $anz_{zugeordnet} := anz_{zugeordnet} + 1$
 12. wenn $anz_{zugeordnet} > anz_0$ setze $typ := 1$
 13. } bis $anz_{zugeordnet} > anz_0 + anz_1$ (Ende Wiederhole)
 14. $typ_{mn} := 2$ für alle $(m, n) \in \{(m, n) | typ_{mn} = \emptyset\}$
 15. wenn Problem Instanz auf Basis der Netzwerke 50b oder 65b,
dann setze $typ_{mn} := 3$ für alle Pfeile, die nicht in den Netzwerken 50a oder 65a
enthalten sind
 16. Ergebnis: Verkehrswegenetzwerk W
-

alle Knoten n aus R entfernt, für die gilt $(y_m^{koord} - y_n^{koord}) > 0$, denn diese Pfeile zeigen in Richtung „Süd“ (6a). Anschließend werden alle Knoten n aus R entfernt, deren zugehörigem Pfeil (m, n) bereits ein Typ zugewiesen wurde, für die also $typ_{mn} \neq -1$ gilt (7). Ist die Menge R eine leere Menge, wird ein neuer Ausgangsknoten gewählt (8), sonst ist für alle Knoten n aus R die Abweichung von der aktuellen Richtung (r_{mn}) zu ermitteln (9). Der Knoten n ist auszuwählen, zu dem die Abweichung r_{mn} minimal ist (10). Ist die aktuelle Richtung beispielsweise „Nord“ wird also der Kunde $n \in R$ ausgewählt für den $|x_m^{koord} - x_n^{koord}|$ minimal ist. Der aktuell zuzuordnende Typ wird dem Pfeil (m, n) und (n, m) zugeordnet, der Knoten n ist der neue Ausgangsknoten. Der Zähler $anz_{zugeordnet}$ ist um eins zu erhöhen (11). Ist bereits einer vorgegebenen Anzahl anz_0 an Pfeilen der Typ 0 zugewiesen worden, wird mit der Zuweisung der Kategorie 1 fortgefahren (12). Ist auch die vorgegebene Anzahl an Pfeilen der Kategorie 1 zugewiesen worden (13), wird den verbleibenden Pfeilen die Kategorie 2 zugewiesen (14).

Basieren die neuen Problem instanzen auf Netzwerk 50a oder 65a, ist das Netzwerk an dieser Stelle fertig bewertet. Der Typ 3 wird nicht zugewiesen. Für Problem instanzen auf Basis der Netzwerke 50b und 65b erfolgt noch ein weiterer Zuweisungsschritt: Alle Pfeile (m, n) aus den Netzwerken 50b (65b), die nicht in dem Verkehrswegenetzwerk 50a (65a) enthalten sind, erhalten den Typ 3 zugeordnet (15). Dieser Typ bildet zusätzliche Verbindungen ab, die durch besonders geringe Durchschnittsgeschwindigkeiten gekennzeichnet sind. Die Verkehrswegenetzwerke 50a und 65a enthalten etwa zu 15% Pfeile vom Typ 0 (schnell), 40% von Typ 1 (mittel) und 45% von Typ 3 (langsam). Für die Problem instanzen auf Basis von 50b und 65b ergeben sich folgende Anteile : 11% Typ 0 (schnell), 30% Typ 1 (mittel), 34% Typ 2 (langsam) und 25% Typ 3 (sehr langsam). Diese Verteilungen von Typen entsprechen etwa den durchschnittlichen Anteilen von Autobahnen, Schnellstraßen, größeren und kleineren Nebenstraßen in den untersuchten Verkehrswegenetzwerken.

5.2.3.3 Geschwindigkeitsfunktionen für dynamisch-stochastische Tourensteuerungsprobleme (dTS)

Die abschnittsweise definierten Geschwindigkeitsfunktionen, die zur Ermittlung der dynamisch-stochastischen Fahrzeiten für dTS dienen, werden als Produkt aus Kategorie-Multiplikator, Typ-Multiplikator und Störfaktor ($kat_{mn} \cdot typ_{mn} \cdot \Delta_\sigma$) errechnet.

Zu Beginn der Auslieferung, wenn also alle Lieferfahrzeuge das Depot verlassen, gelten zunächst für alle Pfeile (m, n) des Verkehrswegenetzwerks die Geschwindigkeitsfunktionen des zugrundeliegenden dTP. Sobald im

dynamisch-stochastischen Kontext ein Störereignis σ bekannt wird, werden die Geschwindigkeitfunktionen g_{mn} angepasst.

In der folgenden Tabelle ist beispielhaft ein Teil der Geschwindigkeitsfunktion des Pfeils $(m, n) = (1, 5)$ abgebildet. Es wird angenommen, dass der Planungszeitraum $[e_0; l_0] = [0; 100]$ beträgt. Ab dem Zeitpunkt $\tau = 25$ wird bereits im dTP eine Verringerung der Geschwindigkeit auf dem Pfeil $(1, 5)$ erwartet und bei der Ermittlung des Tourenplans in die Planung einbezogen.

<i>int</i>	τ_e^{int}	τ_l^{int}	<i>m</i>	<i>n</i>	<i>typ_{mn}</i>	<i>kategorie_{mn}</i>	Δ_σ	g_{mn}^{int}
...								
5	10	25	1	5	1,9	1,0	1	1,90
6	25	75	1	5	1,9	1,6	1	3,04
...								

Tabelle 5.4: Geschwindigkeitsfunktion g_{mn} vor Störereignis

Das Störereignis σ besitzt die folgenden Eigenschaften: $e_\sigma = 20$, $l_\sigma = 40$, $\Delta_\sigma = 0,3$, $m_\sigma = 1$ und $n_\sigma = 5$. Wird das Störereignis σ bekannt gegeben, ist ab dem Zeitpunkt b_σ die Geschwindigkeitsfunktion folgendermassen anzupassen:

<i>int</i>	τ_e^{int}	τ_l^{int}	<i>m</i>	<i>n</i>	<i>typ_{mn}</i>	<i>kategorie_{mn}</i>	Δ_σ	g_{mn}^{int}
...								
5	10	20	1	5	1,9	1,0	1	1,90
6	20	25	1	5	1,9	1,0	0,3	0,57
7	25	40	1	5	1,9	1,6	0,3	0,91
8	40	75	1	5	1,9	1,6	1	3,04
...								

Tabelle 5.5: Geschwindigkeitsfunktion g_{mn} nach Störereignis

Die Geschwindigkeit von 1 zu 5 beträgt zwischen dem Zeitpunkt 20 und dem Zeitpunkt 25 anstelle von $1,9 \cdot 1,0 \cdot 1 = 1,90$ ohne Störereignis nur noch $1,9 \cdot 1,0 \cdot 0,3 = 0,57$. Zwischen dem Zeitpunkt 25 und 40 beträgt die Geschwindigkeit statt 3,04 nur noch 0,91. Die Geschwindigkeitsfunktion besitzt nach dem Bekanntwerden des Störereignisses zwei Intervalle mehr als vorher.

Ein Störereignis σ führt immer genau dann zu einer Teilung der Intervalle der Geschwindigkeitsfunktion, wenn Start- und/oder Endzeitpunkte des Störereignisses e_σ und l_σ nicht mit vorhandenen Intervallgrenzen der Geschwindigkeitsfunktion übereinstimmen. Für alle Intervalle innerhalb des Zeitraums $[e_\sigma; l_\sigma]$ wird die Geschwindigkeitsfunktion mit dem Multiplikator Δ_σ multipliziert. Auf eine algorithmische Darstellung wird an dieser Stelle verzichtet. In Abschnitt 5.6.1 wird auf die Erzeugung von Störereignissen für die experimentelle Analyse im Detail eingegangen.

5.3 Neue Problem instanzen zu dynamisch-deterministischen Tourenplanungsproblemen

Zur Untersuchung von Algorithmen zu dTP werden insgesamt 18 neue Problem instanzen generiert. Die Problem instanzen A1, B1, C1 und D1 basieren auf dem Netzwerk 50a, die Problem instanzen E1 und F1 auf 65a. Die Pfeile des Netzwerks werden mit den weniger dynamischen Kategorie-Multiplikatoren (vgl. Tabelle 5.1) und drei Typ-Bewertungen (Typ 0, Typ 1 und Typ 2 aus Tabelle 5.3) bewertet.

Die Problem instanzen A2, B2, C2 und D2 basieren auf dem Netzwerk 50b, E2 und F2 auf dem Netzwerk 65b. Wiederum werden alle Pfeile mit weniger dynamischen Kategorie-Multiplikatoren (vgl. Tabelle 5.1) bewertet. Alle vier Typ-Multiplikatoren werden verwendet, wobei der Typ 3 nur den Pfeilen des Netzwerks 50b bzw. 65b zugeordnet wird, die in Netzwerk 50a bzw. 65a nicht enthalten sind. Damit sind Lösungen zur Problem instanz A1 auch immer zulässige Lösungen für die Problem instanz A2, das gleiche gilt für B1 und B2, usw. Umgekehrt gilt allerdings nicht, daß zulässige Lösungen zu A2 auch für A1 zulässig sind, denn die zum Tourenplan gehörenden Routen zu A2 können in A1 gegebenenfalls unzulässig oder nicht existent sein.

Die Problem instanzen A3, B3, C3 und D3 basieren ebenfalls auf dem Netzwerk 50b, E3 und F3 auf dem Netzwerk 65b. Die Pfeile werden im Unterschied zu A2 bis F2 allerdings mit den dynamischeren Kategorie-Multiplikatoren (vgl. Tabelle 5.2) bewertet.

Problem instanz	A	B	C	D	E	F
Netzwerk	50a/50b	50a/50b	50a/50b	50a/50b	65a/65b	65a/65b
$ C $	50	50	50	50	65	65
Planungszeit- raum [ZE]	2.000	5.000	2.000	2.000	2.000	5.000
Anteil Kunden mit Zeitfenster- restriktionen	100%	100%	100%	66%	100%	100%
Größe Zeitfenster- restriktionen	eng	eng	weit	eng	eng	eng
Fahrzeug- kapazität Q	200	800	200	200	200	800

Tabelle 5.6: Charakteristika der neuen Problem instanzen zu dTP

Die neuen Problem instanzen zu dTP sind anhand folgender Merkmalsausprägungen charakterisierbar (vgl. auch eine Übersicht in Tabelle 5.6).

- Anzahl Kundenaufträge: Die Probleminstanzen A, B, C und D umfassen 50 Kundenaufträge, die Probleminstanzen E und F 65 Kundenaufträge. Die Kundenstandorte C_0 sind eine zufällig gezogene Teilmenge der Knotenmenge $N \setminus \{0\}$ des Verkehrswegenetzwerks ohne Depot. Alle Probleminstanzen A, B, C und D besitzen dieselben Kundenstandorte, genauso entsprechen sich die Standorte bei den Probleminstanzen E und F.
- Kapazitätsbedarf: Jedem Kundenauftrag $i \in C$ wird ein Kapazitätsbedarf q_i zufällig gleichverteilt aus dem Intervall $[10; 30]$ zugeordnet.
- Fahrzeugkapazität: Es wird unterstellt, daß eine Flotte von homogenen Fahrzeugen mit der gleichen Kapazität die Touren durchzuführen haben. Geringe Fahrzeugkapazitäten Q werden als etwa das 10-fache eines durchschnittlichen Kapazitätsbedarfs q_i eines Kunden i angesetzt (also: $Q := (10 + \frac{1}{2} \cdot (30 - 10)) \cdot 10 = 200$), hohe Fahrzeugkapazitäten als das 40-fache ($Q := 800$). Eine Kapazität von $Q = 200$ erlaubt es im Mittel 10 Kunden, maximal 20 Kunden pro Tour zu beliefern⁵. Die Kapazität des Lieferfahrzeugs ist für die Probleminstanzen A, C, D und E relativ gering. Für die Probleminstanzen B und F wird eine höhere Fahrzeugkapazität ($Q = 800$) angenommen, damit können sich längere Touren (mit mehr Kundenaufträgen) ergeben. Bei den Probleminstanzen A, C, D und E sind gute Lösungen mit etwa 5-7 Touren zu erwarten, bei den Probleminstanzen B und F sollten zwei bis drei Touren für die Belieferung aller Kunden ausreichen.
- Planungshorizont: Die Tourenplanung erfolgt auf Tagesbasis. Wenn auf einer Tour mehrere Kundenaufträge ausgeführt werden sollen, müssen innerhalb des Planungszeitraums Fahrzeiten, Servicezeiten und Wartezeiten anfallen können. Für alle Probleminstanzen zu dTP wird der Zeitpunkt, zu dem das Depot frühestens verlassen werden kann, mit $e_0 := 0$ zugewiesen. Der gesamte Planungszeitraum wird für die Probleminstanzen A, C, D und E in 2000 Zeiteinheiten unterteilt ($l_0 := 2000$). Damit bei B und F die relativ höhere Fahrzeugkapazität Q ausgeschöpft werden kann, wird der Planungshorizont für B und F in 5000 Zeiteinheiten unterteilt ($l_0 := 5000$).
- Zeitfensterrestriktionen: Bei allen Probleminstanzen sind Zeitfensterrestriktionen für den Beginn der Belieferung eines Kundenauftrags zu berücksichtigen. Die Probleminstanzen A, B, D, E und F weisen enge Zeitfenster auf, die Länge des Belieferungsintervalls wird zufällig gleichverteilt aus einem Intervall zwischen 5% und 20% des Planungszeitraums gezogen. Die Probleminstanzen C haben weite Zeitfenster, die Länge des Belieferungsintervalls wird zufällig gleichverteilt aus dem Intervall zwischen 20% und 50% des Planungszeitraums gezogen.

⁵ $\frac{200}{10} = 20$ bei minimaler Nachfrage $q_i = 10$ für alle Kunden i der Tour

Die Probleminstanzen D sind dadurch gekennzeichnet, daß nur 66% aller Kundenaufträge ein Zeitfenster aufweisen, die übrigen Kundenaufträge können zu einem beliebigen Zeitpunkt aus dem gesamten Planungszeitraum beliefert werden. Kundenaufträge, für die kein eigenes Zeitfenster bestimmt wird, erhalten das Zeitfenster des Depots.

Zur Festlegung eines Zeitfensters wird zunächst die Mitte des Zeitfensters τ_1 zufällig gleichverteilt aus dem Planungszeitraum $[e_0; l_0]$ gezogen. Anschließend wird die Dauer des Zeitfensters t_1 festgelegt (wiederum zufällig gleichverteilt zwischen minimaler und maximaler Dauer). Der Zeitpunkt e_i zu dem frühestens mit der Bedienung des Kunden i begonnen werden kann, ergibt sich als $e_i := \max \{e_0; \tau_1 - 0,5 \cdot t_1\}$, analog ergibt sich das Zeitfensterende $l_i := \min \{l_0; \tau_1 + 0,5 \cdot t_1\}$. Dabei ist sicherzustellen, daß jeder Kundenauftrag zulässig bedient werden kann. Jeder Kunde muss vor dem Ende seines Zeitfensters erreicht werden können. Nach Abschluss der Bedienung muss außerdem ausreichend Zeit verfügbar sein, damit das Lieferfahrzeug vor dem Ende des Planungshorizonts wieder das Depot erreichen kann. Es muss also für alle Knoten $i \in C$ gelten: $e_0 + t_{0i}(e_0) \leq l_i$ und $\max \{e_i; e_0 + t_{0i}(e_0)\} + s_i + t_{i0}(\max \{e_i; e_0 + t_{0i}(e_0)\} + s_i) \leq l_0$. Damit zulässige Lösungen auf Basis des Verkehrsnetzwerks ermittelt werden können, wird für jeden Kundenauftrag die Zulässigkeit einer Pendeltour zwischen dem Depot und dem Kunden überprüft. Ist ein Kunde nicht rechtzeitig zu erreichen, oder reicht die Zeit nach der Belieferung nicht aus, um rechtzeitig in das Depot zurückzukehren, muss das Kundenzeitfenster modifiziert werden. Dies erfolgt, indem solange ein neues Zeitfenster generiert und wiederum auf seine Zulässigkeit überprüft wird, bis ein zulässiges Zeitfenster vorliegt.

- Servicedauer: Die Servicedauern s_i entsprechen für alle Kunden $i \in C$ konstant 50 Zeiteinheiten.

5.4 Parameter der untersuchten Verfahren

Alle untersuchten Verfahren wurden in C++ implementiert. Um Lösungen zu visualisieren wurde eine Oberfläche in Visual Basic programmiert, die die jeweilige Problemstellung und Parameterausprägung einliest, die Lösungsverfahren aufruft und die Ergebnisse dokumentiert. Ein Großteil der Experimente wurde auf einem Pentium IV PC 2,4 Ghz mit 512MB Arbeitsspeicher durchgeführt. Weiterhin wurden ein Athlon64-3500+, ein Pentium M 730 (mit 1,6 Ghz), ein Pentium IV 2,8 Ghz und ein Pentium IV 2 Ghz für einen Teil der Untersuchungen verwendet. Um die Rechenzeiten vergleichbar zu gestalten, wurden alle benötigten Rechenzeiten normiert. Dazu wurde auf allen verwendeten PCs die Einfügeheuristik INS, ergänzt durch ein Cross-Exchange

Verbesserungsverfahren mit first-fit Auswahlregel für die Probleminstanz E2 gerechnet. Aus den resultierenden Rechendauern wurden Multiplikatoren für die Rechenzeiten auf unterschiedlichen PCs ermittelt. Alle Rechenzeiten sind in Sekunden genannt und beziehen sich auf die reinen Rechendauern der Lösungsverfahren auf einem Pentium IV PC 2,4 Ghz PC. In Tabelle 5.7 sind die ermittelten Faktoren für die Umrechnung von Rechenzeiten auf unterschiedlichen Computern aufgeführt.

	Pentium IV 2 Ghz	Pentium IV 2,4 Ghz	Pentium IV 2,8 Ghz	Pentium M 730	Athlon64- 3500+
Faktor	0,9489	1,0000	1,3367	1,4251	1,7119

Tabelle 5.7: Faktoren für die Normierung von Rechenzeiten auf unterschiedlichen Computern

Parameter des Verfahrens des besten Nachfolgers

Für die experimentelle Analyse des Verfahrens des besten Nachfolgers (im folgenden: NN) zu dTP wird der Abstandswert (vgl. Algorithmus 6 auf Seite 61) aus Abschnitt (4.2.2) verwendet. Das Verfahren des besten Nachfolgers benötigt darüber hinaus keine Parameter.

Um die Lösungsgüte zu erhöhen, werden die mit dem NN-Verfahren ermittelten Lösungen durch eine CX-Nachbarschaftssuche verbessert. Dabei wird $CX_{iter}^{max} := 2$ und $CX_{iter}^{max} := 7$ mit first-fit Auswahlvorschrift sowie $CX_{iter}^{max} := 2$ mit best-fit untersucht.

Parameter der Einfügeheuristik

Die Einfügeheuristik von Solomon (1987) besitzt zur Ermittlung der „Einfügekosten“ vier Steuerparameter: κ_1 , κ_2 , κ_3 und κ_4 .⁶ Für die folgenden Experimente gilt $\kappa = (\kappa_1, \kappa_2, \kappa_3, \kappa_4) = (1, 1, 1, 0)$, da mit dieser Parametereinstellung bei Vorab-Experimenten die besten Ergebnisse erzielt wurden. Diese Parameterausprägungen wurden von Solomon (1987) und Gietz (1994) ebenfalls empfohlen.⁷

Die mit dem INS-Verfahren ermittelten Lösungen werden durch eine CX-Nachbarschaftssuche verbessert. Dabei wird wiederum $CX_{iter}^{max} := 2$ und $CX_{iter}^{max} := 7$ mit first-fit Auswahlvorschrift sowie $CX_{iter}^{max} := 2$ mit best-fit untersucht.

Parameter des Ameisenkolonialgorithmus von Donati et al. (2003a)

Der Ameisenkolonialgorithmus von Donati et al. (2003a) (im folgenden MACS) bietet eine Vielzahl von Einstellungsparametern. Für die experimentelle Analyse wird auf

⁶vgl. in Abschnitt 4.2.3, Algorithmus (8) auf Seite 64

⁷vgl. Solomon (1987), S. 263f.

die Empfehlungen bezüglich der Parametereinstellungen von Donati et al. (2003a) zurückgegriffen.⁸

Pro Kolonie versuchen $anz_{ameisen} := 10$ Ameisen nacheinander eine Lösung zu konstruieren, bevor ein globales Pheromonupdate stattfindet.⁹ Der Ausbeutungs-Parameter q_0 , der die Wahrscheinlichkeit ausdrückt, daß als nächstes derjenige Kunde j besucht wird, für den die Auswahlwahrscheinlichkeit p_j am größten ist, wird von Donati et al. (2003a) mit $q_0 := 0,9$ festgelegt.¹⁰ Der Gewichtungsfaktor β der lokalen Komponente im Verhältnis zum Pheromonwert wird mit $\beta := 1$ festgelegt.¹¹ Die Pheromon-Verdunstungsrate ρ beträgt $\rho := 0,9$, d.h. nur 10% der Pheromonwerte bleiben nach einem Update liegen.¹²

Der Parameter γ beurteilt die Vorteilhaftigkeit einer Lösung. Die Ausprägung $\gamma := 0,1$ führt dazu, daß eine Lösung mit $travel(X) > travel(X^{best}) \cdot (1 + \gamma)$ sofort, ohne CX-Verbesserungsverfahren verworfen wird.¹³

Für dynamisch-deterministische Tourenplanungsprobleme schlagen Donati et al. (2003a) vor, eine mehrperiodige Pheromonmatrix („time subspaces“) zu verwenden.¹⁴ Da Donati et al. (2003a) allerdings nicht auf sinnvolle Ausprägungen für die Anzahl der „Perioden“ eingehen, wird für die experimentelle Analyse für jedes Zeitintervall mit einem konstanten Kategorie-Multiplikator (vgl. Tabelle 5.1) eine eigene Pheromonmatrix verwendet. Die Probleminstanzen der Gruppen 1 und 2 (A1, A2, B1, ..., F1 und F2) erhalten eine fünfperiodige Pheromonmatrix, für die Probleminstanzen A3, B3, ..., F3 wird entsprechend eine siebenperiodige Pheromonmatrix verwendet.

Weiterhin muss ein geeignetes Abbruchkriterium für MACS gefunden werden. Eine zeitliche Vorgabe für die Rechendauer ist für die neuen Probleminstanzen aufgrund der unterschiedlichen Anzahl von Kundenaufträgen, der unterschiedlichen Freiheitsgrade bei der Zuordnung zu Fahrzeugen und der Belieferungsreihenfolge nicht zweckmäßig. Es kann nicht gewährleistet werden, daß der Ameisenkolonialgorithmus nach einer konstanten Rechendauer konvergiert ist. Vorabuntersuchungen haben ergeben, daß bei MACS davon ausgegangen werden kann, daß das Verfahren konvergiert ist, wenn insgesamt von beiden Kolonien $anz_{iter}^{max} = 4000$ Iterationen gerechnet wurden, ohne daß eine neue beste Lösung X^{best} gefunden wurde. Da es sich bei MACS um einen parallelen Algorithmus handelt, müssen nicht notwendigerweise von ACS-VEI und ACS-TIME je 2000 Iterationen durchgeführt werden. Der MACS-Algorithmus wird vielmehr beendet,

⁸vgl. Donati et al. (2003a), S. 5-7

⁹vgl. Donati et al. (2003a), S. 7

¹⁰vgl. Donati et al. (2003a), S. 5

¹¹vgl. Donati et al. (2003a), S. 5 (implizit, da kein Gewichtungsfaktor gegeben)

¹²vgl. Donati et al. (2003a), S. 6

¹³vgl. Donati et al. (2003a), S. 6

¹⁴vgl. Donati et al. (2003a), S. 12

sobald insgesamt 4000 Iterationen gerechnet wurden, ohne daß eine neue beste Lösung gefunden wurde.

Parameter des Ameisenkolonialgorithmus dynACO

Nach umfangreichen Vorabexperimenten haben sich folgende Parameterausprägungen als besonders gut geeignet zur Lösung von dynamisch-deterministischen Tourenplanungsproblemen erwiesen.

Lokale Komponente: Mit Ausprägungen für $\lambda := (0, 10, 15, 5)$ lassen sich zuverlässig gute Ergebnisse erzielen. Dabei steht $\lambda := (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ für die Ausprägungen der Gewichtungsfaktoren, die für die Ermittlung der Werte der lokalen Komponente η_{ij} (vgl. Abschnitt 4.5.3.1) benötigt werden. Die Entfernung zwischen i und j spielt für die lokale Komponente keine Rolle ($\lambda_1 := 0$). Die Fahrzeit, die von i zu j bei Abreise bei i zum Zeitpunkt τ anfällt, ist relativ wichtig ($\lambda_2 := 10$), allerdings wird die erwartete Wartezeit vor Beginn der Bedienung von Kunde j noch stärker gewichtet ($\lambda_3 := 15$). Schließlich wird der zeitliche Abstand bis zum Ende des Belieferungsintervalls von j berücksichtigt, allerdings mit geringerer Gewichtung als Fahr- und Wartezeit ($\lambda_4 := 5$).

Der Gewichtungsfaktor π , der die Bedeutung der lokalen Komponente η_{ij} zu dem Pheromonwert ϕ_{ij} während der Konstruktionsphase gewichtet, erhält die Ausprägung $\pi := 1,4$. Mit einer „Pheromonerhaltungsrate“ von $\rho := 0,65$ wurden bei Vorabuntersuchungen die besten Ergebnisse erzielt.

Vorabexperimente haben gezeigt, daß durch die Beschränkung des CX-Verbesserungsverfahrens auf vielversprechende Lösungen die Rechenzeiten deutlich reduziert werden können, ohne daß das Verfahren zu frühzeitiger Konvergenz tendiert. Der Faktor $\gamma := 0,4$ hat sich dabei als besonders gut geeignet erwiesen. Wurden niedrigere Werte von γ angesetzt, konvergiert dynACO häufig vorzeitig. Bei Werten von $\gamma > 0,4$ werden bei deutlich ansteigenden Rechenzeiten keine wesentlich besseren Lösungen erzielt.

Da dynACO, wie in Vorabuntersuchungen festzustellen war, ohne Rückgriff auf CX-Verbesserungsverfahren nur sehr schlechte Ergebnisse erzielen kann, wird im Rahmen der experimentellen Analyse ein besonderes Augenmerk auf die Parametrisierung der CX Nachbarschaftssuche gelegt werden. Die CX-Nachbarschaft wird für die experimentelle Analyse auf Abschnitte mit $CX_{\text{abschnitt}}^{\text{max}} = 2$ bzw. $CX_{\text{abschnitt}}^{\text{max}} = 7$ Kundenaufträgen beschränkt.¹⁵ Bei der CX-Nachbarschaftssuche mit

¹⁵Taillard et al. (1997) beschränkt sich bei seinen Experimenten zum VRPTW auch auf eine CX-Nachbarschaft mit einer maximalen Abschnittslänge von 7 sukzessive nacheinander zu bedienenden Kundenaufträgen, vgl. Taillard et al. (1997) S. 182. Es ist zu beachten, das in diesem Fall die CX-Nachbarschaft bei langen Touren (mit mehr als 14 Kunden auf einer Tour) nicht die vollständige 2-opt* Nachbarschaft enthält.

$CX_{abschnitt}^{max} = 2$ wird eine first-fit und eine best-fit Auswahlregel untersucht, bei $CX_{abschnitt}^{max} = 7$ wird nur die first-fit Auswahlregel untersucht, da bei best-fit die Rechenzeiten sehr hoch sind.

Nicht jede Ameise wird die CX-Nachbarschaftssuche solange durchführen, bis eine CX-optimale Lösung erreicht wird, da dies gerade bei relativ schlechten Ausgangslösungen sehr zeitaufwendig ist. Um Rechenzeit zu sparen wird die Nachbarschaftssuche nach CX_{iter}^{max} Iterationen beendet.

Da dynACO in jeder Iteration eine zulässige Lösung generiert und der Zeitbedarf für eine Iteration dementsprechend höher als bei MACS ausfällt, wird angenommen, daß das dynACO Verfahren konvergiert ist, wenn $anz_{iter}^{max} = 1000$ Iterationen durchgeführt wurden, ohne daß eine neue, bessere Lösung gefunden wurde. Vorabexperimente haben ergeben, daß bei dieser Größenordnung von konvergierten Lösungen ausgegangen werden kann.

5.5 Ergebnisse und Interpretation der experimentellen Analyse zu dTP

5.5.1 Ergebnisse der experimentellen Analyse

Die untersuchten Verfahren werden folgendermassen abgekürzt: dynACO-CX[$CX_{abschnitt}^{max}$]-[CX_{iter}^{max}] [first-fit/best-fit]; INS-CX[$CX_{abschnitt}^{max}$]-[first-fit/best-fit]; usw. Im folgenden steht „dynACO-CX2-25 first-fit“ beispielsweise für den Ameisenkoloniealgorithmus dynACO, mit den Parameterausprägungen $CX_{abschnitt}^{max} = 2$ und $CX_{iter}^{max} = 25$. Für jede der 18 Probleminstanzen A1 bis F3 werden die Verfahren NN, INS, MACS und dynACO mit folgenden Parameterausprägungen untersucht (vgl. Tabelle 5.8):

Daraus ergeben sich 17 Verfahren je dTP-Problem Instanz, bei 18 Problem Instanzen werden insgesamt 306 Tourenpläne ermittelt. Zur Beurteilung der Vorteilhaftigkeit von Algorithmen wird in dieser Arbeit ein relativer Vergleich vorgenommen, indem die Lösung des jeweils betrachteten Verfahrens der besten Lösung, die im Rahmen der experimentellen Analysen ermittelt wurde, gegenübergestellt wird.

In den folgenden Tabellen werden die Ergebnisse der experimentellen Analyse nach Problem Instanzen getrennt aufgeführt. Die Beurteilung der Ergebnisse erfolgt anhand der lexikographischen Zielordnung, die in Abschnitt (2.2.3.2) auf Seite 38 beschrieben wurde. Die Ergebnisse sind der Lösungsgüte nach absteigend sortiert, wobei das vorteilhafteste Verfahren an oberster Stelle in den Tabellen aufgeführt wird.

Verfahren	$CX_{\text{abschnitt}}^{\text{max}}$	$CX_{\text{iter}}^{\text{max}}$	Auswahlvorschrift
NN			
NN-CX	2		first-fit
NN-CX	2		best-fit
NN-CX	7		first-fit
INS			
INS-CX	2		first-fit
INS-CX	2		best-fit
INS-CX	7		first-fit
MACS	2		first-fit
MACS	7		first-fit
dynACO	2	25	first-fit
dynACO	2	50	first-fit
dynACO	2	10	best-fit
dynACO	2	25	best-fit
dynACO	2	50	best-fit
dynACO	7	5	first-fit
dynACO	7	10	first-fit

Tabelle 5.8: Parameterkombinationen CX Nachbarschaftssuche für dynACO

5.5.1.1 Probleminstanzen A - enge Zeitfenster, geringe Fahrzeugkapazität

Problem A1: geringe Dynamik, weniger dichtes Netzwerk In Tabelle 5.9 sind die Ergebnisse der untersuchten Verfahren zu Probleminstanz A1 aufgeführt. Dabei steht $CX_{\text{ab.}}^{\text{m.}}$ für $CX_{\text{abschnitt}}^{\text{max}}$.

Die besten Ergebnisse mit einer Gesamtbediendauer ($\text{sched}(X)$) von 7888,41 ZE für die Probleminstanz A1 liefert der in Abschnitt 4.5 vorgestellte neue Ameisenalgorithmus dynACO in der Parameterkonstellation dynACO-CX2-25 best-fit. Eine nur unwesentlich höhere Gesamtbediendauer ($\text{sched}(X) = 7953,9$ ZE) wird mit dynACO-CX7-10 first-fit erreicht, wobei allerdings die Rechenzeit mit 37,25 anstelle von 106,62 Minuten erheblich geringer ausfällt. dynACO-CX2-50 best-fit liefert in noch kürzerer Zeit (21,69 Minuten) eine ebenfalls gute Lösung, die eine nur 1,1% höhere Gesamtbediendauer aufweist als die beste bekannte Lösung.

Das beste Ergebnis unter den Referenzverfahren ($\text{sched}(X)$ ist 1,69% höher als die Bediendauer der besten Lösung zur Probleminstanz A1) liefert das Verfahren des besten Nachfolgers (NN) ergänzt durch eine CX-Nachbarschaftssuche mit einer maximalen Abschnittslänge von $CX_{\text{abschnitt}}^{\text{max}} = 2$ Kundenaufträgen, bei der in jeder Iteration die jeweils beste Lösung gewählt wird (NN-CX2 best-fit).

In Verbindung mit der CX-Nachbarschaftssuche ist NN der INS-Heuristik für die Probleminstanz A1 in einem Fall überlegen (NN-CX2 best-fit), unter sonst gleichen Bedingungen liefert INS bessere Ergebnisse. Ein Vergleich der Lösungen der

Verfahren		$CX_{ab.}^m$	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	25	5	7888,41	4094,40	9445,21	106,62
dynACO	mit CX	7	first-fit	10	5	7953,90	3967,54	9225,35	37,25
dynACO	mit CX	2	best-fit	50	5	7975,81	3994,07	9419,90	21,69
NN	mit CX	2	best-fit		5	8021,23	4372,78	10189,66	38,79
MACS	mit CX	7	first-fit		5	8031,24	4239,65	10246,23	106,88
dynACO	mit CX	2	best-fit	10	5	8048,50	3928,89	9070,43	28,35
dynACO	mit CX	2	first-fit	25	5	8069,45	4482,55	10670,49	75,95
INS	mit CX	2	first-fit		5	8080,84	4862,09	11445,14	2,34
INS	mit CX	2	best-fit		5	8083,33	3873,65	9085,14	4,43
dynACO	mit CX	7	first-fit	5	5	8091,93	3698,72	8536,77	28,49
dynACO	mit CX	2	first-fit	50	5	8129,35	4501,73	10589,76	47,66
NN	mit CX	2	first-fit		5	8190,77	4522,55	10436,35	12,14
MACS	mit CX	2	first-fit		5	8194,94	3770,04	8860,82	135,19
INS	mit CX	7	first-fit		5	8340,38	3849,59	9563,71	4,32
NN	mit CX	7	first-fit		6	8745,03	3847,56	9302,54	13,29
INS					6	9762,27	3870,03	9246,64	0,32
NN					10	15012,67	7483,13	18793,05	0,29

Tabelle 5.9: Ergebnisse Problem A1 (dünnes Netzwerk, wenig Dynamik)

einfachen Heuristiken, ergänzt durch eine CX-Nachbarschaftssuche zeigt, daß eine Abschnittslänge $CX_{abschnitt}^{max} = 7$ für die Probleminstanz A1 nicht von Vorteil ist. Es liegt die Vermutung nahe, daß INS-CX7 first-fit und NN-CX7 first-fit in einem lokalen Optimum enden, welches durch INS-CX2 best-fit, INS-CX2 first-fit, NN-CX2 best-fit und NN-CX2 first-fit nicht erreicht wird. Bei einer Betrachtung der Rechenzeiten fällt auf, daß nur die INS Heuristik in der Lage ist im einstelligen Minutenbereich gute Lösungen zu liefern: INS-CX2 first-fit benötigt nur 2,3 Minuten Rechendauer, die Gesamtbediendauer von 8080,84 ZE liegt dann etwa 2,4% über der Bediendauer der besten bekannten Lösung. Bei der Betrachtung der Rechendauer der Varianten von dynACO ist auffällig, daß die Rechendauer nicht notwendigerweise von der Anzahl der Iterationen CX je Ameise abhängig ist. Mit zunehmender Iterationenzahl von CX wird die Suche bereits im Anschluß an die Konstruktionsphase in gute Bereiche des Lösungsraums gelenkt. Dadurch werden weniger Ameisen-Iterationen benötigt um ein lokales Optimum zu finden, gleichzeitig besteht allerdings die Möglichkeit, daß bestimmte lokale Optima frühzeitig in der Pheromonmatrix verstärkt werden und die Suche somit frühzeitig in einem lokalen Optimum konvergiert. Bei der Probleminstanz A1 benötigt dynACO-CX2-25 best-fit beispielsweise erheblich länger als dynACO-CX2-50 best-fit um das Abbruchkriterium zu erreichen, allerdings wird während der höheren Rechendauer auch eine etwas bessere Lösung gefunden.

Der Ameisenkolonialgorithmus von Donati et al. (2003a) (MACS) benötigt besonders

lange zur Ermittlung von Lösungen zur Probleminstanz A1. Die Lösungsgüte von MACS-CX2 ist darüber hinaus auch noch schlechter als die der meisten einfachen Heuristiken mit CX, gleichzeitig wird mit Abstand die höchste Rechendauer aller untersuchten Verfahren für die Probleminstanz A1 benötigt. Das Prinzip der Suche durch zwei unabhängige Kolonien scheint sich nicht zu bewähren. Es wird zwar eine Lösung mit nur 5 Touren gefunden, Lösungen mit 5 Touren werden allerdings auch durch einfache Heuristiken, ergänzt durch ein CX-Verbesserungsverfahren ermittelt.

Erwartungsgemäß sind die Ergebnisse der Einfügeheuristik INS und des Verfahrens des besten Nachfolgers NN ohne CX-Nachbarschaftssuche deutlich schlechter als alle übrigen Ergebnisse. INS liefert eine Lösung mit einer um 23,8% höheren Gesamtbediendauer als die beste Lösung, zudem werden sechs statt 5 Touren benötigt. Die Lösung mit NN ohne Verbesserungsverfahren benötigt sogar 10 Touren und weist eine Gesamtbediendauer auf, die 90,3% über der der besten bekannten Lösung liegt.

Problem A2: geringe Dynamik, dichtes Netzwerk In Tabelle 5.10 sind die Ergebnisse der Lösungsläufe zu Probleminstanz A2 aufgeführt.

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	50	5	7888,41	3774,31	8692,22	84,75
dynACO	mit CX	2	best-fit	25	5	7958,25	4398,76	10163,12	76,07
dynACO	mit CX	2	first-fit	25	5	7961,67	4010,15	9378,77	50,19
NN	mit CX	2	first-fit		5	7986,89	4336,84	10190,23	9,75
MACS	mit CX	2	first-fit		5	8005,76	3877,14	8623,12	125,23
dynACO	mit CX	2	best-fit	10	5	8009,54	4224,31	9821,10	99,05
INS	mit CX	2	first-fit		5	8018,79	4140,66	9987,99	2,37
dynACO	mit CX	2	first-fit	50	5	8018,79	4140,66	9987,99	36,74
INS	mit CX	7	first-fit		5	8057,63	3985,12	9796,66	1,99
dynACO	mit CX	7	first-fit	5	5	8057,63	3985,12	9796,66	19,96
dynACO	mit CX	7	first-fit	10	5	8057,63	3985,12	9796,66	23,26
INS	mit CX	2	best-fit		5	8293,48	3939,53	9232,67	1,93
NN	mit CX	7	first-fit		5	8371,40	4271,00	10114,49	5,77
MACS	mit CX	7	first-fit		6	8164,46	3675,73	8924,56	82,60
NN	mit CX	2	best-fit		6	8366,13	4439,19	10474,99	24,16
INS					6	9878,05	3884,13	9231,39	0,20
NN					10	15007,95	7470,97	18704,44	0,18

Tabelle 5.10: Ergebnisse Problem A2 (dichtes Netzwerk, wenig Dynamik)

Die Problemstellung A2 unterscheidet sich von A1 durch die höhere Anzahl an Pfeilen im Verkehrsnetzwerk. Alle zulässigen Lösungen zu A1 sind auch für A2 zulässig. Es ist daher zu erwarten, daß die resultierenden Lösungen kürzere Wege und somit die gleiche oder eine niedrigere Bediendauer aufweisen als die Ergebnisse für A1. Die

beste Lösung zur Problem Instanz A2 weist tatsächlich die gleiche Gesamtbediendauer ($sched(X)=7888,41$ Zeiteinheiten) wie die beste Lösung zur Problem Instanz A1 auf. Die dabei benötigte Gesamtfahrzeit $travel(X)$ fällt mit 3774,31 ZE anstelle von 4094,4 ZE und die Gesamtentfernung $dist(X)$ mit 8692,22 EE statt 9445,21 EE geringer als bei der besten Lösung zur Problem Instanz A1.

dynACO-CX2-50 best-fit liefert die beste Lösung, gefolgt von dynACO CX2-25 best fit mit einer Gesamtbediendauer von 7958,25 ZE, die 0,9% über der Gesamtbediendauer von dynACO-CX2-50 liegt.

NN-CX2 first-fit liefert das beste Ergebnis unter den Referenzverfahren. Das beste der Referenzverfahren für A1 (NN-CX2 best-fit) ist für die Problem Instanz A2 offenbar nicht geeignet, es liefert das schlechteste Ergebnis aller Verfahren mit CX-Nachbarschaftssuche und benötigt dafür zudem 24 Minuten Rechenzeit.

Die Varianten von dynACO, die CX mit einer Abschnittslänge von 7 Kundenaufträgen zur Verbesserung heranziehen sind für die Problem Instanz A2 schlechter als die Varianten mit $CX_{abschnitt}^{max} = 2$. Bei einer Betrachtung der Rechendauern liegt die Vermutung nahe, daß die Verfahren vorzeitig in lokalen Minima konvergiert sind. Diese Vermutung wird dadurch erhärtet, daß beide Lösungen übereinstimmen und sie zudem der Lösung entsprechen, die mit INS-CX7 first-fit gefunden wird. Dabei ist allerdings INS-CX7 first-fit deutlich schneller, mit 1,99 Minuten Rechendauer.

dynACO-CX2-50 first-fit und INS-CX2 first-fit finden ebenfalls eine gleiche Lösung für die Problem Instanz A2, wobei INS wiederum schneller ist (2,37 statt 36,74 Minuten).

MACS-CX2 liefert das zweitbeste Ergebnis unter den Referenzverfahren, benötigt dafür allerdings erheblich länger (125 Minuten) als alle übrigen untersuchten Verfahren. Die Gesamtbediendauer ($sched(X)$) liegt bei 8005,76 ZE und damit um 1,49% über der $sched(X)$ der besten bekannten Lösung. MACS-CX7 findet für die Problem Instanz A2 nicht einmal eine Lösung mit 5 Touren. Die Einfügeheuristik INS findet auch ohne CX-Nachbarschaftssuchverfahren eine Lösung mit 6 Touren, benötigt dafür anstelle von mehr als 82 Minuten aber nur ca. 12 Sekunden.

Problem A3: hohe Dynamik, dichtes Netzwerk In Tabelle 5.11 sind die Ergebnisse der Lösungsläufe zu Problem Instanz A3 aufgeführt.

Die Problem Instanz A3 unterscheidet sich von A2 durch die dynamischeren Bewertungen der Pfeile im Verkehrsnetzwerk. Es ist zu erwarten, daß ein geeignetes Verfahren die Tageszeiten mit höheren Geschwindigkeiten für Fahrten über relativ große Distanzen zwischen den Kunden nutzt und zu den Tageszeiten mit geringen Geschwindigkeiten verstärkt nahe beieinander liegende Kunden bedient. Insgesamt sind geringere Gesamtbediendauern zu erwarten als bei A1 und A2, da

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	25	5	7519,61	3546,21	10024,46	118,31
dynACO	mit CX	2	best-fit	10	5	7574,91	3713,62	10794,79	97,83
dynACO	mit CX	2	best-fit	50	5	7601,79	3309,97	9050,14	74,64
dynACO	mit CX	7	first-fit	10	5	7664,09	3563,36	9933,92	75,43
dynACO	mit CX	2	first-fit	50	5	7677,31	3545,93	10494,63	49,48
dynACO	mit CX	7	first-fit	5	5	7703,09	3843,98	11057,12	132,05
dynACO	mit CX	2	first-fit	25	5	7730,71	3753,24	9877,64	63,64
INS	mit CX	7	first-fit		5	7733,89	3679,06	10581,10	2,69
NN	mit CX	7	first-fit		5	7769,39	3478,05	9840,23	5,16
NN	mit CX	2	first-fit		5	7844,53	3739,79	10639,01	8,90
NN	mit CX	2	best-fit		5	7948,61	4075,63	11130,29	14,21
INS	mit CX	2	best-fit		5	8036,24	3978,39	11008,81	4,82
MACS	mit CX	7	first-fit		5	8058,44	3648,57	10964,34	101,40
MACS	mit CX	2	first-fit		5	8072,34	3389,14	9931,60	194,26
INS	mit CX	2	first-fit		5	8248,33	3843,11	10633,31	2,51
INS					7	11098,40	3436,30	10555,45	0,22
NN					10	14006,25	5932,54	19074,81	0,19

Tabelle 5.11: Ergebnisse Problem A3 (dichtes Netzwerk, hohe Dynamik)

die Mindestgeschwindigkeiten vergleichbar sind, die Höchstgeschwindigkeit mit einem Multiplikator von 3 statt von 2 allerdings höher ausfallen.

Von allen untersuchten Verfahren liefert dynACO-CX2-25 best-fit mit 5 Touren und $sched(X) = 7519,61$ ZE das beste Ergebnis, benötigt dafür allerdings nahezu zwei Stunden. Die drei untersuchten Varianten von dynACO mit best-fit Suchstrategie sind allen untersuchten Varianten von dynACO mit first-fit Strategie für die Probleminstanz A3 überlegen. Es fällt auf, daß dynACO für alle Versuche zur Probleminstanz A3 alle untersuchten Referenzverfahren übertrifft. Unter den Referenzverfahren ist INS-CX7 first-fit das beste Verfahren. Das beste Ergebnis der Einfügeheuristik, INS-CX7 first-fit, liefert eine Lösung mit einer Gesamtbediendauer $sched(X)$, die 2,85% höher liegt als die der besten bekannten Lösung. Der Ameisenkolonialgorithmus MACS liefert Lösungen, deren Gesamtbediendauer mit 8058,44 ZE und 8072,34 ZE höher ist als die der meisten Lösungen, die auf Basis von INS oder NN resultieren. Dabei benötigt das Verfahren Rechenzeiten von 101,4 bzw. 194,26 Minuten, innerhalb derer dynACO erheblich bessere Lösungen ermittelt.

Werden die Referenzverfahren NN und INS nicht durch eine CX-Nachbarschaftssuche ergänzt, werden 10 (NN) bzw. 7 (INS) Touren anstelle von 5 Touren (übrige Verfahren) benötigt. Die CX Nachbarschaftssuche ermöglicht eine sehr deutliche Verbesserung der Gesamtbediendauer bei INS und bei NN. So beträgt die Bediendauer von INS-CX7 first-fit nur 69,7% der Bediendauer ohne CX (7733,89 ZE statt 11098,4 ZE). Bei

NN-CX7 first-fit beträgt die Bediendauer sogar nur 55% der Bediendauer ohne CX (7769,39 ZE statt 14006,25 ZE). Bei beiden Verfahren werden mit Einsatz des CX-Verbesserungsverfahrens weniger Touren benötigt als ohne.

Zur Probleminstanz A3 wurden zu dynACO jeweils vier weitere Lösungsläufe je Parameterkonstellation durchgeführt, um zu zeigen, daß nicht nur zufällig mit dynACO gute Lösungen ermittelt wurden. Vielmehr liefert dynACO zuverlässig gute bis sehr gute Ergebnisse (vgl. Abschnitt 5.5.2.1).

5.5.1.2 Probleminstanzen B - enge Zeitfenster, hohe Fahrzeugkapazität

Problem B1: Die Probleminstanz B1 ist im Vergleich zu A1 durch eine relativ hohe Fahrzeugkapazität von $Q = 800$ charakterisiert. Es sind Lösungen zu erwarten, die wenige, lange Touren aufweisen. Die Ergebnisse der Lösungsläufe zu Probleminstanz B1 sind in Tabelle 5.12 aufgeführt. Es ist zu beobachten, daß alle Verfahren, die eine CX-Nachbarschaftssuche verwenden, Lösungen mit zwei Touren und einer Gesamtbediendauer ($sched(X)$) von 9554,67 Zeiteinheiten (ZE) ermitteln.

Für die Probleminstanz B1 mit hohen Fahrzeugkapazitäten liefert dynACO-CX7-10 first-fit nach nur 16 Minuten das beste Ergebnis unter allen untersuchten Verfahren, mit einer Gesamtfahrzeit $travel(X)$ von 4271 ZE. Die untersuchten Ausprägungen von dynACO mit längeren Tauschabschnitten ($CX_{abschnitt}^{max} = 7$) liefern besonders gute Lösungen, wobei dynACO-CX7-10 first-fit bei deutlich geringerer Rechendauer (16,07 statt 58,92 Minuten) etwas besser abschneidet als dynACO-CX7-5 first-fit. Es zeigt sich, daß es sinnvoll ist abzuwägen, ob die zusätzliche Rechendauer je Ameisen-Iteration, die für zusätzliche Iterationsschritte des CX-Verbesserungsverfahrens erforderlich ist, eine Einsparung der zur Erreichung des Abbruchkriteriums insgesamt erforderlichen Anzahl an Ameisen-Iterationen ermöglicht. Die bessere Lösungsgüte in den ersten Ameisen-Iterationen führt dazu, daß gute Lösungsbestandteile „schneller“ in der Pheromonmatrix verstärkt werden. Offenbar ist dies für die Probleminstanz B1 von Vorteil: fünf zusätzliche Iterationen CX je Ameise verbessern die konstruierte Lösung bereits frühzeitig, sodaß anschließend insgesamt weniger Ameisen-Iterationen erforderlich sind um zu einer guten Lösung zu gelangen.

Auch dynACO-CX2 best-fit mit $CX_{abschnitt}^{max} \in \{10, 25, 50\}$ liefert Lösungen, die durch keines der Referenzverfahren übertroffen werden. Werden je Ameisen-Iteration 50 Iterationen des CX-Verbesserungsverfahrens statt 25 Iterationen durchgeführt, erhält man die gleiche Lösung. Offenbar sind 25 Iterationen je Ameise ausreichend, zumal sich auch die Rechenzeiten nur minimal unterscheiden.

Die Varianten dynACO-CX2-25 first-fit und dynACO-CX2-50 first-fit sind für Probleme mit großen Fahrzeugkapazitäten und engen Zeitfenstern nicht

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	7	first-fit	10	2	9554,67	4271,00	9758,96	16,07
dynACO	mit CX	7	first-fit	5	2	9554,67	4274,37	9717,39	58,92
dynACO	mit CX	2	best-fit	10	2	9554,67	4295,61	9844,31	29,61
dynACO	mit CX	2	best-fit	50	2	9554,67	4313,24	9917,35	36,09
dynACO	mit CX	2	best-fit	25	2	9554,67	4313,24	9917,35	36,11
NN	mit CX	7	first-fit		2	9554,67	4335,23	9723,23	64,15
INS	mit CX	7	first-fit		2	9554,67	4337,24	9889,69	2,50
dynACO	mit CX	2	first-fit	50	2	9554,67	4343,54	10048,25	23,20
NN	mit CX	2	best-fit		2	9554,67	4358,12	9942,71	41,76
MACS	mit CX	2	first-fit		2	9554,67	4378,71	9681,10	60,35
MACS	mit CX	7	first-fit		2	9554,67	4378,93	9698,92	179,17
NN	mit CX	2	first-fit		2	9554,67	4452,55	10131,56	22,05
INS	mit CX	2	first-fit		2	9554,67	4499,86	10382,97	1,99
dynACO	mit CX	2	first-fit	25	2	9554,67	4533,14	10403,33	23,86
INS	mit CX	2	best-fit		2	9554,67	4642,30	10716,65	2,04
INS					4	14745,88	5283,71	12394,71	0,21
NN					9	28514,68	6893,66	17567,42	0,17

Tabelle 5.12: Ergebnisse Problem B1 (dünnes Netzwerk, wenig Dynamik)

gut geeignet, die Ergebnisse werden teilweise durch einfache Heuristiken mit CX-Nachbarschaftssuche übertroffen.

Bei relativ hohen Rechenzeiten von 60,35 Minuten für MACS-CX2 respektive 179,17 Minuten für MACS-CX7 liefert der Ameisenkoloniealgorithmus von Donati et al. (2003a) mittelmäßige Ergebnisse. Einfache Heuristiken mit CX-Verbesserungsverfahren (INS-CX7, NN-CX7) erreichen bessere Lösungen in kürzerer Zeit.

Erwartungsgemäß ist es bei der Anwendung der einfachen Heuristiken vorteilhaft, große Werte für $CX_{abschnitt}^{max}$ zu wählen. INS-CX7 first-fit und NN-CX7 first-fit liefern Lösungen mit geringeren Gesamtfahrzeiten als INS-CX2 best-fit, INS-CX2 first-fit, NN-CX2 best-fit und NN-CX2 first-fit. INS-CX ist das schnellste Verfahren, mit Rechenzeiten im einstelligen Minutenbereich.

Ohne auf ein CX-Verbesserungsverfahren zurückzugreifen ergeben sich mit INS (NN) Lösungen, die vier (neun) Touren anstelle von zwei Touren erfordern, die Gesamtbediendauer fällt auch dementsprechend höher aus.

Problem B2: In Tabelle 5.13 stehen die Ergebnisse der Lösungsläufe zu Probleminstanz B2.

Die besten Lösungen zu Probleminstanz B2 besitzen die gleiche Tourenzahl und die gleiche Gesamtbediendauer wie die besten Lösungen zu B1. Die Gesamtbediendauer ist

Verfahren		$CX_{ab.}^m$	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	7	first-fit	10	2	9554,67	4226,26	9540,09	35,39
dynACO	mit CX	2	best-fit	25	2	9554,67	4245,81	9520,94	26,16
dynACO	mit CX	2	best-fit	50	2	9554,67	4245,81	9520,94	26,62
dynACO	mit CX	2	best-fit	10	2	9554,67	4254,63	9726,64	18,29
NN	mit CX	7	first-fit		2	9554,67	4291,01	9409,44	17,51
NN	mit CX	2	first-fit		2	9554,67	4310,55	9454,94	49,15
INS	mit CX	7	first-fit		2	9554,67	4321,06	9723,93	6,36
NN	mit CX	2	best-fit		2	9554,67	4327,97	9674,50	50,81
dynACO	mit CX	7	first-fit	5	2	9554,67	4360,61	10002,25	80,67
INS	mit CX	2	best-fit		2	9554,67	4478,72	9908,86	3,53
INS	mit CX	2	first-fit		2	9554,67	4584,81	10326,78	3,16
dynACO	mit CX	2	first-fit	50	2	9554,67	4612,73	11012,46	22,96
dynACO	mit CX	2	first-fit	25	2	9554,67	4670,33	10872,40	33,93
INS					3	14104,85	5399,38	12643,08	0,22
MACS	mit CX	2	first-fit		5	17324,70	6151,15	15592,85	127,89
MACS	mit CX	7	first-fit		6	25385,42	6739,31	16561,94	73,65
NN					9	28501,35	6874,48	17504,47	0,18

Tabelle 5.13: Ergebnisse Problem B2 (dichtes Netzwerk, wenig Dynamik)

auch bei B2 für alle guten Lösungen übereinstimmend, sie beträgt wie bei B1 9554,67 ZE. Die zusätzlich berücksichtigten Pfeile erlauben offenbar nicht, die Touren insgesamt schneller zu beenden. Allerdings ermöglichen die zusätzlichen Pfeile Lösungen mit kürzeren Wegen und geringeren Fahrzeiten bei gleicher Bediendauer ($travel(X) = 4226,26$ ZE bei dynACO-CX7-10 first-fit statt 4271 ZE), wie bereits der Vergleich von A1 mit A2 gezeigt hat.

Das beste Ergebnis liefert wie für die Probleminstanz B1 dynACO-CX7-10 first-fit, mit zwei Touren, einer Gesamtbediendauer von 9554,67 ZE und einer Gesamtfahrzeit von 4226,26 ZE. dynACO-CX2-25 best-fit und dynACO-CX2-50 best-fit liefern ebenfalls gute Ergebnisse, wobei die Gesamtfahrzeit bei 4245,81 ZE liegt. Auffällig ist, daß die Lösungen von dynACO-CX2-25 best-fit und dynACO-CX2-50 best-fit übereinstimmen. dynACO-CX2-25 best-fit ist dabei allerdings etwas schneller, das ist auf CX-Iterationen zurückzuführen, die später verworfen wurden. Offensichtlich sind $CX_{iter}^{max} = 25$ CX-Iterationen je Ameise auch für die Probleminstanz B2 ausreichend. dynACO-CX2-25 first-fit und dynACO-CX2-50 first-fit liefern keine zufriedenstellenden Lösungen.

Die einfachen Heuristiken NN und INS liefern, wie auch bei B1, besonders gute Ergebnisse, wenn $CX_{abschnitt}^{max} = 7$ gewählt wird, also längere Abschnitte miteinander vertauscht werden können. Allerdings liefert NN-CX2 first-fit ein besseres Ergebnis als INS-CX7 first-fit. Die Rechendauer von INS liegt im einstelligen Minutenbereich,

während NN teilweise länger benötigt eine CX-optimale Lösung zu ermitteln als dynACO. NN-CX2 best-fit benötigt 49 Minuten, gute Lösungen werden mit dynACO nach 18 bis 35 Minuten Rechendauer ermittelt.

Werden INS und NN ohne CX-Verbesserungsverfahren untersucht, werden keine Lösungen mit 2 Touren gefunden. Die Ergebnisse von MACS sind, bei sehr hohen Rechenzeiten (MACS-CX2: 127,89 Minuten, MACS-CX7: 73,65 Minuten), sogar schlechter als INS ohne CX-Nachbarschaftssuchverfahren. MACS findet keine Lösung mit 2 Touren. Auffällig ist, daß MACS bei der Problem Instanz B1 mittelmäßig abgeschnitten hat. Offensichtlich schneidet MACS schlecht ab, wenn mehr alternative Wege zwischen den Knoten des Verkehrsnetzwerks zur Verfügung stehen. Die Lösungen, die MACS zur Problem Instanz B1 liefert, sind auch für B2 zulässig und erheblich besser.

Problem B3: Tabelle 5.14 stellt die Ergebnisse der Lösungsläufe zu Problem Instanz B3 dar.

Verfahren		$CX_{ab}^{m.}$	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	50	2	8068,71	3507,27	9991,20	37,14
dynACO	mit CX	2	best-fit	10	2	8142,41	3834,48	10882,04	31,41
dynACO	mit CX	2	best-fit	25	2	8167,54	3476,49	9814,00	32,70
dynACO	mit CX	2	first-fit	50	2	8167,54	3499,83	9770,94	45,84
NN	mit CX	7	first-fit		2	8167,79	3470,31	10109,83	20,18
MACS	mit CX	7	first-fit		2	8237,26	3328,18	9551,97	120,28
INS	mit CX	7	first-fit		2	8237,26	3354,06	9827,13	32,77
dynACO	mit CX	2	first-fit	25	2	8237,26	3589,98	10288,66	48,20
dynACO	mit CX	7	first-fit	10	2	8332,31	3832,68	10937,39	186,04
dynACO	mit CX	7	first-fit	5	2	8551,81	3390,91	9663,53	65,75
NN	mit CX	2	best-fit		2	8557,42	3536,02	10263,68	58,15
MACS	mit CX	2	first-fit		2	9044,13	3713,62	10327,84	215,28
NN	mit CX	2	first-fit		2	9044,13	3714,19	10680,74	30,26
INS	mit CX	2	first-fit		3	11550,38	4389,13	12176,90	0,85
INS	mit CX	2	best-fit		3	11550,38	4389,13	12176,90	1,74
INS					3	11816,65	4563,80	12956,52	0,23
NN					8	23869,49	6365,57	19437,48	0,18

Tabelle 5.14: Ergebnisse Problem B3 (dichtes Netzwerk, hohe Dynamik)

Im Unterschied zu B1 und B2 liefert dynACO-CX2-50 best-fit die besten Lösungen zu der Problem Instanz B3. Auch die anderen Varianten von dynACO mit best-fit Auswahlregel und $CX_{abschnitt}^{max} = 2$ sind dynACO mit first-fit Auswahlregel überlegen. Mit den Varianten dynACO-CX7 werden nur mittelmäßige Lösungen erzielt, dabei benötigt dynACO-CX7-10 außerdem über 3 Stunden. Offenbar sind 5 oder 10

CX-Iterationen je Ameise zu wenig für die Probleminstanz B3, selbst mit einer Abschnittslänge von $CX_{\text{abschnitt}}^{\text{max}} = 7$. Ein weiterer auffälliger Unterschied ist, daß, anders als bei B1 und B2, nicht alle Verfahren die gleiche Gesamtbediendauer ermitteln. Die Bediendauern und die Fahrzeiten der Lösungen zu B3 sind insgesamt kleiner als die Bediendauern und die Fahrzeiten bei B1 und B2. Gleichzeitig werden längere Wege zurückgelegt, die Gesamtentfernungen sind bei B3 höher als bei B1 oder B2. Offenbar werden die Zeitabschnitte mit relativ höheren Geschwindigkeiten tendentiell genutzt, um weitere Entfernungen in kurzer Zeit zurückzulegen.

Der Ameisenkolonialgorithmus MACS benötigt hohe Rechenzeiten zur Ermittlung von Lösungen. MACS-CX7 wird von der einfachen Heuristik NN-CX7 first-fit übertroffen, wobei MACS mit 120,27 Minuten Rechendauer etwa sechsmal so lange benötigt wie NN-CX7 first-fit.

INS-CX2 first-fit und INS-CX2 best-fit liefern relativ schlechte Lösungen. Es werden je drei Touren benötigt, die gleiche Anzahl, die INS bereits ohne CX-Verbesserungsverfahren benötigt. Die Lösungsläufe konvergieren sehr schnell (weniger als 2 Minuten) in einem lokalen Optimum. Auffällig ist, daß INS-CX2 first-fit und INS-CX2 best-fit die gleiche Lösung ermitteln, wobei INS-CX2 mit der Auswahlregel best-fit etwa doppelt so lange dafür benötigt. Dies ist darauf zurückzuführen, daß in jeder CX-Iteration mehr Zeit benötigt wird, unzulässige Tauschvorgänge zu überprüfen, nachdem bereits eine verbesserte Nachbarlösung bekannt ist.

NN ohne CX-Verbesserungsverfahren liefert die schlechteste Lösung, mit 8 statt möglichen 2 Touren und einer Gesamtbediendauer von 23869,49 ZE (statt $\text{sched}(X) = 8167,79$ ZE bei NN-CX7 first-fit).

5.5.1.3 Probleminstanzen C - weite Zeitfenster, geringe Fahrzeugkapazität

Problem C1: Die Probleminstanz C1 ist durch weite Zeitfensterrestriktionen bei niedrigen Fahrzeugkapazitäten gekennzeichnet. Im Unterschied zu A1 bestehen größere Freiheitsgrade für die Belieferungsreihenfolge der Kundenaufträge. Durch die weiten Zeitfensterrestriktionen existiert bei C1 eine größere Anzahl von unterschiedlichen zulässigen Zuordnungsmöglichkeiten von Kundenaufträgen zu Lieferfahrzeugen als bei A1. Gleichzeitig beschränkt die Fahrzeugkapazität von $Q = 200$ die Tourlänge. Die Probleminstanzen A1 und C1 sind allerdings nicht unmittelbar miteinander vergleichbar, da sowohl die einzelnen Zeitfenster bei den Kunden als auch die zugehörigen Nachfragemengen unabhängig voneinander erzeugt werden. Die Ergebnisse der Lösungsläufe zu Probleminstanz C1 finden sich in Tabelle 5.15.

Probleminstanz C1 wird durch dynACO CX2-25 best-fit und dynACO CX2-50 best-fit

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	25	5	7030,14	3955,20	9295,20	137,38
dynACO	mit CX	2	best-fit	50	5	7030,14	3955,20	9295,20	138,67
dynACO	mit CX	2	best-fit	10	5	7166,81	4199,50	9851,52	355,02
dynACO	mit CX	7	first-fit	10	5	7203,83	4137,99	9303,86	31,78
NN	mit CX	7	first-fit		5	7249,28	4077,64	9625,06	17,09
INS	mit CX	7	first-fit		5	7315,65	3986,86	8951,13	4,31
dynACO	mit CX	7	first-fit	5	5	7318,16	3992,37	8951,12	117,77
dynACO	mit CX	2	first-fit	25	5	7512,53	4329,50	9865,85	95,32
MACS	mit CX	7	first-fit		5	7515,56	3873,84	8906,48	73,94
INS	mit CX	2	best-fit		5	7516,42	3975,15	9370,40	1,50
dynACO	mit CX	2	first-fit	50	5	7568,66	3970,83	9233,68	70,23
INS	mit CX	2	first-fit		5	7632,21	3815,63	8885,86	1,80
MACS	mit CX	2	first-fit		5	7719,72	3913,18	9118,89	80,53
NN	mit CX	2	first-fit		5	7737,50	3858,53	9050,62	5,63
INS					5	8261,59	3175,28	7062,36	0,23
NN	mit CX	2	best-fit		6	7736,30	4016,50	9210,29	39,40
NN					10	14138,63	8066,21	20178,73	0,18

Tabelle 5.15: Ergebnisse Problem C1 (dünnnes Netzwerk, wenig Dynamik)

gleichermassen gut, in nahezu der gleichen Zeit gelöst. Beide Verfahren finden nach etwa 138 Minuten Rechendauer die gleiche Lösung mit $anz(X) = 5$ Touren und einer Gesamtbediendauer von $sched(X) = 7030,14$ ZE. Aus der Tatsache, daß dynACO-CX2-50 best-fit eine etwas höhere Rechenzeit benötigt als dynACO-CX2-25 best-fit, lässt sich schliessen, daß in den wenigsten Ameiseniterationen eine CX-Verbesserung nach dem best-fit Prinzip mit mehr als 25 Iterationen durchgeführt wurde. dynACO CX2-10 best-fit liefert die nächstbeste Lösung, bei der eine Lösung mit einer Gesamtbediendauer von 7166,81 ZE resultiert (1,9 % über der Gesamtbediendauer der besten Lösung). Dabei steigt die Rechendauer von 137,38 auf 355,02 Minuten an. Offensichtlich reichen 10 Iterationen CX je Ameise für die Probleminstanz C1 nicht aus um die gleiche Lösung wie dynACO-CX2-25 best-fit zu finden. Die hohe Rechendauer ist ein Hinweis darauf, daß vielfach das CX-Verbesserungsverfahren ohne CX-optimale Lösung beendet wird und daher eine höhere Anzahl Ameisen-Iterationen benötigt wird.

Auch dynACO-CX7-10 first-fit ist allen untersuchten Referenzverfahren überlegen. Für dynACO-CX7-10 first-fit werden nur 31,78 Minuten Rechendauer benötigt. dynACO-CX7-5 first-fit, dynACO-CX2-25 first-fit und dynACO-CX2-50 first-fit schneiden relativ schlecht ab.

Die Rechendauer beträgt für INS-CX7 first-fit 4,31 Minuten und für NN-CX7 first-fit 17,09 Minuten. MACS-CX7 ist den einfachen Heuristiken INS-CX7 first-fit und NN-CX7 first-fit in Bezug auf die Lösungsgüte unterlegen, wobei

gleichzeitig die benötigte Rechenzeit mit 73,74 Minuten deutlich größer ausfällt. Die Gesamtbediendauer bei MACS-CX7 liegt mit 7515,56 ZE, 6,9% über der besten bekannten Lösung. MACS-CX2 benötigt sogar 80,53 Minuten zur Lösungsfindung, wobei die resultierende Lösung eine im Vergleich zu dynACO-CX2-25 best-fit eine um 9,8% höhere Gesamtbediendauer aufweist.

NN-CX2 first-fit und NN-CX2 best-fit sind wenig geeignet, gute Lösungen zu der Problem Instanz C1 zu ermitteln. Wird NN-CX2 best-fit betrachtet, fällt auf, daß noch nicht einmal eine Lösung mit 5 Touren ermittelt wird, während INS auch ohne CX-Verbesserungsverfahren bereits eine Lösung mit 5 Touren ermittelt. Interessant ist auch zu sehen, daß die Lösung mit der geringsten Gesamtfahrzeit $travel(X) = 3175,28$ ZE und der geringsten Gesamtentfernung $dist(X) = 7062,36$ EE mit der Einfügeheuristik INS ermittelt wurde. Die beste Lösung bezüglich der lexikographischen Zielordnung weist hingegen eine Gesamtfahrzeit von $travel(X) = 3955,20$ ZE und Gesamtentfernung von $dist(X) = 9295,20$ EE auf. Offenbar besteht ein deutlicher Zielkonflikt zwischen der Gesamtfahrzeit bei variablen Fahrzeiten, der zurückzulegenden Gesamtentfernung und der Gesamtbediendauer.

Problem C2: Die Problem Instanz C2 basiert auf dem Verkehrswegenetzwerk 65b mit weniger dynamischen Bewertungen der Pfeile. Sie ist durch relativ geringe Fahrzeugkapazitäten ($Q = 200$) und weite Zeitfensterrestriktionen gekennzeichnet. In Tabelle 5.16 sind die Ergebnisse der Lösungsläufe zur Problem Instanz C2 aufgeführt.

Die beste Lösung im Rahmen der Lösungsläufe zur Problem Instanz C2 mit einer Gesamtbediendauer von 7088,25 ZE liefert dynACO-CX2-25 best-fit nach 143,50 Minuten. Die zweitbeste Lösung wird von dynACO-CX2-10 best-fit, die drittbeste Lösung von dynACO-CX2-50 best-fit gefunden. Genaugenommen ist die beste Lösung zu C2 mit dynACO-CX2-25 best-fit zur Problem Instanz C1 ermittelt worden, da jede Lösung zu C1 auch eine zulässige Lösung zu C2 ist. Die Bediendauer beträgt dann 7030,14 ZE. Diese Lösung wird allerdings von keinem der untersuchten Verfahren für die Problem Instanz C2 gefunden. Es ist nicht immer von Vorteil, zusätzliche Wege im Verkehrswegenetzwerk zu berücksichtigen, da dadurch der Rechenaufwand ansteigt und nicht notwendigerweise eine bessere Lösung gefunden wird.

Die beste Lösung unter den Referenzverfahren liefert NN-CX7 first-fit. Die Gesamtbediendauer der Lösung, die mit NN-CX7 first-fit ermittelt wurde, beträgt 7226,44 ZE. Die dafür benötigte Rechendauer ist mit 9,28 Minuten erheblich geringer als die Rechendauer von dynACO.

Auffällig sind die hohen Rechendauern von dynACO und MACS im Vergleich zu INS und NN mit CX-Verbesserungsverfahren. Die einfachen Heuristiken ergänzt durch CX-Nachbarschaftssuchverfahren benötigen jeweils weniger als 10 Minuten um

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	25	5	7088,25	3949,23	9052,66	143,50
dynACO	mit CX	2	best-fit	10	5	7131,21	3852,88	8884,94	101,89
dynACO	mit CX	2	best-fit	50	5	7143,09	3823,39	8673,76	102,31
dynACO	mit CX	2	first-fit	50	5	7154,99	3878,05	8829,27	227,05
NN	mit CX	7	first-fit		5	7226,44	4195,82	9412,30	9,28
MACS	mit CX	7	first-fit		5	7375,95	3663,82	8447,91	81,66
dynACO	mit CX	7	first-fit	10	5	7453,32	4325,94	9937,81	21,76
MACS	mit CX	2	first-fit		5	7466,44	3937,45	8976,15	191,55
dynACO	mit CX	7	first-fit	5	5	7519,83	3891,44	8933,81	127,00
dynACO	mit CX	2	first-fit	25	5	7537,05	3761,94	8898,55	50,58
INS	mit CX	2	first-fit		5	7613,79	3755,36	8715,62	3,24
INS	mit CX	2	best-fit		5	7670,61	4254,42	9718,26	2,62
INS	mit CX	7	first-fit		5	7691,93	3842,91	8955,92	2,53
NN	mit CX	2	first-fit		5	7756,67	4027,64	9090,39	9,19
INS					5	8959,02	3873,73	8777,33	0,23
NN	mit CX	2	best-fit		6	7713,61	4413,60	10345,16	50,98
NN					10	14126,97	8054,54	20108,63	0,18

Tabelle 5.16: Ergebnisse Problem C2 (dichtes Netzwerk, wenig Dynamik)

Lösungen zu ermitteln, von NN-CX2 best-fit einmal abgesehen. MACS und dynACO rechnen hingegen deutlich länger (ab 50,58 Minuten).

Wie auch bei C1, erreicht INS für die Probleminstanz C2 bereits ohne CX-Verbesserungsverfahren eine Lösung mit nur 5 Touren, während NN-CX2 best-fit in einem lokalen Optimum landet, das 6 Touren benötigt. Die Lösung, die INS ohne CX-Verbesserungsverfahren findet, ist schlechter als die ebenfalls zulässige Lösung, die INS zur Probleminstanz C1 findet.

Problem C3: Die Probleminstanz C3 weist im Vergleich zu C2 stärkere Schwankungen der Geschwindigkeiten auf den Pfeilen des Verkehrswegenetzwerks 50b im Planungszeitraum auf. Die Ergebnisse der Lösungsläufe zur Probleminstanz C3 finden sich in Tabelle 5.17.

Die beste Lösung mit 5 Touren und einer Gesamtbediendauer von 6759,23 ZE liefert dynACO-CX2-25 best-fit nach 277,21 Minuten Rechenzeit. Das Verkehrswegenetzwerk der Probleminstanz C3 erlaubt es Lösungen mit einer insgesamt kürzeren Gesamtbediendauer als bei C2 zu erzielen. Im Vergleich zu C2 fallen außerdem die höheren Rechenzeiten, die dynACO benötigt, auf. Sie sind im wesentlichen auf den höheren Rechenaufwand bei der Ermittlung von Wegen im Netzwerk zurückzuführen.

Von den Referenzverfahren liefert MACS-CX7 nach 55,11 Minuten die beste Lösung. Die Gesamtbediendauer beträgt 6870,96 ZE, 1,6% über der besten bekannten Lösung

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	25	5	6759,23	3337,24	9780,73	277,21
dynACO	mit CX	2	best-fit	10	5	6769,47	3516,97	10079,92	204,98
dynACO	mit CX	2	best-fit	50	5	6771,09	3773,66	10533,21	242,74
dynACO	mit CX	2	first-fit	50	5	6790,38	3101,18	8912,94	195,77
dynACO	mit CX	7	first-fit	5	5	6825,57	3280,37	9275,79	84,94
MACS	mit CX	7	first-fit		5	6870,96	3295,33	9170,17	55,11
INS	mit CX	7	first-fit		5	6903,57	3139,51	8956,96	7,00
dynACO	mit CX	7	first-fit	10	5	7107,96	3311,11	9645,06	223,23
MACS	mit CX	2	first-fit		5	7117,49	3135,97	9128,11	79,31
NN	mit CX	7	first-fit		5	7205,24	3085,09	8838,91	10,87
dynACO	mit CX	2	first-fit	25	5	7268,25	3389,57	9481,06	58,62
INS	mit CX	2	first-fit		5	7443,45	3064,00	8694,22	6,18
INS	mit CX	2	best-fit		5	7524,94	3262,02	8873,59	7,26
NN	mit CX	2	first-fit		5	7742,85	3685,92	10608,00	9,61
INS					5	8735,20	3091,63	8349,66	0,24
NN	mit CX	2	best-fit		6	6955,29	3943,71	11474,13	36,84
NN					8	11890,30	5813,38	17573,24	0,22

Tabelle 5.17: Ergebnisse Problem C3 (dichtes Netzwerk, hohe Dynamik)

liegt. Damit ist MACS schneller als dynACO.

Von den untersuchten einfachen Heuristiken liefert INS-CX7 first fit die beste Lösung. Nach 7 Minuten resultiert eine Lösung mit einer Gesamtbediendauer $sched(X) = 6903,57$ ZE, sie liegt damit 2,1% über der besten bekannten Lösung. NN-CX7 first-fit ist im Gegensatz zu C1 und C2 zur Lösung der Probleminstanz C3 nicht das beste der Referenzverfahren.

Die Lösungsgüte der einfachen Heuristiken NN und INS ist schlecht, wobei auch für die Probleminstanz C3 gilt, daß INS ohne CX-Verbesserungsverfahren dem Verfahren des besten Nachfolgers NN überlegen ist. Wie bei C1 und C2 erreicht die Lösungsgüte von NN-CX2 best-fit auch bei der Probleminstanz C3 nicht einmal das Niveau von INS ohne CX-Verbesserungsverfahren.

5.5.1.4 Probleminstanzen D - enge Zeitfenster, geringe Fahrzeugkapazität, 33% der Kundenaufträge ohne Zeitfenster

Problem D1: Die Probleminstanz D1 unterscheidet sich von der Probleminstanz A1 dadurch, daß nicht alle Kundenaufträge Zeitfensterrestriktionen besitzen. 67% der Kundenaufträge besitzen ein enges Zeitfenster für die Belieferung, die übrigen 33% dürfen in jedem beliebigen Zeitpunkt innerhalb der Öffnungszeiten des Depots $[e_0; l_0]$ beliefert werden. Die Probleminstanzen A1 und D1 sind allerdings nicht unmittelbar

miteinander vergleichbar, da sowohl die einzelnen Zeitfenster bei den Kunden als auch die zugehörigen Nachfragemengen je Probleminstanz unabhängig voneinander erzeugt wurden. Die Ergebnisse der Lösungsläufe zur Probleminstanz D1 sind in Tabelle 5.18 aufgeführt. Trotz der c.p. höheren Freiheitsgrade liegen die Zeitfenster bei einzelnen Kunden derart ungünstig, daß eine zulässige Lösung mit 5 Touren von keinem der untersuchten Algorithmen gefunden wird.

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	10	6	7676,62	4052,69	9434,73	1202,37
dynACO	mit CX	2	best-fit	25	6	7693,14	4028,24	9386,90	400,22
dynACO	mit CX	2	best-fit	50	6	7703,24	3775,95	8885,88	269,10
dynACO	mit CX	2	first-fit	50	6	7727,11	4580,56	10954,79	305,84
dynACO	mit CX	7	first-fit	10	6	7948,70	4225,91	9849,40	49,43
NN	mit CX	2	best-fit		6	8333,03	3316,90	7970,64	27,56
dynACO	mit CX	2	first-fit	25	6	8378,04	4342,64	10260,73	123,84
INS	mit CX	2	best-fit		6	8429,22	3866,59	8773,36	6,50
NN	mit CX	2	first-fit		6	8535,66	4066,36	9294,09	7,91
INS	mit CX	7	first-fit		6	8547,98	4009,42	9608,33	18,36
NN	mit CX	7	first-fit		6	8602,37	3790,38	8706,79	14,24
MACS	mit CX	7	first-fit		6	8605,51	3498,44	8135,81	92,31
INS	mit CX	2	first-fit		6	8636,98	4279,50	10082,26	4,36
dynACO	mit CX	7	first-fit	5	6	8659,58	3441,49	7912,50	41,71
MACS	mit CX	2	first-fit		6	9376,23	4364,63	10690,76	2108,66
INS					6	11145,15	3989,40	9466,06	0,24
NN					10	16881,26	6726,04	16498,90	0,18

Tabelle 5.18: Ergebnisse Problem D1 (dünnes Netzwerk, wenig Dynamik)

Die beste Lösung liefert dynACO-CX2-10 best-fit mit 6 Touren und $sched(X) = 7676,62$ ZE, gefolgt von dynACO-CX2-25 best-fit. Allerdings ist die hohe Rechendauer von dynACO-CX2-10 best-fit zu beachten, nahezu 20 Stunden werden benötigt um diese Lösung zu ermitteln. Diese Rechenzeit ist für reale Einsatzzwecke zu hoch, wenn die Planung auf Tagesbasis erfolgen soll. In etwa einem Drittel der Zeit liefert dynACO-CX2-25 best-fit die zweitbeste Lösung, die eine Gesamtbediendauer von 7693,14 ZE, 0,2% über der minimal bekannten Gesamtbediendauer aufweist. Auch dynACO-CX7-10 first-fit ist noch allen Referenzverfahren überlegen, wobei die Rechendauer weniger als eine Stunde beträgt. Die Gesamtbediendauer mit dynACO-CX7-10 first-fit liegt aber bereits 3,5% über der Gesamtbediendauer der besten bekannten Lösung. dynACO-CX7-5 first-fit ist zur Lösung der Probleminstanz D1 offenbar nicht geeignet, die ermittelte Lösung ist schlechter als die Ergebnisse der meisten untersuchten Referenzverfahren. Offenbar sind 5 Iterationen CX je Ameisen-Iteration nicht ausreichend.

Unter den Referenzverfahren liefert NN-CX2 best-fit mit einer Gesamtbediendauer von 8333,03 ZE (8,6 % höher ist die der besten bekannten Lösung) die besten Ergebnisse. INS-CX2 best-fit liefert die nächstbesten Ergebnisse unter den Referenzverfahren. Sowohl die Ergebnisse, die mit INS und NN und einer CX-Nachbarschaftssuche mit einer Abschnittslänge von $CX_{abschnitt}^{max} = 7$ erzielt werden, als auch die Ergebnisse der einfachen Heuristiken mit einer CX-Nachbarschaftssuche mit $CX_{abschnitt}^{max} = 2$ und einer first-fit Suche sind unterlegen.

Der Ameisenkolonialgorithmus MACS schneidet für die Problem Instanz D1 relativ schlecht ab, wobei die Variante mit CX2 mit über 35 Stunden (!) Rechendauer die höchste Rechenzeit aller untersuchten Verfahren für D1 aufweist. Die Lösungsgüte von MACS ist deutlich den einfachen Heuristiken INS und NN unterlegen, wenn die gleiche CX-Nachbarschaftssuche verwendet wird. Nur die einfachen Heuristiken INS und NN ohne CX-Nachbarschaftssuchverfahren liefern noch schlechtere Lösungen als MACS-CX2. INS erzielt immerhin auch ohne CX eine Lösung mit der kleinsten bekannten Tourenzahl, wobei $sched(X) = 11145,15$ ZE 45% über der Gesamtbediendauer der besten bekannten Lösung liegt.

Problem D2: Bei der Problem Instanz D2 besitzen wie bei D1 33% der Kundenaufträge keine Zeitfesterrestriktion. Das der Problem Instanz D2 zugrundeliegende Verkehrswegenetzwerk $50b$ besitzt mehr Pfeile als das Verkehrswegenetzwerk der Problem Instanz D1 ($50a$). Dadurch steht eine größere Zahl alternativer Wege im Netzwerk zur Verfügung. Die Ergebnisse der experimentellen Analyse zur Problem Instanz D2 stehen in Tabelle 5.19.

dynACO-CX2-50 best-fit liefert mit 6 Touren und einer Gesamtbediendauer von 7632,04 ZE die beste Lösung für die Problem Instanz D2, gefolgt von den übrigen untersuchten Varianten von dynACO mit best-fit Verbesserungsstrategie. Die Rechenzeit, die dynACO-CX2-10 best-fit benötigt ist dabei erheblich höher als die Rechenzeit mit $CX_{iter}^{max} = 50$ und $CX_{iter}^{max} = 25$. Alle untersuchten Ausprägungen von dynACO liefern bessere Ergebnisse als die untersuchten Referenzverfahren, wobei dynACO-CX2 first-fit zu besseren Ergebnissen führt als dynACO-CX7 first-fit. Allerdings fallen mit dynACO-CX7 die Rechenzeiten geringer aus.

Unter den Referenzverfahren schneidet MACS am besten ab, wobei MACS-CX7 mit $sched(X) = 8020,04$ etwa 5% über der Gesamtbediendauer des besten Verfahrens liegt. Bei etwas geringerer Rechendauer erreicht MACS-CX7 eine etwas bessere Lösung als MACS-CX2. INS-CX2 best-fit und INS-CX7 first-fit sind allen untersuchten Ausprägungen der NN Heuristik überlegen. Alle Verfahren, die auf die CX-Nachbarschaftssuche zurückgreifen, finden Lösungen mit 6 Touren. Werden die Problem Instanzen D1 und D2 im Vergleich betrachtet, fällt auf, daß wiederum INS-CX2

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	50	6	7632,04	4138,76	9786,91	316,33
dynACO	mit CX	2	best-fit	10	6	7678,40	3889,30	9249,99	750,94
dynACO	mit CX	2	best-fit	25	6	7693,81	4121,03	9680,70	356,16
dynACO	mit CX	2	first-fit	25	6	7709,68	4538,45	10592,96	185,10
dynACO	mit CX	2	first-fit	50	6	7758,43	4357,68	9960,46	270,10
dynACO	mit CX	7	first-fit	5	6	7762,19	4380,73	10660,64	88,72
dynACO	mit CX	7	first-fit	10	6	7931,13	3932,14	9040,48	76,78
MACS	mit CX	7	first-fit		6	8020,04	4207,06	9654,15	104,26
MACS	mit CX	2	first-fit		6	8097,27	4030,32	9484,40	110,83
INS	mit CX	2	best-fit		6	8207,43	4474,07	10273,70	5,96
INS	mit CX	7	first-fit		6	8210,12	3938,78	9097,80	3,73
NN	mit CX	2	best-fit		6	8399,16	3436,03	8131,26	34,53
NN	mit CX	7	first-fit		6	8600,95	3765,34	8900,40	6,93
NN	mit CX	2	first-fit		6	8677,40	3897,01	9303,71	14,23
INS	mit CX	2	first-fit		6	9289,80	3842,62	9138,09	4,34
INS					7	12552,77	4094,77	9418,71	0,23
NN					10	16879,77	6718,51	16408,60	0,18

Tabelle 5.19: Ergebnisse Problem D2 (dichtes Netzwerk, wenig Dynamik)

best-fit und NN-CX2 best-fit den Varianten INS-CX7 first-fit und NN-CX7 first-fit überlegen sind.

Die Einfügeheuristik ohne CX-Verbesserungsverfahren konstruiert eine Lösung mit 7 Touren. Diese Lösung ist schlechter als die Lösung zur Probleminstanz D1, die für D2 auch zulässig ist. Die in D2 zusätzlich berücksichtigten Pfeile führen dazu, daß die Einfügeheuristik eine schlechtere Lösung konstruiert, als für D1. Wie sich bei allen Probleminstanzen beobachten lässt, liefert NN ohne CX die schlechteste Lösung. Die resultierende Gesamtbediendauer liegt bei über dem doppelten der besten bekannten Lösung und es werden 10 anstelle von 6 Touren benötigt.

Problem D3: Die Ergebnisse der Versuche zu der Probleminstanz D3 sind in Tabelle 5.20 aufgeführt.

dynACO ist für D3 in den unterschiedlichsten Ausprägungen den Referenzverfahren überlegen. dynACO-CX2-25 best-fit liefert die beste Lösung mit 6 Touren und einer Gesamtbediendauer von 7102,65 ZE nach 282,05 Minuten Rechendauer. Die Lösungsgüte von dynACO-CX2-50 best-fit unterscheidet sich davon nur unwesentlich, die Gesamtbediendauer liegt etwa 0,03% darüber. dynACO-CX2-10 best-fit liefert das nächstbeste Ergebnis. dynACO mit $CX_{abschnitt}^{max} = 2$ und best-fit Suchstrategie ist allen untersuchten Verfahren überlegen.

Wie bei der Probleminstanz D2 ist auch bei D3 MACS-CX7 first-fit das beste

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	25	6	7102,65	3177,84	9821,14	282,05
dynACO	mit CX	2	best-fit	50	6	7104,86	3402,23	9750,10	515,30
dynACO	mit CX	2	best-fit	10	6	7119,42	3151,41	9335,43	324,62
dynACO	mit CX	7	first-fit	10	6	7208,89	3244,30	9269,78	72,49
dynACO	mit CX	7	first-fit	5	6	7248,71	3414,29	9701,82	86,08
dynACO	mit CX	2	first-fit	50	6	7340,54	2932,79	8941,59	265,49
dynACO	mit CX	2	first-fit	25	6	7356,56	3247,79	9880,43	84,23
MACS	mit CX	7	first-fit		6	7394,84	3002,66	8981,89	80,39
INS	mit CX	7	first-fit		6	7426,46	3158,91	9714,46	8,30
NN	mit CX	2	best-fit		6	7631,44	3309,63	10143,95	56,50
NN	mit CX	7	first-fit		6	7707,29	3067,28	9037,42	19,24
MACS	mit CX	2	first-fit		6	7717,46	3216,73	9254,44	224,61
INS	mit CX	2	first-fit		6	7770,05	2839,88	8452,99	12,86
NN	mit CX	2	first-fit		6	7787,34	3220,17	9299,42	19,84
INS	mit CX	2	best-fit		6	7883,19	3420,68	10136,60	14,38
INS					6	11565,71	3429,41	10214,48	0,23
NN					11	18614,44	5609,71	18989,46	0,20

Tabelle 5.20: Ergebnisse Problem D3 (dichtes Netzwerk, hohe Dynamik)

untersuchte Referenzverfahren. Die Gesamtbediendauer beträgt 7394,84 ZE und liegt somit 4,1% über der Bediendauer der besten Lösung mit dynACO. INS-CX7 first-fit liefert von den einfachen Heuristiken das beste Ergebnis nach 8,3 Minuten Rechendauer, mit 6 Touren und einer Gesamtbediendauer von 7424,46 ZE. INS-CX2 best-fit endet nach 14,38 Minuten Rechendauer in einem unvorteilhaften lokalen Optimum.

Das schlechteste Ergebnisse liefert für die Probleminstanz D3 das Verfahren des besten Nachfolgers NN ohne CX-Verbesserungsverfahren, mit 11 Touren und $sched(X) = 18614,44$ ZE.

5.5.1.5 Probleminstanzen E - enge Zeitfenster, geringe Fahrzeugkapazität, 65 Kundenaufträge

Problem E1: Die Probleminstanz E1 basiert auf dem Verkehrsnetzwerk 65a, mit 65 Kundenaufträgen, engen Zeitfensterrestriktionen und geringen Fahrzeugkapazitäten ($Q = 200$). Die Ergebnisse der Lösungsläufe zur Probleminstanz E1 sind in Tabelle 5.21 aufgeführt.

Die besten Ergebnisse zur Probleminstanz E1 liefert nach 225,42 Minuten Rechendauer dynACO-CX2-50 best-fit, mit 7 Touren und einer Gesamtbediendauer von 10339,19 ZE, gefolgt von dynACO-CX2-25 best-fit und dynACO-CX2-10 best-fit, die jeweils höhere Rechenzeiten aufweisen. dynACO-CX7-10 first-fit liefert bereits nach 171,95

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	50	7	10339,19	5017,13	12162,55	225,42
dynACO	mit CX	2	best-fit	25	7	10422,96	5142,52	12651,53	249,02
dynACO	mit CX	2	best-fit	10	7	10595,49	5407,91	12698,50	348,47
dynACO	mit CX	7	first-fit	10	7	10631,87	5076,25	12131,53	171,95
dynACO	mit CX	2	first-fit	25	7	10657,98	4981,20	11772,60	92,04
NN	mit CX	7	first-fit		7	10703,21	5512,09	13504,76	22,59
dynACO	mit CX	2	first-fit	50	7	10807,99	5343,29	12710,35	211,06
MACS	mit CX	7	first-fit		7	10813,55	5484,97	13195,69	139,42
INS	mit CX	2	first-fit		7	11060,14	5070,86	11797,08	4,42
INS	mit CX	7	first-fit		7	11418,01	5185,03	12247,55	3,60
dynACO	mit CX	7	first-fit	5	7	11418,01	5185,03	12247,55	37,53
INS	mit CX	2	best-fit		7	11431,15	5297,12	12516,26	3,66
MACS	mit CX	2	first-fit		7	11638,44	5622,04	13784,63	104,55
NN	mit CX	2	first-fit		7	11716,55	5568,28	13109,93	17,69
INS					7	12127,22	4778,74	11003,28	0,46
NN	mit CX	2	best-fit		8	11451,52	6108,11	14344,48	61,21
NN					14	19142,56	8521,91	21040,57	0,40

Tabelle 5.21: Ergebnisse Problem E1 (dünnnes Netzwerk, wenig Dynamik)

Minuten eine Lösung, die zwar von keinem der Referenzverfahren übertroffen wird, aber die Gesamtbediendauer 2,8% liegt über der Gesamtbediendauer der besten bekannten Lösung. Noch schneller (nach 92,04 Minuten) liefert dynACO-CX2-25 first-fit eine Lösung, die auch von keinem der Referenzverfahren erreicht wird, mit $sched(X) = 10657,98ZE$ liegt die Gesamtbediendauer allerdings 3,08% über der besten bekannten Lösung. Im Vergleich zu der Probleminstanz A1 sind mit nur 15 zusätzlichen Kundenaufträgen sehr starke Steigerungen der Rechendauern verbunden. Die mittlere Rechendauer bei A1 beträgt nur 38,74 Minuten, bei E1 hingegen 99,62 Minuten. Dies ist allerdings zu erwarten gewesen, da die Anzahl der möglichen zulässigen Zuordnungsmöglichkeiten und Belieferungsreihenfolgen mit der Problemgröße überproportional ansteigt.

Unter den Referenzverfahren ist NN-CX7 first-fit das beste Verfahren, gefolgt von MACS-CX7 first-fit. Bei gleicher Tourenzahl beträgt die Gesamtbediendauer 10703,21 ZE respektive 10813,55 ZE, sie liegt 3,5% bzw. 4,6% über der Gesamtbediendauer der besten bekannten Lösung.

NN und INS ergänzt durch CX2 best-fit ist für die Probleminstanz E1 offenbar nicht besonders gut geeignet. NN-CX2 best-fit endet mit einer Lösung, die sogar eine Tour mehr benötigt als die Ausgangslösung von INS ohne CX-Verbesserungsverfahren. NN ohne CX-Verbesserungsverfahren liefert die schlechteste Lösung zur Probleminstanz E1. Mit 14 Touren werden dabei doppelt so viele Touren eingeplant wie die Lösung mit

INS benötigt.

Problem E2: Die Probleminstanz E2 ist durch ein im Vergleich zu E1 dichteres Verkehrswegenetzwerk gekennzeichnet. Die Anzahl, Standorte und Eigenschaften der Kundenaufträge stimmen überein. In Tabelle 5.22 stehen die Ergebnisse der Lösungsläufe zu Probleminstanz E2.

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	50	7	10004,26	5051,48	11578,09	126,68
dynACO	mit CX	2	best-fit	25	7	10127,50	4990,96	11716,12	358,67
dynACO	mit CX	2	best-fit	10	7	10138,04	5163,26	12050,54	368,34
dynACO	mit CX	2	first-fit	25	7	10413,44	5072,43	12191,17	56,28
dynACO	mit CX	2	first-fit	50	7	10486,11	5196,94	12343,70	157,83
INS	mit CX	7	first-fit		7	10555,29	4768,03	10846,19	1,30
dynACO	mit CX	7	first-fit	5	7	10555,29	4768,03	10846,19	21,50
dynACO	mit CX	7	first-fit	10	7	10555,29	4768,03	10846,19	27,32
INS	mit CX	2	best-fit		7	10604,45	5208,14	12557,71	4,55
MACS	mit CX	7	first-fit		7	10733,53	5068,94	12354,63	93,16
NN	mit CX	7	first-fit		7	10825,05	5151,98	12393,52	16,81
NN	mit CX	2	first-fit		7	10972,75	5859,56	13827,82	12,90
INS	mit CX	2	first-fit		7	11046,81	5413,16	12610,23	2,16
NN	mit CX	2	best-fit		7	11110,25	5261,98	12192,77	44,15
MACS	mit CX	2	first-fit		7	11740,09	5003,04	11999,86	8110,60
INS					7	12252,37	4861,62	11105,46	0,28
NN					14	19138,10	8507,25	20900,46	0,24

Tabelle 5.22: Ergebnisse Problem E2 (dichtes Netzwerk, wenig Dynamik)

Die besten Lösungen zur Probleminstanz E2 liefert nach 126,68 Minuten Rechenzeit dynACO-CX2-50 best-fit, mit 7 Touren und einer Gesamtbediendauer von 10004,26 ZE. Die nächstbesten Ergebnisse lassen werden mit dynACO-CX2-50 und dynACO-CX2-10, jeweils mit best-fit Auswahlregel, erzielt. Die Ergebnisse, die mit dynACO-CX2-25 first-fit und mit dynACO-CX2-50 first-fit erzielt werden, übertreffen auch alle Vergleichsverfahren, allerdings liegt die Gesamtbediendauer mit 10413,44 ZE bzw. 10486,11 ZE etwa 4,1% bzw. 4,8% über der Gesamtbediendauer der besten bekannten Lösung. Im Vergleich zur Probleminstanz E1 fällt die Gesamtbediendauer geringer aus, die zusätzlichen Pfeile ermöglichen offenbar deutliche Einsparungen bei den benötigten Fahrzeiten.

dynACO mit den Parameterausprägungen $CX_{laenge}^{max} = 7$ konvergiert für die Probleminstanz E2 sehr schnell. Die Rechendauer beträgt bei dynACO-CX7-5 first-fit 21,5 Minuten und bei dynACO-CX7-10 first-fit 27,32 Minuten. Innerhalb dieser Zeit wird dieselbe Lösung gefunden, die auch mit INS-CX7 first-fit nach

1,3 Minuten gefunden wird. INS-CX7 first-fit ist das beste der untersuchten Referenzverfahren für die Probleminstanz E2. Die Gesamtbediendauer liegt 5,5% über der Gesamtbediendauer der besten bekannten Lösung, wobei die Lösung bereits nach 1,3 Minuten gefunden wird. Damit ist INS-CX7 first-fit nicht nur das beste sondern auch das schnellste unter den Referenzverfahren, die auf ein CX-Verbesserungsverfahren zurückgreifen.

NN-CX findet in allen untersuchten Varianten eine Lösung mit 7 Touren. Die Lösungen zu E1, die mit NN-CX ermittelt wurden, sind für die Probleminstanz E2 auch zulässig und von besserer Qualität.

Mit MACS-CX7 wird eine Lösung erhalten, die von jeder untersuchten Ausprägung von dynACO übertroffen wird, wobei die Rechenzeit 93,16 Minuten beträgt. Besonders auffällig ist die extrem hohe Rechendauer für MACS CX2. Nach über 135 Stunden Rechendauer und mehr als 378.000 Iterationen wurde das Verfahren manuell abgebrochen. Das Ergebnis, das dann resultiert, ist schlechter als das von allen übrigen untersuchten Verfahren, die auf ein CX-Verbesserungsverfahren zurückgreifen.

Erwartungsgemäß liefern NN und INS ohne CX-Verbesserungsverfahren die schlechtesten Ergebnisse für die Probleminstanz E2.

Problem E3: Die Probleminstanz E3 unterscheidet sich von E2 durch unterschiedliche Bewertungen der Pfeile des zugrundeliegenden Verkehrswegenetzwerks 65b. Es sind 65 Kundenaufträge zu bedienen, die alle enge Zeitfensterrestriktionen aufweisen. Die Fahrzeugkapazitäten sind relativ klein ($Q = 200$), wodurch zahlreiche kurze Touren zu erwarten sind. Die Ergebnisse der Lösungsläufe zu Probleminstanz E3 stehen in Tabelle 5.23.

Das beste Ergebnis zur Probleminstanz E3 liefert dynACO-CX2-50 best-fit (mit 7 Touren und einer Bediendauer von 9521,38 ZE), gefolgt von dynACO-CX2-25 best-fit. Die dafür benötigten Rechenzeiten betragen 719,62 bzw. 475,99 Minuten, deutlich mehr als für die Ermittlung der besten Lösungen für die Probleminstanz E2 benötigt wird. Das im Vergleich zur Probleminstanz E2 verändert bewertete Verkehrswegenetzwerk erhöht offenbar die Rechendauer deutlich, gleichzeitig sind, bedingt durch die höheren Geschwindigkeiten, Lösungen mit geringerer Gesamtbediendauer möglich.

Unter den Referenzverfahren liefert NN-CX7 first-fit das beste Ergebnis, mit 7 Touren und einer Gesamtbediendauer von 9910,02 ZE. Die Gesamtbediendauer liegt etwa 4% über der der besten bekannten Lösung. Die Lösung mit NN-CX7 first-fit übertrifft die Lösungsgüte von allen untersuchten Varianten von dynACO mit first-fit Auswahlvorschrift. dynACO-CX7-5 first-fit ist eine Ausprägung von dynACO, die nicht besonders gut geeignet ist: die nach 61,08 Minuten Rechenzeit erhaltene Lösung stimmt mit der Lösung überein, die mit INS-CX7 first-fit bereits nach 8,61 Minuten resultiert.

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	2	best-fit	50	7	9521,38	4385,58	12350,98	719,62
dynACO	mit CX	2	best-fit	25	7	9690,93	4219,54	11717,34	475,99
NN	mit CX	7	first-fit		7	9910,02	4389,72	12488,46	12,23
dynACO	mit CX	2	first-fit	25	7	10019,09	4414,22	12302,55	161,59
dynACO	mit CX	2	first-fit	50	7	10048,80	4773,54	13453,51	194,53
dynACO	mit CX	2	best-fit	10	7	10070,03	4616,10	13802,19	428,37
dynACO	mit CX	7	first-fit	10	7	10214,11	4548,10	13874,11	166,71
INS	mit CX	2	first-fit		7	10264,37	4340,03	11982,26	8,92
INS	mit CX	7	first-fit		7	10279,88	4564,85	12750,00	8,64
dynACO	mit CX	7	first-fit	5	7	10279,88	4564,85	12750,00	61,08
INS	mit CX	2	best-fit		7	10354,79	4421,27	11807,78	7,38
MACS	mit CX	7	first-fit		7	10378,52	4910,22	13558,88	92,13
NN	mit CX	2	best-fit		7	10524,89	4675,34	12910,89	46,41
NN	mit CX	2	first-fit		7	10561,96	4443,74	12005,39	15,30
MACS	mit CX	2	first-fit		7	10726,21	4447,27	12892,71	226,21
INS					7	12479,78	4050,36	11592,66	0,29
NN					13	16631,49	6439,42	19931,07	0,24

Tabelle 5.23: Ergebnisse Problem E3 (dichtes Netzwerk, hohe Dynamik)

Unter den Referenzverfahren ist INS in jeder untersuchten Ausprägung MACS bezüglich der Rechendauer und bezüglich der Lösungsgüte überlegen. MACS-CX2 first-fit ist das schlechteste der untersuchten Referenzverfahren, die auf CX-Verbesserungsverfahren zurückgreifen. Nur INS und NN liefern noch schlechtere Lösungen, wobei INS auch ohne CX-Verbesserungsverfahren immerhin noch eine Lösung mit 7 Touren findet, während bei NN eine Lösung mit 13 Touren resultiert.

5.5.1.6 Probleminstanzen F - enge Zeitfenster, hohe Fahrzeugkapazität, 65 Kundenaufträge

Problem F1: Die Probleminstanz F1 basiert auf dem Verkehrswegenetzwerk 65a, mit 65 Kundenaufträgen, engen Zeitfenstern und relativ großen Fahrzeugkapazitäten ($Q = 800$), so daß wenige Touren mit relativ vielen Kundenaufträgen pro Tour zu erwarten sind. Die Ergebnisse der experimentellen Analyse zu Probleminstanz F1 stehen in Tabelle 5.24.

NN-CX7 first-fit liefert nach 226,25 Minuten Rechendauer das beste Ergebnis für die Probleminstanz F1. Es werden 2 Touren benötigt um alle Kunden zu beliefern. Dabei fällt eine Gesamtbediendauer von 8956,32 ZE an und die Gesamtfahrzeit beträgt 4747,16 ZE. Keine der anderen zu F1 ermittelten Lösungen erreicht eine geringere Gesamtbediendauer, Gesamtfahrzeit oder Gesamtentfernung.

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
NN	mit CX	7	first-fit		2	8956,32	4747,16	11316,68	226,25
dynACO	mit CX	2	best-fit	10	2	8956,32	4912,46	11997,75	61,61
dynACO	mit CX	2	first-fit	50	2	8956,32	4985,91	11804,30	70,19
dynACO	mit CX	7	first-fit	10	2	8956,32	5013,67	11976,75	49,06
dynACO	mit CX	7	first-fit	5	2	9084,97	5599,03	13595,67	180,14
dynACO	mit CX	2	first-fit	25	3	9216,27	5548,37	13575,86	87,74
NN	mit CX	2	first-fit		3	9324,89	4965,00	12311,37	279,91
dynACO	mit CX	2	best-fit	50	3	9866,70	5078,75	12235,93	112,80
dynACO	mit CX	2	best-fit	25	3	9866,70	5078,75	12235,93	113,52
INS	mit CX	2	first-fit		3	10021,06	5002,64	11965,28	3,45
MACS	mit CX	7	first-fit		3	10028,82	5205,78	12723,87	212,73
INS	mit CX	7	first-fit		3	10164,91	4913,60	11810,10	8,14
INS	mit CX	2	best-fit		3	10164,91	4913,60	11810,10	10,07
NN	mit CX	2	best-fit		3	11608,55	5623,44	13439,93	131,85
INS					3	11608,78	5276,00	12703,74	0,62
MACS	mit CX	2	first-fit		5	20734,57	7524,82	18639,96	182,59
NN					8	27128,44	8314,86	20763,20	0,40

Tabelle 5.24: Ergebnisse Problem F1 (dünnes Netzwerk, wenig Dynamik)

Nach deutlich geringerer Rechenzeit liefert dynACO-CX2-10 best-fit das nächstbeste Ergebnis. Bei gleicher Tourenzahl und gleicher Bediendauer wird eine um 1,8% höhere Fahrzeit benötigt. Auch dynACO-CX2-50 first-fit und dynACO-CX7-10 first-fit finden Lösungen mit zwei Touren und der minimalen Gesamtbediendauer. Es werden jeweils Rechenzeiten zwischen 49,06 und 70,19 Minuten benötigt. dynACO-CX7-5 first-fit findet ebenfalls eine Lösung mit nur 2 Touren, bei einer Gesamtbediendauer von 9084,97.

dynACO-CX2-25 best-fit und dynACO-CX2-50 best-fit finden jeweils nur eine Lösung mit 3 Touren. Eine genauere Betrachtung der Ergebnisse zu dynACO-CX2-50 best-fit und dynACO-CX2-25 best-fit zeigt, daß bereits nach wenigen Iterationen der resultierende Tourenplan gefunden wurde. Die Suche konvergiert frühzeitig, weil die Ameisen zu wenig Raum für Verbesserungen besitzen. dynACO CX2-25 first-fit, dynACO CX2-25 best-fit und dynACO CX2-50 best-fit finden keine Lösung mit 2 Touren, sondern konvergieren bei Lösungen mit drei Touren. Das überraschend gute Ergebnis des NN Verfahrens lässt darauf schliessen, daß sich die Parameterauswahl für dynACO für die Problem Instanz F1 noch verbessern lässt.

Alle untersuchten Ausprägungen der Referenzverfahren ausser NN-CX7 first-fit, erzielen Lösungen mit mindestens 3 Touren. Die CX-Nachbarschaftssuche mit best-fit Strategie ist für die Problem Instanz F1 schlechter als eine first-fit Strategie. Es fällt auf, daß die spezialisierte Ameisenkolonie ACS-VEI im Rahmen der Lösungsfindung des

MACS-Algorithmus keine zulässigen Lösungen mit zwei Touren generiert. MACS-CX2 konvergiert sogar bei einer Lösung mit 5 Touren, nach über 3 Stunden Rechendauer. Sogar INS ohne CX-Verbesserungsverfahren ist MACS-CX2 überlegen, mit einer Lösung mit 3 Touren nach nur 0,62 Minuten Rechendauer.

Trotz der ungeeigneten Parameterwahl ist dynACO sowohl INS als auch MACS in Bezug auf die Lösungsgüte bei allen untersuchten Parameterkonstellationen überlegen. Die Rechenzeiten lassen vermuten, daß dynACO vorzeitig konvergiert ist. Diese Vermutung wird bei einer detaillierten Betrachtung der Ergebnisse bestätigt, bereits nach weniger als 100 Iterationen wird der Ergebnis-Tourenplan bei dynACO gefunden. NN und MACS weisen unter den untersuchten Verfahren die höchsten Rechenzeiten auf, die Ausgangslösung ist von derart schlechter Qualität, daß zur Verbesserung sehr viel Zeit in Anspruch genommen wird. Dabei ist erkennbar, daß eine Verbesserung einer Ausgangslösung durch ein CX-Nachbarschaftsuchverfahren bei ähnlicher Rechendauer bessere Ergebnisse liefert als der Ameisenkolonialgorithmus von Donati et al. (2003a).

Problem F2: Die Probleminstanz F2 unterscheidet sich von F1 durch die größere Anzahl an Pfeilen im Verkehrswegenetzwerk 65b, die für die Planung der Auslieferungstouren zur Verfügung steht. Die Ergebnisse der Lösungsläufe zu F2 finden sich in Tabelle 5.25.

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
dynACO	mit CX	7	first-fit	10	2	8942,74	4661,75	11340,00	132,40
NN	mit CX	7	first-fit		2	8942,74	4856,74	11681,51	129,86
MACS	mit CX	7	first-fit		2	8942,74	4881,84	11694,83	221,01
dynACO	mit CX	7	first-fit	5	2	8942,74	4893,92	11568,29	89,54
dynACO	mit CX	2	best-fit	50	2	8942,74	4901,67	11920,96	145,75
dynACO	mit CX	2	best-fit	25	2	8957,12	5099,97	12309,31	108,32
dynACO	mit CX	2	first-fit	25	2	9286,86	4947,80	11576,12	302,77
INS	mit CX	7	first-fit		3	9106,25	4774,52	11322,26	18,17
dynACO	mit CX	2	first-fit	50	3	9255,74	4787,22	11272,93	145,78
dynACO	mit CX	2	best-fit	10	3	9586,82	5033,41	12322,74	129,87
INS	mit CX	2	first-fit		3	10151,33	4638,40	10963,76	11,65
INS	mit CX	2	best-fit		3	10178,95	5005,19	12323,19	9,01
NN	mit CX	2	first-fit		3	10719,43	4544,26	10845,16	129,39
INS					3	11102,30	5509,26	13638,89	0,37
NN	mit CX	2	best-fit		3	11525,88	5595,68	13668,37	89,33
MACS	mit CX	2	first-fit		4	16368,03	7990,13	19718,27	251,50
NN					8	27112,38	8256,47	20434,86	0,25

Tabelle 5.25: Ergebnisse Problem F2 (dichtes Netzwerk, wenig Dynamik)

dynACO-CX7-10 first-fit liefert für die Probleminstanz F2 nach 132,4 Minuten

Rechendauer das beste Resultat. Wie bei der Problem Instanz F1 kommen die beste Tourenpläne mit nur 2 Touren aus. Die besten fünf Verfahren weisen die gleiche Gesamtbediendauer von 8942,74 ZE auf. Die zweitbeste Lösung wird von NN-CX7 first-fit gefunden, gefolgt von MACS-CX7. dynACO-CX7-5 first-fit und dynACO-CX2-50 best-fit finden ebenfalls Lösungen mit einer Gesamtbediendauer von 8942,74 ZE. dynACO liefert, wie bei den Problem Instanzen B1 und B2, mit $CX_{abschnitt}^{max} = 7$ und 10 CX-Iterationen bessere Ergebnisse als mit $CX_{abschnitt}^{max} = 2$.

Alle untersuchten Varianten des Referenzverfahren INS sowie alle Referenzverfahren mit $CX_{abschnitt}^{max} = 2$ sind nicht in der Lage Lösungen mit 2 Touren zu finden. Offenbar ist es für die Problem Instanz F2 von Vorteil das CX-Verbesserungsverfahren mit $CX_{abschnitt}^{max} = 7$ zu verwenden.

Es fällt auf, daß INS sogar ohne CX-Verbesserungsverfahren NN-CX2 best-fit überlegen ist. MACS-CX2 liefert eine noch schlechtere Lösung, und benötigt dabei sogar 4 Touren. Die Kolonie ACS-VEI versagt offenbar bei der Problem Instanz F2, die gute Lösungsgüte von MACS-CX7 scheint auf das CX-Verbesserungsverfahren mit längeren Abschnitten zurückzuführen zu sein, insbesondere weil NN-CX7 first-fit bessere Lösungen erzielt als MACS-CX7. Wie bei allen untersuchten Problem Instanzen liefert NN ohne CX-Verbesserungsverfahren von den untersuchten Verfahren die schlechteste Lösung.

Problem F3: Die Problem Instanz F3 weist 65 Kundenaufträge auf. Das zugrundeliegende Verkehrsnetzwerk 65b ist im Unterschied zu F2 durch stärkere Schwankungen im Zeitablauf gekennzeichnet. In Tabelle 5.26 stehen die Ergebnisse zur Problem Instanz F3. Die besten Ergebnisse zur Problem Instanz F3 liefert NN-CX7 first-fit nach 280,75 Minuten Rechenzeit, mit zwei Touren und einer Gesamtbediendauer von 8486,62 ZE, gefolgt von INS-CX2 best-fit. INS-CX2 best-fit benötigt zur Lösungsfindung nur 10,16 Minuten, wobei die Gesamtbediendauer mit 8554,80 ZE 0,8% über der Gesamtbediendauer der Lösung liegt, die mit NN-CX7 best-fit ermittelt wurde.

Verfahren		CX_{ab}^m	CX_{typ}	CX_{iter}^{max}	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechendauer (min)
NN	mit CX	7	first-fit		2	8486,62	4053,71	11735,17	280,75
INS	mit CX	2	best-fit		2	8554,80	4339,50	11666,90	10,16
INS	mit CX	7	first-fit		2	8554,94	4290,04	11625,14	118,59
dynACO	mit CX	7	first-fit	5	2	8554,94	4378,06	12257,66	525,97
NN	mit CX	2	first-fit		2	8562,94	4298,78	12151,76	191,14
NN	mit CX	2	best-fit		2	8578,05	4870,40	14622,40	100,62
INS	mit CX	2	first-fit		2	8578,60	4107,05	11171,70	113,74
dynACO	mit CX	2	best-fit	10	2	8578,60	4398,00	11974,71	87,87
dynACO	mit CX	2	best-fit	50	2	8578,60	4418,14	12160,07	62,70
dynACO	mit CX	2	best-fit	25	2	8578,60	4418,14	12160,07	62,90
dynACO	mit CX	7	first-fit	10	2	8623,61	4372,06	12100,15	243,77
dynACO	mit CX	2	first-fit	25	2	8645,05	4348,74	11731,49	274,93
dynACO	mit CX	2	first-fit	50	2	8676,40	4275,57	11606,34	269,98
MACS	mit CX	7	first-fit		3	11575,29	4404,83	12934,35	740,84
INS					4	14539,31	4675,15	13047,83	0,34
MACS	mit CX	2	first-fit		4	16278,87	6156,57	16833,50	305,63
NN					8	26525,82	7163,89	20770,50	0,25

Tabelle 5.26: Ergebnisse Problem F3 (dichtes Netzwerk, hohe Dynamik)

Offensichtlich sind die Parameter für dynACO, die sich für die übrigen untersuchten Probleminstanzen bewährt haben, für die Probleminstanz F3 ungeeignet. dynACO ist den Referenzverfahren eindeutig unterlegen. In Abschnitt 5.5.2.2 werden weitere Parameterausprägungen für dynACO untersucht. An dieser Stelle sei vorweggenommen, daß dynACO mit geeigneter Parameterausprägung auch die Lösungsgüte von NN-CX7 first-fit übertrifft und eine neue beste Lösung ermittelt wird.

Es zeigt sich, daß dynACO-CX7-5 first-fit mit der größten Rechendauer (525,97 Minuten) auch das relativ beste Ergebnis liefert. Die erreichte Gesamtbediendauer liegt dann 0,8% über der Gesamtbediendauer von NN-CX7 first-fit. Wesentlich schneller ist die Suche von dynACO-CX2-10 best-fit beendet: nach 87,87 Minuten wird eine Lösung ermittelt, dessen Gesamtbediendauer 1,08% über der besten bekannten Gesamtbediendauer liegt. dynACO in allen untersuchten Ausprägungen, sowie NN und INS mit CX-Verbesserungsverfahren finden Lösungen mit zwei Touren.

Nur das Referenzverfahren MACS und die einfachen Heuristiken ohne CX-Verbesserungsverfahren schneiden schlechter ab als dynACO. Es wird keine Lösung mit nur 2 Touren gefunden. Gleichzeitig benötigt MACS sehr hohe Rechenzeiten. MACS-CX2 erreicht noch nicht einmal die Lösungsgüte von INS ohne CX-Verbesserungsverfahren. Das schlechteste Ergebnis erzielt NN ohne Verbesserungsverfahren.

5.5.2 Parametrische Analyse:

5.5.2.1 Problem Instanz A3: Zusätzliche Versuche zu dynACO

Die relative Vorteilhaftigkeit von verschiedenen Algorithmen zur Lösung von dynamisch-deterministischen Tourenplanungsproblemen wurden beispielhaft anhand von 18 Problem Instanzen mit unterschiedlich großen und unterschiedlich zahlreichen Zeitfensterrestriktionen für die Bedienung der Kunden, unterschiedlichen Fahrzeugkapazitäten und unterschiedlich starken Schwankungen der Fahrzeiten innerhalb des Planungszeitraums untersucht. Für jede Problem Instanz, jedes untersuchte Verfahren und jede dargestellte Parameterkonstellation wurde je ein Lösungslauf durchgeführt. Die Verfahren INS und NN sind deterministischer Natur, bei einer Wiederholung eines der Lösungsläufe wird man stets das gleiche Ergebnis erhalten. Der Ablauf von MACS und dynACO wird durch Zufallsvariablen gelenkt, daher besteht die Möglichkeit, daß in jedem Lösungslauf eine andere Lösung resultiert. Es wäre daher zweckmäßig, mehr als einen Versuch je gegebener Parameterausprägung durchzuführen. Da allerdings die benötigten Rechenzeiten sehr hoch sind, bleiben die Lösungsläufe auf je einen Lauf je Problem Instanz beschränkt.

Um zu untersuchen, ob dynACO nur zufällig besonders gute Lösungen findet, werden für die Problem Instanz A3 beispielhaft vier zusätzliche Lösungsläufe je dynACO-Parameterkonstellation durchgeführt. Die besten, schlechtesten und die durchschnittlichen Ergebnisse der dadurch insgesamt fünf Versuche sind in Tabelle 5.27 aufgeführt.

Die Ergebnisausprägungen der fünf zusätzlichen Lösungsläufe je Parameterkonstellation liegen relativ eng beieinander. Drei der fünf Ergebnisausprägungen von dynACO-CX7-5 first-fit und drei von fünf Ergebnisausprägungen von dynACO-CX7-10 first-fit werden von INS-CX7 first-fit bezüglich der Lösungsgüte übertroffen. Alle übrigen Parameterkonstellationen liefern in allen durchgeführten Lösungsläufen Lösungen mit einer Gesamtbediendauer, die unterhalb von 7733,89 ZE der $sched(X)$ des besten Referenzverfahrens INS-CX7 first-fit liegen.

Verfahren		$anz(X)$	$sched(X)$	Rechendauer (min)
dynACO-CX7-5	first-fit	5	min: 7699,57	89,55
			max: 7893,72	
			mittel: 7779,89	
dynACO-CX7-10	first-fit	5	min: 7664,09	38,06
			max: 7882,46	
			mittel: 7779,28	
dynACO-CX2-25	first-fit	5	min: 7641,24	56,85
			max: 7831,49	
			mittel: 7738,53	
dynACO-CX2-50	first-fit	5	min: 7583,18	117,06
			max: 7677,31	
			mittel: 7622,65	
dynACO-CX2-10	best-fit	5	min: 7544,36	105,87
			max: 7650,05	
			mittel: 7578,78	
dynACO-CX2-25	best-fit	5	min: 7519,61	79,49
			max: 7683,33	
			mittel: 7595,96	
dynACO-CX2-50	best-fit	5	min: 7601,79	69,42
			max: 7646,56	
			mittel: 7614,20	

Tabelle 5.27: Ergebnisse der zusätzlichen Lösungsläufe zu Probleminstanz A3 mit dynACO

Auch die relative Vorteilhaftigkeit der Verfahren untereinander variiert nicht wesentlich. dynACO-CX2-25 best-fit liefert die beste Lösung und wird im Mittel nur von dynACO-CX2-10 best-fit übertroffen. Tendentiell sind die Varianten mit best-fit Auswahlregel den Varianten mit first-fit Auswahlregel überlegen. Die Aussage, daß dynACO besonders gut geeignet ist, die Probleminstanz A3 zu lösen, kann nach diesen zusätzlichen Lösungsläufen bestätigt werden. Die Ergebnisqualität von dynACO ist zuverlässig gut.

5.5.2.2 Probleminstanz F3: Parametrische Analyse zu dynACO

Anhand der Probleminstanz F3 wird der Einfluss unterschiedlicher Parameter des neuen Ameisenalgorithmus dynACO näher untersucht. Die Probleminstanz F3 wird näher untersucht, da dynACO für diese Probleminstanz relativ schlechte Resultate ergeben hat. Es wird gezeigt, daß bei nur leicht veränderter Parametrisierung dynACO zur Lösung der Probleminstanz F3 gut geeignet ist. Da sich die Variante dynACO-CX2-25 best-fit für viele der untersuchten Probleminstanzen in Bezug auf

Lösungsgüte und Rechendauer bewährt hat, wird nur dynACO-CX2-25 best-fit näher untersucht.

Einfluss der Pheromonerhaltungsrate ρ auf dynACO: Zunächst wird die Pheromonerhaltungsrate ρ zwischen 0,45 und 0,9 variiert. Die Resultate sind in Tabelle 5.28 dargestellt.

Verfahren	ρ	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechen- dauer (min)
dynACO-CX2-25 best-fit	0,45	4	14539,31	4676,18	13047,83	16,35
dynACO-CX2-25 best-fit	0,50	2	8610,73	4264,33	12068,87	49,34
dynACO-CX2-25 best-fit	<i>0,55</i>	<i>2</i>	<i>8474,64</i>	<i>4371,46</i>	<i>13046,27</i>	<i>73,04</i>
dynACO-CX2-25 best-fit	<i>0,60</i>	<i>2</i>	<i>8404,57</i>	<i>4145,54</i>	<i>11851,23</i>	<i>54,46</i>
dynACO-CX2-25 best-fit (*)	0,65	2	8578,60	4418,14	12160,07	62,90
dynACO-CX2-25 best-fit	0,70	2	8554,80	4339,50	11666,90	28,32
dynACO-CX2-25 best-fit	0,75	2	8562,35	4242,10	11593,98	67,80
dynACO-CX2-25 best-fit	<i>0,80</i>	<i>2</i>	<i>8389,96</i>	<i>4354,41</i>	<i>12494,88</i>	<i>81,05</i>
dynACO-CX2-25 best-fit	0,85	2	8562,35	4242,10	11593,98	62,02
dynACO-CX2-25 best-fit	0,90	2	8562,35	4339,02	11784,97	155,03
NN-CX7 first-fit (*)		2	8486,62	4053,71	11735,17	280,75

Tabelle 5.28: dynACO - Einfluss der Pheromonerhaltungsrate ρ

(*) diese Ergebnisse stammen aus Tabelle 5.26

Die Ergebnisse, die mit (*) gekennzeichnet sind, stammen aus Tabelle 5.26 von Seite 146. Das Ergebnis von NN-CX7 first-fit wird zu Vergleichszwecken angegeben. Lösungen, die die vorläufig beste bekannte Lösung, die mit NN-CX7 first-fit ermittelt wurde, bezüglich der Lösungsgüte übertreffen, sind kursiv gedruckt. Wird bei gegebenen Parametern (vgl. Abschnitt 5.4) für dynACO eine Pheromonerhaltungsrate ρ von 0,45 verwendet, ergibt sich eine Gesamtbediendauer von 14539,31 ZE. Weiterhin werden 4 Touren anstelle von 2 Touren benötigt. Offenbar ist $\rho = 0,45$ für die Problem Instanz F3 zu niedrig gewählt. Auch mit $\rho = 0,50$ resultiert eine Lösung, die schlechter ist als die dynACO-Referenzlösung mit $\rho = 0,65$. Mit $\rho = 0,55$ wird sowohl die Referenzlösung mit $\rho = 0,65$ als auch die Lösung von NN-CX7 first-fit übertroffen. $\rho = 0,60$ liefert ein noch besseres Ergebnis. Mit Werten für $\rho = 0,70$ und $\rho = 0,75$ ist die Lösungsgüte weniger gut als bei $\rho = 0,60$. Bei $\rho = 0,80$ wird allerdings eine neue beste Lösung zur Problem Instanz F3, mit einer Gesamtbediendauer ($sched(X)$) von 8389,96 ZE gefunden. Die Rechendauer ist dabei mit 81,05 Minuten geringer als die des NN-CX7 first-fit Verfahrens. Die Gesamtbediendauer beträgt nur 97,8% der Gesamtbediendauer der Referenzlösung mit $\rho = 0,65$ und 98,9% der Gesamtbediendauer der Lösung, die mit NN-CX7 first-fit ermittelt wurde. Mit Werten

für die Pheromonerhaltungsrate $\rho \geq 0,85$ nimmt die Lösungsgüte bei wachsender Rechenzeit ab.

Einfluss des Gewichtungsfaktors π zwischen der lokalen Komponente η_{ij} und dem Pheromonwert ϕ_{ij} Der Gewichtungsfaktor π dient dazu, den Einfluss der lokalen Komponente im Verhältnis zum Pheromonwert bei der Konstruktion von neuen Lösungen durch die Ameise festzulegen. Je höher π gewählt wird, desto stärker beeinflusst die lokale Komponente die Wahl des nächsten zu beliefernden Kunden.

Den Untersuchungen in Abschnitt 5.5.1 wird $\pi = 1,4$ zugrundegelegt. In Tabelle 5.29 stehen die unterschiedlichen Ergebnisse, die sich ergeben, wenn ausgewählte Werte für π zwischen 1 und 3 gewählt werden.

Verfahren	π	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechen- dauer (min)
dynACO-CX2-25 best-fit	1,0	2	8554,94	4340,81	11666,90	35,24
dynACO-CX2-25 best-fit	1,1	2	8484,36	4333,54	11880,98	70,28
dynACO-CX2-25 best-fit	1,2	2	8554,80	4339,50	11666,90	41,16
dynACO-CX2-25 best-fit	1,3	2	8554,80	4331,37	11764,95	54,56
dynACO-CX2-25 best-fit (*)	1,4	2	8578,60	4418,14	12160,07	62,90
dynACO-CX2-25 best-fit	1,5	2	8493,24	4641,42	13668,78	83,66
dynACO-CX2-25 best-fit	1,6	2	8554,80	4339,50	11666,90	29,30
dynACO-CX2-25 best-fit	1,7	2	8399,38	4241,89	12394,50	49,91
dynACO-CX2-25 best-fit	1,8	2	8425,45	4157,44	11900,80	80,16
dynACO-CX2-25 best-fit	1,9	2	8691,02	4501,72	12363,83	25,97
dynACO-CX2-25 best-fit	2,0	2	8554,80	4505,39	12447,04	37,60
dynACO-CX2-25 best-fit	2,5	4	14539,31	4676,18	13047,83	12,50
dynACO-CX2-25 best-fit	3,0	4	14539,31	4676,18	13047,83	15,22
NN-CX7 first-fit (*)		2	8486,62	4053,71	11735,17	280,75

Tabelle 5.29: dynACO - Einfluss des Parameters π

(*) diese Ergebnisse stammen aus Tabelle 5.26

Offenbar ist dynACO relativ robust gegenüber Variationen des Parameters π im Bereich zwischen 1 und 2. Das beste Ergebnis mit $sched(X) = 8399,38$ ZE wird mit $\pi = 1,7$ erzielt. Auch wird $\pi = 1,1$, $\pi = 1,5$ und $\pi = 1,8$ wird die Lösungsgüte von NN-CX7 first-fit übertroffen. Mit $\pi \geq 1,9$ nimmt die Lösungsgüte merklich ab. Die relativ geringeren Rechenzeiten weisen darauf hin, daß dynACO dann frühzeitig konvergiert. Für $\pi = 2,5$ und $\pi = 3,0$ ist die Gewichtung der lokalen Komponente viel zu hoch, das Resultat entspricht der Lösung von INS ohne CX-Verbesserungsverfahren - der Startlösung des dynACO-Algorithmus. Die Ameisen sind nicht länger in der Lage, verbesserte Lösungen zu finden, wenn die lokale Komponente zu stark gewichtet wird. Da dann auch keine Lösung als vielversprechend angesehen wird - die Lösungen

erreichen nicht einmal das Niveau von INS ohne CX-Verbesserungsverfahren - wird kein CX-Verbesserungsverfahren durchgeführt.

Einfluss von unterschiedlichen Gewichtungsfaktoren λ der lokalen Komponente

Die lokale Komponente von dynACO setzt sich aus unterschiedlich gewichteten Bestandteilen zusammen.¹⁶ Der Gewichtungsfaktor λ_1 gewichtet die Entfernung zwischen i und j . λ_2 gewichtet die Fahrzeit, die von i zu j bei Abreise bei i zum Zeitpunkt τ anfällt. λ_3 gewichtet die erwartete Wartezeit vor Beginn der Bedienung von Kunde j und λ_4 gewichtet den zeitlichen Abstand bis zum Ende des Belieferungsintervalls von j . Den Untersuchungen in Abschnitt 5.5.1 wird $\lambda = (0, 10, 15, 5)$ zugrundegelegt, da diese Parameterausprägungen in Vorabuntersuchungen besonders gute Resultate geliefert haben. In Tabelle 5.30 finden sich die Ergebnisse, die mit unterschiedlichen, alternativen Ausprägungen für λ ermittelt wurden.

Verfahren	λ	$anz(X)$	$sched(X)$	$travel(X)$	$dist(X)$	Rechen- dauer (min)
dynACO-CX2-25 best-fit	$(0,0,0,10)$	2	8421,53	3997,36	12017,70	78,44
dynACO-CX2-25 best-fit	$(0,0,10,10)$	2	8496,34	4166,03	12151,85	90,03
dynACO-CX2-25 best-fit	$(0,10,0,0)$	2	8555,71	4445,38	12259,98	35,45
dynACO-CX2-25 best-fit	$(0,10,0,10)$	2	8727,77	4539,70	12295,43	80,39
dynACO-CX2-25 best-fit	$(0,10,10,0)$	2	8554,80	4339,50	11666,90	28,31
dynACO-CX2-25 best-fit	$(0,10,10,10)$	2	8554,80	4339,50	11666,90	28,56
dynACO-CX2-25 best-fit	$(10,0,0,0)$	2	8554,80	4339,50	11666,90	27,88
dynACO-CX2-25 best-fit	$(10,0,0,10)$	2	8554,80	4339,50	11666,90	30,28
dynACO-CX2-25 best-fit	$(10,0,10,0)$	2	8554,80	4505,39	12447,04	33,86
dynACO-CX2-25 best-fit	$(10,0,10,10)$	2	8419,41	4603,88	13293,55	77,14
dynACO-CX2-25 best-fit	$(10,10,0,0)$	2	8554,80	4339,50	11666,90	27,94
dynACO-CX2-25 best-fit	$(10,10,0,10)$	2	8578,60	4439,37	12260,70	28,77
dynACO-CX2-25 best-fit	$(10,10,10,0)$	2	8554,80	4339,50	11666,90	28,18
dynACO-CX2-25 best-fit	$(10,10,10,10)$	2	8404,57	4145,54	11851,23	38,57
dynACO-CX2-25 best-fit (*)	$(0,10,15,5)$	2	8578,60	4418,14	12160,07	62,90
NN-CX7 first-fit (*)		2	8486,62	4053,71	11735,17	280,75

Tabelle 5.30: dynACO - Einfluss von unterschiedlichen Gewichtungsfaktoren λ

(*) diese Ergebnisse stammen aus Tabelle 5.26

Wenn $\lambda = (0, 0, 0, 0)$ oder $\lambda = (0, 0, 10, 0)$ gesetzt wird, kann der Ameisenalgorithmus dynACO keine Lösungen ermitteln. λ_3 gewichtet die erwartete Wartezeit bei dem Kunden j ($\max\{e_j - \tau - t_{ij}(\tau); 0\}$).¹⁷ Wenn die Wartezeit beim Kunden j null beträgt, weil das Lieferfahrzeug zu einem Zeitpunkt nach e_j beim Kunden j

¹⁶vgl. Beschreibung in Abschnitt 4.5.3.1 auf Seite 93

¹⁷vgl. zur Berechnung der lokalen Komponente (4.1) auf Seite 93

ankommt, beträgt der komplette Term der lokalen Komponente unter dem Nenner null. Da $\frac{1}{0}$ nicht definiert ist, ist auch η_{ij} nicht definiert und es lassen sich keine Auswahlwahrscheinlichkeiten für die konstruierende Ameise ermitteln.

Unabhängig davon, welche Parameterkombination für λ gewählt wird, werden Lösungen mit 2 Touren ermittelt. Die benötigte Rechenzeit beträgt im Mittel 45,27 Minuten. Die beste Lösung wird nach 38,57 Minuten mit $\lambda = (10, 10, 10, 10)$ ermittelt, die Gesamtbediendauer beträgt 8404,57 ZE, sie übertrifft die Lösung, die mit NN-CX7 first-fit ermittelt wurde. Mit Parametervariationen von π und ρ wurden größere Verbesserungen erzielt (vgl. $\rho = 0,7$ in Tabelle 5.28 und $\pi = 1,7$ in Tabelle 5.29). Die besten drei Lösungen, die NN-CX7 first-fit bezüglich der Rechendauer und der Gesamtbelieferungszeit übertreffen, weisen nur eine einzige Gemeinsamkeit auf: Die Ausprägung des Parameters $\lambda_4 = 10$.

Interessant ist zu beobachten, daß relativ häufig (6 Ergebnisse von 14 Versuchen) mit unterschiedlichsten Parameterausprägungen λ nach etwa einer halben Stunde Rechendauer das gleiche Ergebnis ermittelt wird, mit einer Gesamtbediendauer von 8554,80 ZE, einer Gesamtfahrzeit von 4339,50 ZE und einer Gesamtentfernung von 11666,90 EE. Ein besonders schlechtes Ergebnis wurde mit $\lambda = (0, 10, 0, 10)$ erzielt, $sched(X)$ beträgt dann 8727,77 ZE.

Zusammenfassend läßt sich feststellen, daß sich die Lösungsgüte von dynACO nicht monoton sondern vielmehr sprunghaft mit der Änderung der Parameter ändert. Insbesondere die Pheromonerhaltungsrate ist sorgfältig experimentell zu wählen. Eine vorteilhafte Ausprägung für eine bestimmte Problem Instanz ist dabei nicht notwendigerweise auch vorteilhaft für eine andere Problem Instanz. Für praktische Einsatzzwecke wären Experimente mit einer größeren Stichprobe strukturell ähnlicher Problemstellungen erforderlich, um geeignete Parameter zu wählen. Dies geht allerdings über die Zielsetzung dieser Arbeit hinaus.

5.5.3 Zusammenfassung der Ergebnisse der experimentellen Analyse von dynamisch-deterministischen Tourenplanungsproblemen (dTP)

Rechenzeiten:

Anhand von 18 Problem Instanzen wurden vier unterschiedliche Heuristiken zur Lösung von dynamisch-deterministischen Tourenplanungsproblemen untersucht. Es ist erkennbar, daß die einfachen Heuristiken INS und NN innerhalb von sehr kurzer Zeit zulässige Lösungen zum dTP liefern. Wird ein CX-Verbesserungsverfahren ergänzend eingesetzt, benötigt NN-CX viele Iterationen um CX-optimale Lösungen zu ermitteln.

INS-CX findet relativ schnell CX-optimale Lösungen, da die Ausgangslösung bereits besser ist.

Interessant ist zu beobachten, daß die Probleminstanzen D erheblich größere Rechenzeiten aufweisen als A, B und C. Dies kann darauf zurückgeführt werden, daß bei einem Anteil von 33% Kundenaufträgen ohne Zeitfensterrestriktionen die CX-Nachbarschaftssuche deutlich mehr Rechenzeit benötigt, da eine größere Zahl von zulässigen Belieferungsreihenfolgen existiert und die Zulässigkeitsbedingungen aus Abschnitt (4.1) nur eine geringere Anzahl an unzulässigen Touren aussortieren können.

Vergleich unterschiedlicher Verkehrswegenetzwerke Um mögliche Unterschiede bezüglich der Lösungsgüte der Algorithmen in Abhängigkeit vom Detaillierungsgrad des Verkehrswegenetzwerks zu untersuchen, wurde den Probleminstanzen A2, B2, C2 und D2 (E2 und F2) das Verkehrswegenetzwerk 50b (65b) zugrundegelegt, welches dem Verkehrswegenetzwerk 50a (65a) der Probleminstanzen A1, B1, C1 und D1 (E1, F1), durch zusätzliche Pfeile ergänzt, entspricht. Alle gemeinsamen Pfeile besitzen die gleichen Bewertungen, die zusätzlichen Pfeile erhalten niedrige Geschwindigkeiten zugeordnet. Sie sind nur dann vorteilhaft zu wählen, wenn die Wegstreckenersparnis besonders groß ist.

Es wäre zu erwarten gewesen, daß bei mehr alternativen Routen, die zur Verfügung stehen, die Rechenzeiten ansteigen, weil mehr Wege zu untersuchen wären. Tatsächlich liegt im Durchschnitt die Rechendauer bei A1, B1, C1, D1 und F1 oberhalb der durchschnittlichen Rechendauer von A2, B2, C2, D2 bzw. F2. Eine Erklärung dafür bietet der modifizierte Dijkstra-Algorithmus zur Bestimmung von schnellsten Wege im Verkehrswegenetzwerk (vgl. Abschnitt 2.2.2.4). Gerade bei geographisch nahe beieinanderliegenden Knoten, existieren nicht viele vorteilhafte alternative Wege. Daher werden auch nicht viele zusätzliche Pfeile des dichteren Netzwerks zu überprüfen sein. Die zusätzlichen Pfeile, die es erlauben einen Zielknoten direkt zu erreichen, beschleunigen teilweise sogar die Suche nach dem schnellsten Weg im Verkehrswegenetzwerk, da gegebenenfalls weniger Pfeile zu überprüfen sind.

Ein Vergleich der Rechendauern von Probleminstanzen mit gleichen Netzwerkstrukturen 50b bzw. 65b, allerdings unterschiedlichen Bewertungen der Pfeile (A2, B2, C2, D2, F2 mit A3, B3, C3, D3, F3) zeigt, daß mit mehr Intervallen für die Geschwindigkeitsfunktionen auch die Bestimmung der schnellsten Wege aufwendiger wird. Dies ist darauf zurückzuführen, daß mit zunehmender Anzahl an Intervallen für die Geschwindigkeitsfunktion die Ermittlung der benötigten Fahrzeit $t_{ij}(\tau_i)$ zwischen zwei Knoten des Verkehrswegenetzwerks aufwendiger wird.¹⁸

¹⁸Für die Probleminstanzen E2 und E3 gilt dies auf den ersten Blick nicht. Da in die mittlere Rechendauer von E2 auch der Ausreißerwert von MACS-CX2 eingeht, ist dieser Vergleich allerdings

Einfache Heuristiken NN und INS Erwartungsgemäß sind die einfachen Heuristiken NN und INS besonders schnell bei der Ermittlung von zulässigen Lösungen zum dTP. Dabei liefert NN etwas schneller Lösungen als INS, allerdings lässt die Lösungsgüte deutlich zu wünschen übrig. Wird in kurzer Zeit eine zulässige Lösung benötigt, ist INS eine gute Wahl. Es werden schnell zulässige Ausgangslösungen erzeugt, deren Lösungsgüte in der Regel besser als die von NN ist.

Durch die relativ guten Ausgangslösungen benötigt auch die CX-Nachbarschaftssuche auf Basis von INS nicht so viel Rechenzeit um CX-optimale Lösungen zu ermitteln wie auf Basis einer Ausgangslösung, die mit NN ermittelt wurde. Ohne den Einsatz der CX-Nachbarschaftssuche ist die Lösungsgüte des NN Verfahrens sehr schlecht. Zum Teil werden mehr als doppelt so viele Touren benötigt wie mit CX, und dabei Gesamtbediendauern erreicht, die deutlich über den besten ermittelten $sched(X)$ liegen. Das ist darauf zurückzuführen, daß die Ausgangslösung vor der lokalen Suche bei NN von erheblich schlechterer Qualität ist und somit eine erheblich größere Zahl CX-Iterationen benötigt wird, bis die Lösung CX-optimal ist. Bei deutlich höheren Rechenzeiten liefert NN-CX häufiger als INS-CX (13 von 18 untersuchte Probleminstanzen) die besseren Ergebnisse.

Ameisenkolonialgorithmus MACS Die Lösungsqualität des Ameisenkolonialgorithmus von Donati et al. (2003a) (MACS) liegt für die hier untersuchten dynamisch-deterministischen Tourenplanungsprobleme etwa auf dem Niveau einfacher Heuristiken mit CX-Nachbarschaftssuche, wobei gleichzeitig ein höherer Rechenzeitbedarf besteht. Nur für die Probleminstanzen C3, D2 und D3 übertrifft MACS-CX7 die untersuchten einfachen Heuristiken INS und NN, ergänzt durch CX-Verbesserungsverfahren.

MACS tendiert mit den von Donati et al. (2003a) vorgeschlagenen Parametern dazu in schlechten lokalen Minima zu konvergieren. Die Kolonie ACS-VEI zur Suche von zulässigen Lösungen mit möglichst wenigen Touren ist überflüssig, denn auch mit Hilfe einer CX-Nachbarschaftssuche können Lösungen mit der gleichen Tourenzahl erzielt werden. In keinem Fall erhält man mit MACS die besten bekannten Lösungen.

Sowohl MACS als auch NN-CX gehen von der gleichen Lösung aus (NN ohne CX-Verbesserungsverfahren). Daher kann verglichen werden, ob die Suchstrategie von Donati et al. (2003a) oder ein reines CX-Verbesserungsverfahren besser geeignet sind, gute Lösungen in akzeptabler Rechenzeit zu dTP zu ermitteln. Die Rechenzeiten von MACS sind bei allen Probleminstanzen ausserordentlich hoch. Häufig sind gleichzeitig die Lösungen, die mit NN-CX7 ermittelt werden, besser als die Lösungen von

problematisch. Wird die Rechenzeit der übrigen Verfahren verglichen, gilt auch für E2 und E3, daß mit steigender Dynamik die Rechenzeiten steigen.

MACS-CX7 (jeweils first-fit).

dynACO Die Rechenzeiten des neu entwickelten Ameisenkolonialgorithmus dynACO sind erwartungsgemäß höher als die Rechenzeiten der einfachen Heuristiken NN und INS, sie liegen allerdings meistens unterhalb der Rechendauern, die MACS benötigt. Dafür resultieren Lösungen, die durch eine einfache CX-Nachbarschaftssuche nicht erreicht werden.

Dabei hat der neu vorgestellte Lösungsansatz dynACO mit einer fest vorgegebenen Parameterauswahl für 16 von 18 Probleminstanzen die besten Lösungen ermittelt. Anhand der Probleminstanz F3 wurde außerdem gezeigt, daß mit einer geeigneten Anpassungen der Parameterausprägungen dynACO wiederum in der Lage ist, die beste Lösung unter allen untersuchten Verfahren zu ermitteln.

Gerade für die Probleminstanzen, die relativ schwierig zu lösen sind, hat dynACO hervorragende Ergebnisse geliefert. Die best-fit Suchstrategie für das CX-Nachbarschaftssuchverfahren ist dabei besonders vorteilhaft. Für die Probleminstanzen mit weniger zahlreichen, längeren Touren ist es vorteilhaft längere Abschnitte bei der CX-Nachbarschaftssuche miteinander zu tauschen. Häufig sind 25 Iterationen CX je Ameise ausreichend, mit $CX_{iter}^{max} = 50$ werden nur vereinzelt bessere Tourenpläne ermittelt. Die Variante von dynACO mit den Standardparametern (vgl. Abschnitt 5.4), die eine Abschnittslänge von maximal 2 Knoten nachdem best-fit Prinzip mit $CX_{iter}^{max} = 25$ vertauscht, liefert für 7 von 18 Probleminstanzen die beste bekannte Lösung, für 16 von 18 Probleminstanzen wird einer der besten drei Tourenpläne gefunden.

Die Verfahren INS-CX und dynACO beginnen von der gleichen Lösung ausgehend verbesserte Lösungen zu dTP zu suchen. Während INS-CX häufig schnell (nach kurzer Rechendauer) in einem lokalen Optimum endet und bezüglich der Bediendauer oft von NN-CX übertroffen wird, verlässt dynACO diese lokalen Optima und durchsucht größere Teile des Lösungsraums.

dynACO liefert regelmäßig bei geringerer Rechendauer bessere Lösungen als MACS. Die geringeren Rechenzeiten sind zum einen darauf zurückzuführen, daß MACS eine Ausgangslösung auf Basis von NN ermittelt und dann verbessert, während dynACO durch die Lösung des INS Verfahrens initialisiert wird. Gleichzeitig benötigt dynACO weniger Ameisen-Iterationen als MACS, da bei dynACO in jeder Iteration ein zulässiger Tourenplan ermittelt wird. Die einzelnen Ameisen von dynACO führen außerdem nicht beliebig viele Iterationen der lokalen CX-Suche durch, sondern beenden die Suche nach CX_{max}^{iter} Verbesserungsschritten, um dann auf Basis der gesammelten Erfahrung (Pheromon) neue Lösungen zu konstruieren.

Die Aussage, daß höhere $CX_{\text{abschnitt}}^{\text{max}}$ immer zu besseren Lösungen führen, lässt sich für dynACO nicht bestätigen. Eindeutig lässt sich sagen, daß 5 Iterationen CX je Ameisen-Iteration zu wenig sind. Die Abschnittslänge für die CX-Nachbarschaftssuche hat überraschenderweise keinen besonders großen Einfluss auf die Rechendauer. Die mittleren Rechendauern unterscheiden sich zwischen dynACO-CX2-25 best-fit (186 Minuten) und dynACO-CX2-50 best-fit (181 Minuten) kaum. dynACO-CX7-10 first-fit liefert häufig schneller Tourenpläne (102 Minuten), allerdings ist diese Variante von dynACO nur bei den Probleminstanzen mit hohen Fahrzeugkapazitäten (B1, B2 und F2) vorteilhaft. Bei gleichen $CX_{\text{abschnitt}}^{\text{max}}$ und $CX_{\text{iter}}^{\text{max}}$ ist die best-fit Auswahlregel zwar durchschnittlich langsamer, aber von besserer Lösungsgüte als die first-fit Auswahlregel.

5.6 Experimentelle Analyse von dynamisch-stochastischen Tourensteuerungsproblemen

5.6.1 Probleminstanzen zur Untersuchung von dynamisch-stochastischen Tourensteuerungsproblemen (dTS)

Von der Tourenplanung wird in dem Moment zu Tourensteuerung übergegangen, in dem das Depot verlassen wird, um mit der Belieferung der Kunden zu beginnen. Im folgenden wird die Vorgehensweise zur Erzeugung von Probleminstanzen zur Untersuchung von dynamisch-stochastischen Tourensteuerungsproblemen (dTS) beschrieben.

In Abschnitt 5.5.1 werden die Ergebnisse der experimentellen Untersuchungen zu dTP dargestellt. Für jede Probleminstanz zu dTP wurden 17 Tourenpläne X ermittelt, die Kunden zu Fahrzeugen zuordnen, eine vorläufige Belieferungsreihenfolge und die dazugehörigen Routen festlegen. Daraus ergeben sich $17 \cdot 18 = 306$ Tourenpläne. Darunter befinden sich 13 identische Tourenpläne, die mit unterschiedlichen Algorithmen oder Parameterausprägungen eines Algorithmus ermittelt wurden. Auch bei einem Vergleich von unterschiedlichen Tourenplänen sind Wiederholungen von einzelnen Touren nicht auszuschließen. Im folgenden wird jede einzelne Tour der 306 Tourenpläne durch Störpläne ergänzt. Sie bilden die im folgenden untersuchten Probleminstanzen zu dTS. Touren, die in mehreren Tourenplänen enthalten sind, werden im folgenden mehrfach berücksichtigt, da sie mehrfach Bestandteil von vorteilhaften Lösungen sind.

Störplan I-50 und I-65

Für die Probleminstanzen A, B, C und D auf Basis des Verkehrswegenetzwerks 50b wird der Störplan $I - 50$ mit 1000 Störereignissen σ generiert. Für die Probleminstanzen E und F wird auf Basis des Verkehrswegenetzwerks 65b der Störplan $I - 65$ mit 1200 Störereignissen generiert.

Den Probleminstanzen zu dTS, die aus den in Abschnitt (5.5.1) dargestellten Lösungen zu A1 bis D1, A2 bis D2 und A3 bis D3 hervorgegangen sind, wird der Störplan $I - 50$ zugeordnet, den Probleminstanzen E1 bis E3 und F1 bis F3 der Störplan $I - 65$. Alle Störungen, die Pfeile betreffen, die nur in dem Verkehrswegenetzwerk 50b (bzw. 65b) und nicht im Verkehrswegenetzwerk 50a (bzw. 65a) existieren, werden für die Probleminstanzen A1-F1 als nicht entscheidungsrelevant verworfen. Aus dem Störplan $I - 50$ verbleiben von 1000 Störereignissen dadurch 753 entscheidungsrelevante Störereignisse.

Jedes Störereignis σ ist durch einen Multiplikator Δ_σ gekennzeichnet. Die Ausprägungen von Δ_σ werden zufällig gleichverteilt aus dem Intervall $[0, 1; 0, 6]$ gezogen. Die Dauer der Wirkung eines Störereignisses ($l_\sigma - e_\sigma$) beträgt zwischen 5% und 15% des Planungszeitraums, die Ausprägungen der Dauer von Störereignissen werden ebenfalls unabhängig voneinander, zufällig gleichverteilt innerhalb dieser Dauer festgelegt.

Von den insgesamt 796 Pfeilen des Verkehrswegenetzwerks 50b sind zu jedem Zeitpunkt im Planungshorizont durchschnittlich etwa 10% oder 80 Pfeile durch ein Störereignis betroffen. Die Dauer einer Störung beträgt durchschnittlich etwa 8% des Planungszeitraums, da Störereignisse, die zu Beginn und zum Ende des Planungshorizonts liegen, durch e_0 oder l_0 begrenzt werden. Es ergibt sich $8\% [mittlereDauer] \cdot 1000 [AnzahlStoerereignisse] \approx 80 [Pfeile]$. Analog gilt für das Verkehrswegenetzwerk 50a folgende Überlegung: Von den 1000 Störereignissen sind 753 Störereignisse entscheidungsrelevant, die übrigen Störereignisse betreffen Verbindungen, die nicht Teil des Verkehrswegenetzwerks 50a sind. Es ergeben sich $8\% [mittlereDauer] \cdot 753 [AnzahlStoerereignisse] \approx 60 [Pfeile]$, was wiederum zu jedem Zeitpunkt des Planungshorizonts etwa 10% der insgesamt 596 Pfeile entspricht.

Für die Verkehrswegenetzwerke 65a und 65b gelten analoge Betrachtungen, zu jedem Zeitpunkt des Planungszeitraums sind etwa 10% aller Verbindungen durch ein Störereignis betroffen.

Unter Vorwarnzeit wird im folgenden die Zeitdauer verstanden, die ein Störereignis vor Eintritt seiner Wirkung bekannt wird ($b_\sigma - e_\sigma$). Die Vorwarnzeit beträgt für 10% der Störereignisse zwischen 5% und 10% des Planungszeitraums, für 90% der Störereignisse stimmen b_σ und e_σ überein.

Störplan II-50 und II-65

Der Störplan $II - 50$ ($II - 65$) unterscheidet sich von dem Störplan $I - 50$ ($I - 65$) dadurch, daß die Ausprägungen des Multiplikators Δ_σ für 20% der Störereignisse zufällig gleichverteilt aus dem Intervall $[1, 05; 1, 10]$ gezogen wird. 20% der Störereignisse führen zu einer Erhöhung der Ist-Geschwindigkeit um 5 – 10% im Vergleich zur Plan-Geschwindigkeit. Für 80% der Störereignisse wird Δ_σ wie bei Störplan $I - 50$ und $I - 65$ zufällig gleichverteilt aus dem Intervall $[0, 1; 0, 6]$ gezogen. Alle übrigen Eigenschaften der Störpläne I und II stimmen überein.

Störplan III-50 und III-65

Der Störplan $III - 50$ ($III - 65$) ist im Vergleich zu $I - 50$ ($I - 65$) durch weniger zahlreiche, länger anhaltende Störereignisse gekennzeichnet. Für die Probleminstanzen A, B, C und D enthält der Störplan $III - 50$ 500 Störereignisse, für die Probleminstanzen E und F enthält der Störplan $III - 65$ 600 Störereignisse. Die Ausprägungen der Dauer der Wirkung eines Störereignisses ($l_\sigma - e_\sigma$) wird zufällig gleichverteilt zwischen 10% und 30% des Planungszeitraums festgelegt. Alle übrigen Eigenschaften der Störpläne I und III stimmen überein.

Von den insgesamt 596 Pfeilen des Verkehrswegenetzwerks 50a sind zu jedem Zeitpunkt im Planungshorizont durchschnittlich wiederum etwa 10% oder 60 Pfeile durch ein Störereignis betroffen. Die Dauer einer Störung beträgt durchschnittlich etwa 16% des Planungszeitraums, es sind 376 der 500 Störereignissen für das Verkehrswegenetzwerk 50a entscheidungsrelevant. Es ergeben sich etwa $16\% [\text{mittlere Dauer}] \cdot 376 [\text{Anzahl Störereignisse}] \approx 60 [\text{Pfeile}]$, die zu jedem Zeitpunkt durch eine Störung betroffen sind. Analoge Überlegungen gelten für die Verkehrswegenetzwerke 50b, 65a und 65b, es sind ca. 10% aller Pfeile zu jedem Zeitpunkt durch ein Störereignis betroffen.

Störplan IV-50 und IV-65

Der Störplan $IV - 50$ ($IV - 65$) unterscheidet sich von Störplan $I - 50$ ($I - 65$) durch den höheren Anteil an Störereignissen mit Vorwarnzeit. Für 50% der Störereignisse (statt 10%) beträgt die Vorwarnzeit, wie bei $I - 50$ und $I - 65$, zwischen 5% und 10% des Planungszeitraums. Alle übrigen Eigenschaften der Störpläne I und IV stimmen überein.

Störplan V-50 und V-65

Im Unterschied zu Störplan $I - 50$ ($I - 65$) wird bei Störplan $V - 50$ ($V - 65$) der Multiplikator Δ_σ des Störereignisses σ zufällig gleichverteilt aus dem Intervall $[0, 6; 0, 9]$ gezogen. Die Störungen fallen weniger stark aus, somit wird der Verkehr weniger beeinträchtigt. Alle übrigen Eigenschaften der Störpläne I und V stimmen überein.

Störplan VI-50 und VI-65

Der Störplan $VI - 50$ ($VI - 65$) ist im Vergleich zu $I - 50$ ($I - 65$) durch eine höhere Anzahl von Störereignissen gekennzeichnet. Der Störplan $VI - 50$ für die Probleminstanzen A, B, C und D enthält 2000 Störereignisse, der Störplan $VI - 65$ für die Probleminstanzen E und F enthält 2400 Störereignisse. Zu jedem Zeitpunkt sind etwa 20% der Pfeile durch ein Störereignis betroffen. Die Eigenschaften der Störereignisse aus den Störplänen I und VI stimmen überein.

5.6.2 Ergebnisse und Interpretation der experimentellen Analyse zu dTS

5.6.2.1 Übersicht Störpläne

Störplan	<i>I</i>		<i>II</i>		<i>III</i>		<i>IV</i>		<i>V</i>		<i>VI</i>	
	50	65	50	65	50	65	50	65	50	65	50	65
Anzahl σ	1000	1200	1000	1200	500	600	1000	1200	1000	1200	2000	2400
Δ_σ	[0,1;0,6]		20% zwischen [1,05;1,10] 80% zwischen [0,1;0,6]		[0,1;0,6]		[0,1;0,6]		[0,6;0,9]		[0,1;0,6]	
$(l_\sigma - e_\sigma)$	5-15%		5-15%		10-30%		5-15%		5-15%		5-15%	
Anteil mit $(b_\sigma < e_\sigma)$	10%		10%		10%		50%		10%		10%	
$(e_\sigma - b_\sigma)$	5-10%		5-10%		5-10%		5-10%		5-10%		5-10%	

Table 5.31: Übersicht Störungspläne I bis VI

In Tabelle 5.31 sind noch einmal die wesentlichen Charakteristika der Störpläne I bis VI zusammenfassend dargestellt. Es ist zu beachten, daß die Störpläne zwar wesentliche gemeinsame Charakteristika aufweisen, aber die einzelnen Ausprägungen b_σ , e_σ , l_σ und Δ_σ der Störereignisse für jeden Störplan unabhängig generiert wurden.

Für jede der Probleminstanzen zu dynamisch-stochastischen Tourensteuerungsproblemen werden die drei Störungsreaktionsverfahren ROUTE, PT und EX unabhängig voneinander untersucht und der Handlungsalternative ALT (keine Reaktion) gegenübergestellt. Ein konkreter Tourenplan X gibt die Plan-Größen für die Anzahl an Touren, die Gesamtbediendauer und die Gesamtfahrzeit an und wird im folgenden als Plan-Tourenplan bezeichnet. Die Touren T und die Routen R aus dem Plan-Tourenplan X geben die vorläufige Belieferungsreihenfolge und eine vorläufige Fahrtroute vor.

Mit Verlassen des Depots treten sukzessive Störereignisse ein, die dem Störplan S zu entnehmen sind. Der Planungszeitraum von e_0 bis l_0 wird in 50 Intervalle unterteilt, zu Beginn jedes Intervalls (τ_{reakt}) werden alle bis dahin bekannten Störereignisse σ (mit $\sigma \in \{S | b_\sigma \leq \tau_{reakt}\}$) berücksichtigt und für jede untersuchte Tour der als nächstes erreichte Knoten m und der erwartete Ankunftszeitpunkt τ_m bestimmt. Anschließend wird das betrachtete Störungsreaktionsverfahren durchgeführt. Dabei werden gegebenenfalls die vorläufigen Routen R und Touren T modifiziert.

Nachdem eine Tour mit der Rückkehr ins Depot beendet wird, kann die Ist-Verspätung, die Ist-Fahrzeit und die Ist-Bediendauer ermittelt werden. Für alle zurückgelegten und ggf. modifizierten Touren eines Plan-Tourenplans werden die Teillösungen zu einem Ist-Tourenplan zusammengefasst. Die Ist-Gesamtverspätung eines Ist-Tourenplans entspricht der Summe der tatsächlich eingetretenen Verspätungen bei allen zu beliefernden Kunden i ($\sum_{i \in C_0} \max\{(y_i - l_i); 0\}$ über alle Touren des Tourenplans), analog entspricht die Ist-Gesamtbediendauer der Summe der tatsächlich benötigten Bediendauern und die Ist-Gesamtfahrzeit der Summe der tatsächlich benötigten Fahrzeiten der zum Ist-Tourenplan zugehörigen Touren zur Belieferung aller Kunden.

Die Anzahl an Touren ist bei dTS nicht entscheidungsrelevant, denn sie stimmt immer mit der Planung überein. Da die Ausprägungen von Ist-Verspätungen, Ist-Bediendauern und Ist-Fahrzeiten von der Qualität der Ausgangslösung abhängen, hier aber vor allem die Eignung von Störungsreaktionsverfahren untersucht werden soll, wird über die resultierenden 17 Ist-Tourenpläne je dTP-Probleminstanz der Mittelwert über die Gesamtbediendauer, über die Gesamtverspätung und über die Anzahl der verspätet belieferten Kunden ermittelt und dem Mittelwert über die Bediendauer der 17 Plan-Tourenpläne $sched(\bar{X})$ gegenübergestellt. Für 18 Probleminstanzen zu dTP, kombiniert mit 6 Störplänen und 4 Störungsreaktionsverfahren ergeben sich insgesamt 432 Ergebnisse.

5.6.2.2 Detailliergegebnisse der experimentellen Analyse

In Tabelle 5.32 werden die Ergebnisse der einzelnen Ist-Tourenpläne, die sich beim Einsatz der Handlungsalternative ALT (keinerlei Anpassung von Routen oder Touren) ergeben, wenn die Plan-Tourenpläne zur Probleminstanz A1 (vgl. Tabelle 5.9), ergänzt durch den Störplan $I - 50$ zugrundegelegt werden.

Ausgangslösung	Ist-Belieferungsdauer	Ist-Verspätung	Anzahl verspätet belieferteter Kunden	Plan-Belieferungsdauer
dynACO-CX2-25 best-fit	8655,58	997,06	8	7888,41
dynACO-CX7-10 first-fit	8742,92	721,93	5	7953,90
dynACO-CX2-50 best-fit	9007,58	1568,24	12	7975,81
NN-CX2 best-fit	9250,90	2611,10	15	8021,23
MACS-CX7	8923,81	2114,57	11	8031,24
dynACO-CX2-10 best-fit	9002,70	1209,01	9	8048,50
dynACO-CX2-25 first-fit	9394,24	2731,93	15	8069,45
INS-CX2 first-fit	9277,52	2856,51	16	8080,84
INS-CX2 best-fit	8804,58	693,34	7	8083,33
dynACO-CX7-5 first-fit	8713,11	433,58	5	8091,93
dynACO-CX2-50 first-fit	9036,36	1248,17	9	8129,35
NN-CX2 first-fit	9434,34	2639,50	15	8190,77
MACS-CX2	9099,01	992,33	9	8194,94
INS-CX7 first-fit	9172,75	946,15	7	8340,38
NN-CX7 first-fit	9709,29	2032,80	13	8745,03
INS	10602,25	761,27	7	9762,27
NN	16009,37	2096,93	12	15012,67
Durchschnitt	9578,61	1567,91	10,29	8623,29

Tabelle 5.32: Einzelergebnisse dTS A1, Störplan I, Störungsreaktionsverfahren ALT

Da die Darstellung jeder einzelnen Lösung zu aufwendig und sehr platzraubend ist, wird im folgenden nicht jedes einzelne Ergebnis aufgeführt. Für jede dTP-Probleminstanz, auf deren Ergebnissen die dynamisch-stochastischen Tourensteuerungsprobleme (dTS) basieren und jeden ergänzenden Störplan werden im folgenden die Mittelwerte bezüglich der Verspätung, bezüglich der Anzahl der verspätet belieferten Kunden, bezüglich der Ist-Bediendauer und bezüglich der Plan-Bediendauer, nach Störungsreaktionsverfahren differenziert, ermittelt. Die derart zusammengefassten Ergebnisse der experimentellen Analyse können den Tabellen 5.33 bis 5.41 entnommen werden. Damit auf den ersten Blick die relative Vorteilhaftigkeit der Störungsreaktionsverfahren ersichtlich ist, sind die Ergebnisse je dTP-Probleminstanz und Störplan nicht-absteigend nach der mittleren Verspätung sortiert dargestellt.¹⁹

¹⁹vgl. die lexikographische Zielfunktion (3.1) zu dTS auf Seite 49.

So stehen beispielsweise in der Zeile aus Tabelle 5.33, die mit “Problem A1-I, Verfahren ALT” beschrieben ist, die Mittelwerte über alle Ist-Tourenpläne X^{Ist} , die auf den Tourenplänen zur Probleminstanz A1 basieren und die bei denen während der Auslieferung die Handlungsalternative ALT verwendet wurde.

Die Minimierung der mittleren Verspätung ist die primäre Zielsetzung zum dTS (vgl. Abschnitt 3.2.1). In 48 von 108 Fällen liefert das Störungsreaktionsverfahren EX (Tausch der Position von zwei beliebigen Kunden auf einer Tour) im Durchschnitt die beste Lösung, in 101 von 108 Fällen ist EX eines der beiden besten Verfahren. In 52 von 108 Fällen liefert PT (Tausch der Position von zwei direkt nacheinander eingeplanten Kunden auf einer Tour) die beste Lösung (in 98 von 108 Fällen unter den beiden Besten). In 8 Fällen ist es besonders vorteilhaft, nur die Routen anzupassen (ROUTE) und auf eine Modifikation der Belieferungsreihenfolge zu verzichten. Es zeigt sich, daß ALT (keinerlei Änderung der Routen oder Touren) in keinem Fall die beste und auch nur in einem einzigen Fall die zweitbeste Handlungsalternative ist. Treten stochastische Störungen der Geschwindigkeiten während der Auslieferung auf, ist es in jedem untersuchten Fall vorteilhaft von der Planung der Routen abzuweichen und auf Störungen zu reagieren.

Es ist bei der Interpretation der Tabellen zu beachten, daß in Einzelfällen eine mittlere Ist-Bediendauer erreicht wird, die geringer als die Plan-Bediendauer ausfällt. Die Ursache dafür liegt darin begründet, daß die Tourenpläne, die mit NN oder INS ohne CX-Verbesserungsverfahren ermittelt wurden, durch PT oder EX verbessert werden können.

5.6.2.3 Ausgewählte Detailergebnisse der experimentellen Analyse

Probleminstanzen A1-I bis A1-VI: Die durchschnittlichen Ergebnisse der Ist-Tourenpläne zu den Probleminstanzen A1 sind in Tabelle 5.33 (oben) dargestellt. Für die Probleminstanz A1-I liefert das Störungsreaktionsverfahren ROUTE im Mittel die Lösungen mit den geringsten Verspätungen. Die mittlere Verspätung beträgt dabei 218,70 Zeiteinheiten (ZE), das entspricht 2,5% der Plan-Gesamtbediendauer. Es werden im Mittel weniger als 3 Kunden überhaupt verspätet beliefert, bei jedem verspätet belieferten Kunden beträgt die Verspätung im Durchschnitt 74,39 Zeiteinheiten. Die mittlere Ist-Gesamtbediendauer (im folgenden $sched(X^{Ist})$) liegt mit 9006,99 Zeiteinheiten (ZE) etwa 4,4% über der mittleren Planbediendauer (im folgenden $sched(\bar{X})$) von 8623,29 ZE. Die mittlere Verspätung, die sich bei Verwendung der Verfahren EX und PT ergibt (308,03 ZE bzw. 317,36 ZE), liegt über der mittleren Verspätung mit ROUTE. Offenbar wird als Reaktion auf früh im Planungszeitraum bekannt werdende Störereignisse die Belieferungsreihenfolge derart verändert, daß sich

Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
A1 - I	ROUTE	218,70	2,94	9006,99	8623,29
	EX	308,03	3,18	9059,22	8623,29
	PT	317,36	3,24	9024,52	8623,29
	ALT	1567,91	10,29	9578,61	8623,29
A1 - II	ROUTE	171,49	2,12	8908,24	8623,29
	EX	180,79	1,82	8906,27	8623,29
	PT	181,35	1,94	8902,53	8623,29
	ALT	1055,44	8,29	9331,49	8623,29
A1 - III	PT	175,84	2,59	8892,31	8623,29
	EX	185,03	2,71	8909,30	8623,29
	ROUTE	202,77	2,88	8928,03	8623,29
	ALT	878,57	7,18	9247,50	8623,29
A1 - IV	PT	86,18	1,47	8884,09	8623,29
	ROUTE	96,12	1,71	8914,52	8623,29
	EX	101,22	1,76	8901,06	8623,29
	ALT	1131,02	8,41	9390,51	8623,29
A1 - V	ROUTE	8,31	0,29	8688,72	8623,29
	ALT	10,88	0,35	8696,87	8623,29
	EX	10,99	0,35	8687,99	8623,29
	PT	11,21	0,29	8685,21	8623,29
A1 - VI	PT	837,57	7,00	9516,48	8623,29
	EX	857,86	7,29	9559,52	8623,29
	ROUTE	950,19	8,29	9576,50	8623,29
	ALT	3955,54	19,41	10345,10	8623,29
Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
A2 - I	PT	145,91	1,82	8917,61	8594,29
	EX	170,00	1,88	8927,85	8594,29
	ROUTE	183,68	2,18	8922,12	8594,29
	ALT	1302,86	8,12	9353,62	8594,29
A2 - II	PT	50,80	0,76	8782,13	8594,29
	EX	60,82	0,82	8784,01	8594,29
	ROUTE	68,62	0,94	8783,09	8594,29
	ALT	904,44	6,06	9202,67	8594,29
A2 - III	PT	117,58	2,06	8895,30	8594,29
	EX	118,78	2,12	8901,32	8594,29
	ROUTE	119,59	1,82	8884,90	8594,29
	ALT	1759,28	9,47	9349,21	8594,29
A2 - IV	PT	136,79	1,82	8846,50	8594,29
	EX	166,51	2,12	8861,68	8594,29
	ROUTE	179,06	2,18	8865,44	8594,29
	ALT	1210,40	9,94	9380,11	8594,29
A2 - V	PT	1,20	0,06	8633,05	8594,29
	EX	1,75	0,12	8634,43	8594,29
	ROUTE	3,73	0,18	8642,48	8594,29
	ALT	5,02	0,24	8652,26	8594,29
A2 - VI	PT	612,76	6,00	9282,27	8594,29
	EX	672,90	6,29	9428,17	8594,29
	ROUTE	817,04	7,41	9433,46	8594,29
	ALT	3452,54	16,12	10189,61	8594,29

Tabelle 5.33: Ergebnistabellen dTS-A1 und dTS-A2

beim Eintritt von später bekanntwerdenden Störereignissen die neuen Touren und Routen als nachteilig erweisen.

Wird auf Störereignisse gar nicht reagiert (Reaktionsverfahren ALT), beträgt $sched(X^{Ist})$ 9578,61 ZE, etwa 11,1% über der mittleren Planbediendauer. Gleichzeitig werden im Mittel 10,29 Kunden zu spät beliefert. Werden die Einzelergebnisse zu ALT betrachtet (vgl. Tabelle 5.32), zeigt sich, daß relativ unabhängig von der Lösungsgüte der Planlösung X die Gesamtverspätung hoch ausfällt. Bei relativ guten Plan-Tourenplänen X ist auch die Ist-Gesamtbediendauer $sched(X^{Ist})$ tendentiell gering, umgekehrt fällt bei relativ schlechten Ausgangslösungen X auch eine relativ nachteilige Ist-Gesamtbediendauer $sched(X^{Ist})$ an.

Die Probleminstanz A1-II ist im Vergleich zu A1-I durch einen Anteil von 20% von Störereignissen mit $\Delta_\sigma > 1$ gekennzeichnet. Ein Teil der Fahrzeiten ist zu hoch geschätzt worden. Durch eine Anpassung der Routen und ggf. der Touren sollte eine bessere Lösung als bei A1-I zu erwarten sein. Diese Erwartung wird erfüllt, wie die mittlere Verspätung zeigt. Sie ist für A1-II bei allen untersuchten Störungsreaktionsverfahren niedriger als bei A1-I. Auch für A1-II ist ROUTE das beste der Störungsreaktionsverfahren, gefolgt von EX, PT und ALT. Die relative Vorteilhaftigkeit der Verfahren ist die gleiche wie bei A1-I, wobei die Unterschiede der mittleren Verspätungen mit ROUTE, EX und PT deutlich geringer ausfallen als bei A1-I. Es fällt auf, daß EX und PT eine geringere Gesamtbediendauer benötigen als ROUTE. Offenbar wird die Belieferungsreihenfolge und die dazugehörigen neuen Routen einiger Kundenaufträge zunächst als Reaktion auf Störereignisse derart verändert, daß die Gesamtbediendauer (noch ohne Verspätungen) minimiert wird. Später treten dann weitere Störereignisse auf, die diese neuen Touren und Routen unvorteilhaft werden lassen. Ein Rücktausch ist in diesen Fällen allerdings nicht mehr möglich, weil Teile der Touren bereits zurückgelegt worden sind. Insgesamt fällt dann eine größere Verspätung an. Bei einer Betrachtung der Einzelergebnisse zu A1-II zeigt sich, daß das Reaktionsverfahren ROUTE für nur 2 von 17 Tourenplänen (auf Basis von dynACO-CX2-25 best-fit und auf Basis von MACS-CX7 first-fit) die minimale Verspätung ermittelt.²⁰ Allerdings führt die Änderung der Touren und Routen durch EX und PT zu so deutlich höheren Verspätungen, daß ROUTE insgesamt im Durchschnitt die geringste Verspätung liefert. Die mittlere Verspätung der Touren und Routen, die mit ALT ermittelt wurden, liegt bei etwa dem fünffachen der mittleren Verspätung bei Einsatz von EX, PT und ROUTE.

Die Lösungen zur Probleminstanz A1-III basieren auf einem Störplan mit weniger zahlreichen, länger andauernden Störereignissen. Die mittlere Gesamtverspätung wird

²⁰Die Detailergebnisse werden im folgenden nicht in tabellarischer Form angegeben, da dies den Rahmen dieser Arbeit sprengen würde.

durch PT minimiert, gefolgt von EX, ROUTE und ALT. Mit dem Einsatz des Störungsreaktionsverfahrens PT lässt sich die mittlere Verspätung auf etwa 20% der mittleren Verspätung bei Plan-Routen und Plan-Touren senken, von 878,57 ZE auf 175,84 ZE. Selbst wenn nur die Routen angepasst werden, sinkt die mittlere Verspätung auf 202,77 ZE, etwa 23% der mittleren Verspätung ohne Anpassung der Routen.

Die Problem Instanz A1-IV unterscheidet sich von A1-I dahingehend, daß 50% der Störereignisse eine Vorwarnzeit ($b_\sigma < e_\sigma$) besitzen, statt nur 10% der Störereignisse. Es zeigt sich, daß PT die beste Lösung ermittelt, mit einer mittleren Verspätung von 86,18 ZE (gegenüber 218,70 ZE bei A1-I). Von dem größeren Anteil von Störereignissen mit Vorwarnzeit profitiert auch ROUTE stark und liefert mit einer mittleren Verspätung von 96,12 ZE die zweitbeste Lösung. Im Mittel werden bei PT, ROUTE und EX weniger als zwei Kunden verspätet beliefert. Gegenüber ALT zeigt sich deutlich, daß die Anwendung von Störungsreaktionsverfahren vorteilhaft ist: die mittlere Verspätung beträgt auch mit einem größeren Anteil von Störereignissen mit Vorwarnzeit noch 1131,02 ZE. Da die Störpläne I und IV unabhängig voneinander generiert worden sind, unterscheiden sie sich nicht nur hinsichtlich der Vorwarnzeiten. Es sind auch unterschiedliche Pfeile zu unterschiedlichen Zeiten von unterschiedlichen Störereignissen betroffen und somit unterschiedlich starke Beeinträchtigungen der Fahrzeiten bei der gleichen Auslieferungstour und -route eingetreten.

Bei der Problem Instanz A1-V beeinträchtigen die Störereignisse den Verkehr weniger stark als bei A1-I ($\Delta_\sigma \in [0, 6; 0, 9]$ statt $\Delta_\sigma \in [0, 1; 0, 6]$). Die Unterschiede zwischen den Plan-Bediendauern und den realisierten Bediendauern fallen daher auch geringer aus. Das beste Ergebnis zu A1-V liefert ROUTE mit einer mittleren Gesamtverspätung von 8,31 ZE und $sched(X^{Ist})$ von 8688,72 ZE. Nur bei wenigen Plan-Tourenplänen X werden einzelne Kunden verspätet beliefert (0,29 im Mittel), in der Regel werden alle Kunden pünktlich erreicht. Die Abweichung von $sched(X^{Ist})$ zu $sched(\bar{X})$ beträgt nur 0,75%. EX und PT liefern minimal schlechtere Lösungen als ALT, eine Verspätung von 10,99 bei EX und 11,21 bei PT statt 10,88 bei ALT. Eine genauere Betrachtung der Einzelergebnisse zu A1-V zeigt, daß nur für den Tourenplan, der mit MACS-CX7 first-fit bestimmt wurde, PT und EX eine schlechtere Lösung ergeben als ROUTE und ALT. Die Verspätung bei ALT und ROUTE beträgt dann 0 ZE, mit PT ergibt sich eine Verspätung von 58,61 ZE und mit EX von 45,54 ZE. Für alle übrigen Ausgangslösungen fallen entweder gar keine Verspätungen an oder sie unterscheiden sich nicht zwischen ROUTE, PT und EX. Folgende Erklärung kann für diesen Sachverhalt gegeben werden: Nachdem die ersten Störereignisse eingetreten sind, wird die Tour (noch ohne Verspätungen) angepasst und versucht $sched(X^{Ist})$ weiter zu senken. Im Verlauf der Tourdurchführung treten weitere Störereignisse auf, die die ursprüngliche Änderung unvorteilhaft werden lassen. Diese Änderung lässt

sich dann allerdings gegebenenfalls nicht mehr umkehren und somit wird zwar die Ist-Gesamtbediendauer geringer ausfallen, gleichzeitig aber eine Verspätung eintreten, die ohne Reaktionsverfahren zu vermeiden gewesen wäre.

Ein Vergleich der Probleminstanz A1-I mit der Probleminstanz A1-VI zeigt, daß mit einer Verdopplung der Anzahl an Störereignissen die zu erwartenden Verspätungen überproportional steigen. Die minimale mittlere Verspätung wird mit PT erreicht, sie beträgt 837,57 ZE. Für die Auslieferung werden 9516,48 ZE benötigt, anstelle der geplanten 8623,29 ZE. Im Mittel werden 7 von 50 Kunden verspätet erreicht. Besonders groß fällt die mittlere Verspätung bei ALT aus, sie beträgt bei einer durchschnittlichen Plan-Gesamtbediendauer von 8623,29 ZE durchschnittlich 3955,54 ZE. Dabei werden fast 40% der Kunden zu spät erreicht. Auch wenn die Störungsreaktionsverfahren nicht sicherstellen können, daß alle Kunden rechtzeitig erreicht werden, ist es dennoch sinnvoll sie zu verwenden. Die Handlungsalternative nichts zu tun (ALT) führt nahezu immer zu noch schlechteren Lösungen. Selbst wenn man sich darauf beschränkt nur die Routen anzupassen (ROUTE) und die Touren unverändert lässt, sind enorme Verringerungen der Verspätung erzielbar (950,19 ZE mit ROUTE statt 3955,54 ZE mit ALT).

Probleminstanzen A2-I bis A2-VI und A3-I bis A3-VI: Im Vergleich zu A1 enthält das Verkehrsnetzwerke 50b der Probleminstanzen zu A2 und A3 mehr Pfeile. Somit stehen mehr alternative Routen zur Verfügung. Es ist zu erwarten, daß die Störungsreaktionsverfahren die höheren Freiheitsgrade zur Touren- und -routenanpassung nutzen und damit die mittleren Verspätungen geringer ausfallen als bei A1. Die durchschnittlichen Ergebnisse der Ist-Tourenpläne zu den Probleminstanzen A2 sind in Tabelle 5.33 (unten) dargestellt.

Zunächst fällt auf, daß bei allen untersuchten Störplänen das Störungsreaktionsverfahren PT zu den Probleminstanzen A2 die Lösungen mit der geringsten mittleren Verspätung liefert. Die nächstbesten Lösungen liefert immer EX, gefolgt von ROUTE. ALT ist immer die schlechteste Handlungsalternative. Die größere Anzahl an Möglichkeiten auf alternative Wege auszuweichen und somit Störungen zu umfahren, wird besonders von PT und EX genutzt, da mit einer angepassten Belieferungsreihenfolge beim Umfahren eines Staus auch ursprünglich für einen späteren Zeitpunkt vorgesehene Kundenaufträge “unterwegs” bedient werden können.

Die Probleminstanzen, die auf der dTP-Probleminstanz A3 basieren (Ergebnisse vgl. Tabelle 5.34 oben) sind durch stärker dynamische Fahrzeiten bereits während der Planung gekennzeichnet. Dementsprechend fällt die mittlere Bediendauer $sched(\bar{X})$ bei A3 im Mittel niedriger als bei A2 aus, sie beträgt 8369,82 ZE. Abgesehen von

Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
A3 - I	EX	189,10	1,35	8650,23	8369,82
	PT	194,47	1,35	8639,36	8369,82
	ROUTE	273,50	2,12	8714,33	8369,82
	ALT	1087,17	8,12	9179,48	8369,82
A3 - II	PT	41,69	0,65	8486,96	8369,82
	EX	43,24	0,71	8557,60	8369,82
	ROUTE	51,10	0,88	8585,20	8369,82
	ALT	383,66	3,76	8910,91	8369,82
A3 - III	PT	66,93	1,06	8566,17	8369,82
	EX	70,10	1,18	8580,38	8369,82
	ROUTE	109,08	1,65	8608,87	8369,82
	ALT	738,63	6,12	8935,99	8369,82
A3 - IV	ROUTE	19,90	0,47	8565,71	8369,82
	EX	22,24	0,53	8551,47	8369,82
	PT	28,89	0,59	8556,36	8369,82
	ALT	339,23	3,82	8852,17	8369,82
A3 - V	PT	0,00	0,00	8401,75	8369,82
	EX	1,68	0,06	8404,21	8369,82
	ROUTE	3,42	0,18	8424,58	8369,82
	ALT	4,23	0,18	8440,86	8369,82
A3 - VI	EX	609,51	5,65	9072,43	8369,82
	PT	609,58	5,71	9089,58	8369,82
	ROUTE	617,03	6,00	9071,27	8369,82
	ALT	2614,89	16,06	9719,76	8369,82
Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
B1 - I	EX	107,12	1,59	11088,71	10974,41
	PT	110,23	1,65	11038,50	10974,41
	ROUTE	110,47	1,71	11111,38	10974,41
	ALT	1143,25	5,47	11349,86	10974,41
B1 - II	EX	23,18	0,41	11022,32	10974,41
	PT	33,60	0,53	10978,00	10974,41
	ROUTE	39,82	0,65	11034,13	10974,41
	ALT	555,32	3,59	11189,89	10974,41
B1 - III	ROUTE	51,35	1,24	11028,67	10974,41
	EX	91,63	0,47	11023,95	10974,41
	PT	115,26	0,59	10976,51	10974,41
	ALT	217,90	1,76	11159,71	10974,41
B1 - IV	EX	44,49	1,00	11072,39	10974,41
	PT	49,84	1,12	11018,97	10974,41
	ROUTE	89,05	1,47	11074,64	10974,41
	ALT	609,04	3,47	11241,72	10974,41
B1 - V	PT	0,00	0,00	10940,19	10974,41
	ROUTE	0,00	0,00	11000,68	10974,41
	ALT	0,00	0,00	11002,36	10974,41
	EX	2,53	0,06	10991,57	10974,41
B1 - VI	EX	555,24	4,06	11240,03	10974,41
	PT	565,88	4,24	10891,19	10974,41
	ROUTE	691,83	4,94	11313,93	10974,41
	ALT	4537,35	15,06	11877,52	10974,41

Tabelle 5.34: Ergebnistabellen dTS-A3 und dTS-B1

A3-IV sind die Störungsreaktionsverfahren EX und PT, die die Belieferungsreihenfolge anpassen, den Verfahren ROUTE und ALT überlegen. Bei A3-IV ist die tatsächliche Bediendauer $sched(X^{Ist})$ bei PT und EX geringer als bei ROUTE, die Verspätung allerdings größer: Eine erst später als ungeeignet erkennbare Veränderung der vorläufigen Belieferungsreihenfolgen, die mit INS-CX7 first-fit und mit MACS-CX7 first-fit ermittelt wurden, führt zu einer höheren mittleren Verspätung. Für die meisten Tourenpläne X gilt, daß für A3-IV mit PT und EX keine Verspätung anfällt. Der erhöhte Anteil von Störereignissen mit Vorwarnzeit bietet viele Verbesserungsmöglichkeiten, die durch die Störungsreaktionsverfahren PT und EX auch genutzt werden. Bei der Probleminstanz A3-V fällt auf, daß bei schwachen Beeinträchtigungen der Geschwindigkeiten durch Störereignisse mit PT sogar für alle untersuchten Tourenpläne die zu beliefernden Kunden noch pünktlich erreicht werden können. Die Störereignisse erhöhen zwar die $sched(X^{Ist})$ im Vergleich zu $sched(\bar{X})$ um 31,93 ZE ohne dabei allerdings zu Verspätungen zu führen.

dTS auf Basis der Probleminstanzen B1, B2 und B3 Die Ergebnisse der Lösungsläufe zu den Probleminstanzen B1 sind in Tabelle 5.34 (unten), die Ergebnisse zu den Probleminstanzen B2 und B3 sind in Tabelle 5.35 dargestellt. Für 13 von 18 Problemen liefert EX die Lösung mit der geringsten Gesamtverspätung. Für die Probleme B1-V, B2-V, B3-IV und B3-V ist PT dem Störungsreaktionsverfahren EX überlegen. Nur für B1-III liefert ROUTE die Lösung mit der geringsten mittleren Verspätung, wobei mit ROUTE im Mittel 1,24 Kunde verspätet beliefert werden, während mit PT und EX durchschnittlich weniger als ein Kunde verspätet erreicht wird. Werden die Detailergebnisse betrachtet, zeigt sich, daß ROUTE ausgehend von jedem Plan-Tourenplan zu Verspätungen führt, diese Verspätungen allerdings relativ gleichmäßig und gering ausfallen. Werden die Ist-Tourenpläne mit PT ermittelt, werden 12 von 17 Tourenpläne ohne Verspätungen durchgeführt, mit EX sogar 13 von 17. Die Lösungsgüte ist in diesen Fällen besser als beim Einsatz von ROUTE. Die wenigen Ist-Tourenpläne mit Verspätungen sind allerdings stark verspätet, was im Durchschnitt zu einer höheren mittleren Verspätung führt. Vor einem praktischen Einsatz der Störungsreaktionsverfahren sind detaillierte Untersuchungen von Kombinationen aus Lösungsalgorithmus und Störungsreaktionsverfahren unerlässlich, um die Stabilität und die Lösungsgüte der Lösungsverfahren in einem dynamischen und stochastischen Umfeld beurteilen zu können. Eine solche Analyse würde an dieser Stelle allerdings zu weit gehen.

Das relativ schlechte Abschneiden von EX für die Probleminstanz B1-V ist zunächst überraschend. Bei einer Betrachtung der Detailergebnisse findet sich ein einzelner Ausreißerwert (für den Ist-Tourenplan, der auf dem Plan-Tourenplan, der mit MACS-CX2 ermittelt wurde, basiert), der zu einer derart großen Verspätung führt,

Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
B2 - I	EX	127,86	1,82	12444,67	12325,35
	PT	189,68	2,12	12409,05	12325,35
	ROUTE	267,44	2,71	12511,55	12325,35
	ALT	1765,63	7,88	12838,77	12325,35
B2 - II	EX	45,62	0,53	12388,12	12325,35
	PT	109,79	0,94	12359,20	12325,35
	ROUTE	125,48	1,18	12432,93	12325,35
	ALT	939,69	5,18	12640,19	12325,35
B2 - III	EX	124,35	1,00	12347,86	12325,35
	PT	177,46	1,24	12315,17	12325,35
	ROUTE	272,52	2,59	12424,22	12325,35
	ALT	948,02	4,53	12631,09	12325,35
B2 - IV	EX	19,98	0,94	12385,06	12325,35
	PT	64,30	1,35	12358,24	12325,35
	ROUTE	73,63	1,59	12429,07	12325,35
	ALT	740,80	4,47	12689,13	12325,35
B2 - V	PT	0,00	0,00	12291,81	12325,35
	EX	0,00	0,00	12320,66	12325,35
	ROUTE	0,00	0,00	12363,11	12325,35
	ALT	0,00	0,00	12367,25	12325,35
B2 - VI	EX	855,61	5,12	12636,46	12325,35
	PT	1168,19	5,71	12367,81	12325,35
	ROUTE	1171,26	5,88	12729,12	12325,35
	ALT	6932,14	18,12	13537,12	12325,35
Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
B3 - I	EX	101,48	1,29	9962,79	9867,12
	ROUTE	107,01	1,47	9978,43	9867,12
	PT	111,02	1,35	9932,47	9867,12
	ALT	1167,90	5,88	10257,74	9867,12
B3 - II	EX	40,11	0,65	9933,82	9867,12
	PT	40,66	0,65	9902,31	9867,12
	ROUTE	41,46	0,71	9949,78	9867,12
	ALT	608,27	4,53	10128,41	9867,12
B3 - III	EX	50,89	1,12	9916,66	9867,12
	PT	66,24	1,29	9884,97	9867,12
	ROUTE	70,79	1,47	9941,48	9867,12
	ALT	998,14	5,24	10205,82	9867,12
B3 - IV	PT	14,33	0,41	9876,60	9867,12
	EX	14,33	0,41	9913,25	9867,12
	ROUTE	32,61	0,59	9936,79	9867,12
	ALT	372,25	3,24	10119,81	9867,12
B3 - V	PT	1,53	0,12	9834,71	9867,12
	EX	1,53	0,12	9867,89	9867,12
	ROUTE	1,53	0,12	9887,85	9867,12
	ALT	2,53	0,12	9892,52	9867,12
B3 - VI	EX	589,66	4,00	10239,16	9867,12
	PT	715,22	4,24	10241,67	9867,12
	ROUTE	952,40	5,47	10305,75	9867,12
	ALT	5238,58	15,71	10941,28	9867,12

Tabelle 5.35: Ergebnistabellen dTS-B2 und dTS-B3

daß EX im Durchschnitt am schlechtesten abschneidet. Bei B1-V wird mit PT eine Ist-Bediendauer erreicht, die geringer als die Plan-Bediendauer ist. Die Ursache dafür ist, daß die Tourenpläne, die auf NN und INS ohne CX-Verbesserungsverfahren zurückgehen, durch PT verbessert werden. Die Plan-Bediendauer beträgt beispielsweise bei INS 14745,88 ZE, mit EX beträgt die Ist-Bediendauer 14697,91 ZE und mit PT kann sie auf 14435,10 ZE gesenkt werden.

Die Tourenpläne zu B2 weisen im Vergleich zu B1 eine höhere mittlere Gesamtbediendauer $sched(X^{Ist})$ auf. Der Grund dafür ist das schlechte Abschneiden von MACS im Rahmen der Tourenplanung. Als Folge ist auch die mittlere Verspätung bei B2 höher als bei B1. In Verbindung mit Störplan V können sowohl EX als auch PT die Ist-Bediendauer unter die Plan-Bediendauer senken. Dabei werden nicht nur die Tourenpläne, die von INS und NN ermittelt worden sind verbessert, auch der Tourenplan, der mit MACS ermittelt wurde, wird verbessert.

Die Probleminstanzen B3 weisen eine deutlich geringere mittlere Plan-Gesamtbediendauer $sched(\bar{X})$ und mittlere Ist-Gesamtbediendauer $sched(X^{Ist})$ auf als B2. Die Verspätung ist immer geringer als bei B2, mit Ausnahme von der Handlungsalternative ALT bei B3-III. Bei B3-V weisen alle Störungsreaktionsverfahren eine Verspätung auf. Die (sehr geringe) Verspätung ist auf zwei Einzelergebnisse zurückzuführen, sie stimmt zudem bei ROUTE, PT und EX überein. PT erreicht dabei wiederum eine Ist-Gesamtbediendauer die kleiner ist als die Plan-Gesamtbediendauer. Das ist auf die Verbesserung der Tourenpläne, die mit INS und NN ermittelt wurden, zurückzuführen.

dTS auf Basis der Probleminstanzen C, D, E und F

Die Ergebnisse der Lösungsläufe zu den Probleminstanzen C1 und C2 sind in Tabelle 5.36, die Ergebnisse zu den Probleminstanzen C3 in Tabelle 5.37 (oben) dargestellt.

Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
C1 - I	ROUTE	185,04	1,88	8332,22	7859,82
	EX	253,94	1,65	8356,36	7859,82
	PT	273,23	1,88	8353,99	7859,82
	ALT	595,15	4,82	8764,19	7859,82
C1 - II	EX	26,68	0,35	8155,57	7859,82
	PT	35,29	0,53	8138,41	7859,82
	ROUTE	69,40	0,94	8200,96	7859,82
C1 - III	ALT	409,73	3,12	8522,04	7859,82
	EX	0,00	0,00	8120,81	7859,82
	ROUTE	7,94	0,12	8196,33	7859,82
C1 - IV	PT	17,21	0,12	8125,94	7859,82
	ALT	322,71	3,12	8652,75	7859,82
	EX	23,75	0,35	8203,78	7859,82
C1 - V	PT	26,07	0,47	8202,46	7859,82
	ROUTE	62,34	0,76	8236,84	7859,82
	ALT	417,67	3,29	8642,52	7859,82
	PT	0,00	0,00	7848,78	7859,82
C1 - VI	EX	0,00	0,00	7867,67	7859,82
	ROUTE	0,00	0,00	7929,68	7859,82
	ALT	0,00	0,00	7935,29	7859,82
	PT	579,55	4,41	8905,38	7859,82
C2 - I	ROUTE	589,00	4,76	8828,33	7859,82
	EX	614,49	4,76	8862,00	7859,82
	ALT	1770,81	9,18	9339,66	7859,82
C2 - II	ROUTE	223,81	1,88	8376,03	7919,35
	EX	288,37	1,82	8375,48	7919,35
	PT	296,86	1,88	8270,03	7919,35
	ALT	554,74	4,41	8853,00	7919,35
C2 - III	PT	18,04	0,29	8056,82	7919,35
	EX	20,78	0,29	8178,27	7919,35
	ROUTE	27,43	0,53	8220,23	7919,35
	ALT	267,74	2,65	8570,12	7919,35
C2 - IV	EX	0,55	0,06	8169,62	7919,35
	PT	18,41	0,18	8043,85	7919,35
	ROUTE	19,94	0,29	8233,78	7919,35
	ALT	429,46	3,06	8696,51	7919,35
C2 - V	EX	5,17	0,18	8261,76	7919,35
	PT	9,96	0,29	8167,20	7919,35
	ROUTE	49,43	0,76	8258,31	7919,35
	ALT	466,54	3,53	8674,35	7919,35
C2 - VI	PT	0,44	0,06	7902,53	7919,35
	EX	0,44	0,06	7913,06	7919,35
	ROUTE	1,15	0,06	7989,61	7919,35
	ALT	1,15	0,06	7998,61	7919,35
C2 - VII	PT	553,00	4,06	8852,51	7919,35
	EX	578,22	4,24	8860,45	7919,35
	ROUTE	593,63	4,24	8828,80	7919,35
	ALT	1458,37	8,41	9424,11	7919,35

Tabelle 5.36: Ergebnistabellen dTS-C1 und dTS-C2

Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
C3 - I	PT	36,90	0,35	7639,53	7451,71
	ROUTE	72,25	0,76	7794,90	7451,71
	EX	79,73	0,82	7742,20	7451,71
	ALT	230,84	2,35	8152,93	7451,71
C3 - II	PT	4,74	0,18	7558,21	7451,71
	EX	5,21	0,12	7652,34	7451,71
	ROUTE	19,07	0,35	7685,83	7451,71
C3 - III	ALT	118,79	1,59	7976,56	7451,71
	PT	11,01	0,12	7600,01	7451,71
	EX	15,62	0,18	7628,80	7451,71
	ROUTE	18,04	0,24	7647,32	7451,71
C3 - IV	ALT	190,18	1,65	8013,87	7451,71
	PT	13,72	0,29	7610,01	7451,71
	ROUTE	21,60	0,35	7729,82	7451,71
	EX	36,97	0,41	7726,69	7451,71
C3 - V	ALT	129,85	1,65	8040,36	7451,71
	PT	0,00	0,00	7478,77	7451,71
	EX	0,00	0,00	7493,75	7451,71
	ROUTE	0,00	0,00	7514,14	7451,71
C3 - VI	ALT	0,16	0,06	7523,94	7451,71
	PT	102,85	1,29	8076,83	7451,71
	EX	138,96	1,29	8079,74	7451,71
	ROUTE	166,03	1,82	8073,64	7451,71
C3 - VII	ALT	801,80	5,24	8581,89	7451,71
	PT				
	EX				
	ROUTE				
Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
D1 - I	EX	59,62	0,71	9204,62	8991,71
	PT	66,02	0,71	9206,31	8991,71
	ROUTE	90,47	1,18	9231,78	8991,71
	ALT	494,26	4,00	9575,39	8991,71
D1 - II	EX	50,57	0,59	9130,35	8991,71
	PT	55,96	0,53	9133,41	8991,71
	ROUTE	76,22	0,88	9166,77	8991,71
	ALT	230,10	2,00	9352,54	8991,71
D1 - III	EX	18,67	0,41	9190,39	8991,71
	PT	21,30	0,41	9199,73	8991,71
	ROUTE	70,80	1,00	9247,15	8991,71
	ALT	503,90	3,59	9565,63	8991,71
D1 - IV	EX	48,32	0,65	9230,90	8991,71
	PT	55,85	0,65	9243,69	8991,71
	ROUTE	77,17	0,88	9282,47	8991,71
	ALT	236,78	2,41	9498,10	8991,71
D1 - V	EX	1,31	0,12	8997,31	8991,71
	ROUTE	3,88	0,18	9050,99	8991,71
	ALT	3,88	0,18	9056,63	8991,71
	PT	6,85	0,12	8996,46	8991,71
D1 - VI	EX	490,19	5,06	9807,52	8991,71
	PT	526,18	5,35	9755,71	8991,71
	ROUTE	632,20	5,65	9740,89	8991,71
	ALT	1713,06	10,06	10258,30	8991,71

Tabelle 5.37: Ergebnistabellen dTS-C3 und dTS-D1

Bei C liegt die Ist-Gesamtbediendauer im Mittel 5,61% über der

Plan-Gesamtbediendauer. Für 11 von 18 Tourenpläne mit Störereignissen liefert PT und für 5 von 18 EX die geringste durchschnittliche Verspätung. In zwei Fällen (C1-I und C1-II) liefert ROUTE die Lösung mit der geringsten mittleren Verspätung. Die Handlungsalternative ALT ist für alle Probleminstanzen C immer die schlechteste Handlungsalternative.

Die weiten Zeitfenster lassen mehr Flexibilität bei den Anlieferungszeitpunkten zu, so daß die mittleren Verspätungen, gerade mit PT und mit EX, geringer ausfallen als bei den Probleminstanzen A. Die Störungsreaktionsverfahren EX und PT können als Reaktion auf Störereignisse die Touren anpassen, wobei eine relativ große Anzahl an Touren und Routen existierten, die insgesamt ohne Verspätungen durchgeführt werden können. Wird nur eine Anpassung der Routen zugelassen (ROUTE), wird viel Verbesserungspotential verschenkt.

Bei den Probleminstanzen C1, C2 und C3 (weite Zeitfensterrestriktionen, geringe Fahrzeugkapazitäten), kombiniert mit Störplan V (weniger starke Beeinträchtigungen der Geschwindigkeiten durch Störereignisse) treten nahezu keine oder nur sehr vereinzelt Verspätungen auf. PT erreicht bei C1-V und C2-V eine geringere mittlere Ist-Bediendauer als geplant. Bei C2-V wird auch mit EX eine Verringerung der mittleren Bediendauer erzielt. Die Probleme C1-III können für alle betrachteten Plan-Tourenpläne ohne Verspätung gelöst werden, wenn als Störungsreaktionsverfahren EX verwendet wird. Zwar ist die Ist-Belieferungszeit größer als die Plan-Belieferungszeit, aber dennoch werden durch eine angepasste Belieferungsreihenfolge alle Kunden rechtzeitig erreicht. Bei den Probleminstanzen C3 fallen insgesamt relativ geringere Verspätungen an, sowohl im Vergleich zu C2 als auch im Vergleich zu A3.

Bei den Probleminstanzen D (vgl. Tabellen 5.37 unten und 5.38) liegt die Ist-Gesamtbediendauer im Mittel 3,75% über der Plan-Gesamtbediendauer. Die mittlere Plan-Bediendauer $sched(\bar{X})$ ist bei D2 kleiner als bei D1, und bei D3 am geringsten. Die mittleren Verspätungen sind bei D in der Regel deutlich geringer als bei A, der 33 prozentige Anteil an Kundenaufträgen ohne Zeitfensterrestriktionen erleichtert die pünktliche Belieferung eines nicht vernachlässigbaren Teils der Kunden. Für alle untersuchten Probleminstanzen D sind PT und EX besser als ROUTE und ALT. Für 10 von 18 Problemen liefert EX die geringsten Verspätungen. Für die übrigen 8 Probleme werden mit dem Störungsreaktionsverfahren PT die geringsten Verspätungen erzielt. Die Möglichkeit, Kundenaufträge ohne Zeitfensterrestriktionen als Reaktion auf Störereignisse an einen späteren Platz auf den Touren zu verschieben, ohne daß es dadurch zu Verspätungen kommt, erweist sich als vorteilhaft.

Für die Probleminstanzen D1-V liefert PT die größten Verspätungen (allerdings nur 6,85 ZE), die relativ höchste Verspätung fällt dabei mit der geringsten mittleren

Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
D2 - I	EX	41,89	0,76	9136,99	8888,18
	PT	42,76	0,76	8981,97	8888,18
	ROUTE	108,40	1,53	9167,74	8888,18
	ALT	654,89	5,00	9604,45	8888,18
D2 - II	EX	19,09	0,35	9053,90	8888,18
	PT	25,46	0,47	9031,24	8888,18
	ROUTE	53,18	0,76	9091,30	8888,18
	ALT	505,25	3,53	9415,14	8888,18
D2 - III	PT	110,31	1,12	9085,55	8888,18
	EX	121,44	1,18	9108,71	8888,18
	ROUTE	142,40	1,53	9154,97	8888,18
	ALT	635,02	4,71	9477,14	8888,18
D2 - IV	PT	42,37	0,82	9069,95	8888,18
	EX	45,98	0,88	9096,07	8888,18
	ROUTE	58,59	1,00	9132,17	8888,18
	ALT	298,30	3,12	9412,38	8888,18
D2 - V	PT	2,53	0,06	8883,89	8888,18
	EX	2,53	0,06	8915,86	8888,18
	ROUTE	5,09	0,12	8958,84	8888,18
	ALT	8,11	0,18	8965,25	8888,18
D2 - VI	PT	551,70	4,88	9511,73	8888,18
	EX	601,59	5,00	9798,80	8888,18
	ROUTE	889,20	6,47	9766,60	8888,18
	ALT	2305,36	12,41	10335,92	8888,18
Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
D3 - I	PT	53,33	0,65	8556,57	8351,71
	EX	68,20	0,71	8563,04	8351,71
	ROUTE	99,66	1,18	8602,07	8351,71
	ALT	352,36	3,35	8914,31	8351,71
D3 - II	PT	37,75	0,53	8502,67	8351,71
	EX	52,61	0,59	8503,38	8351,71
	ROUTE	65,91	0,82	8542,75	8351,71
	ALT	245,75	2,41	8801,00	8351,71
D3 - III	PT	16,02	0,29	8443,07	8351,71
	EX	18,04	0,29	8473,73	8351,71
	ROUTE	38,69	0,71	8519,39	8351,71
	ALT	215,80	2,06	8745,91	8351,71
D3 - IV	EX	20,46	0,29	8486,20	8351,71
	PT	32,41	0,41	8470,58	8351,71
	ROUTE	46,58	0,82	8562,12	8351,71
	ALT	223,67	2,12	8704,63	8351,71
D3 - V	PT	0,37	0,06	8314,21	8351,71
	EX	0,37	0,06	8335,90	8351,71
	ROUTE	1,85	0,18	8396,01	8351,71
	ALT	2,88	0,18	8403,97	8351,71
D3 - VI	EX	138,67	1,82	8888,56	8351,71
	PT	158,16	2,12	8859,34	8351,71
	ROUTE	361,66	3,94	8968,67	8351,71
	ALT	1098,37	7,82	9406,92	8351,71

Tabelle 5.38: Ergebnistabellen dTS-D2 und dTS-D3

Ist-Bediendauer ($sched(X^{Ist}) = 8996,46$ ZE) zusammen.

Für die Probleminstanzen D2-V kann mit PT eine mittlere Ist-Bediendauer $sched(X^{Ist})$ von 8883,80 ZE erreicht werden, die geringer als die Plan-Bediendauer $sched(\bar{X})$ mit 8888,19 ZE ist. Die gleiche Beobachtung lässt sich für die Probleminstanzen D3-V mit den Störungsreaktionsverfahren PT und EX machen. Auch hier ist die Ursache für die durchschnittlich geringere Gesamtverspätung die Verbesserung der Ausgangslösungen von NN und INS ohne CX-Verbesserungsverfahren.

Die Ergebnisse zu den Probleminstanzen E1 und E2 finden sich in den Tabellen 5.39, die Ergebnisse zu E3 in Tabelle 5.40 oben. Bei den Probleminstanzen E liefert PT in 13 von 18 Fällen die besten Ist-Tourenpläne. Mit dem Störungsreaktionsverfahren EX werden stets bessere Lösungen ermittelt als mit ROUTE. ALT liefert immer die Lösungen mit den größten Verspätungen.

Für die Probleminstanz E2-V liefert EX die geringste Verspätung. PT erreicht bei etwas höherer Verspätung eine mittlere Ist-Bediendauer $sched(X^{Ist})$, die geringer ausfällt als die mittlere Plan-Bediendauer $sched(\bar{X})$. Auch hier ist die Ursache die Verbesserung der Tourenpläne, die ohne CX-Verbesserungsverfahren ermittelt wurden. Bei den Probleminstanzen E3-V treten keine Verspätungen auf. Für die Probleminstanzen E3-I liefert ROUTE die geringste Verspätung.

Bei den Probleminstanzen F liegt die Ist-Gesamtbediendauer im Mittel 4,34% über der Plan-Gesamtbediendauer (vgl. Tabelle 5.40 unten und Tabelle 5.41). Für die Probleminstanzen F2-V, F2-VI, F3-I und F3-II wird mit PT die mittlere Plan-Bediendauer unterschritten. Das Störungsreaktionsverfahren EX verbessert die mittlere Bediendauer für die Probleminstanzen F2-V und F3-V.

Gerade für die Probleminstanz F ist EX ein besonders gut geeignetes Störungsreaktionsverfahren. Die Gesamtverspätung ist in 14 von 18 Fällen am kleinsten, wobei der Abstand zu ROUTE und ALT sehr deutlich ausfällt. Für F1-II, F2-V und F3-V schneidet PT im Mittel besser ab als EX. Nur für F1-V liefert EX eine unbefriedigende Lösung. Die Ursache ist, daß bei mehreren Plan-Tourenplänen Tauschvorgänge vorgenommen werden, die sich nachträglich als unvorteilhaft erweisen und nicht mehr rückgängig gemacht werden können.

5.6.2.4 Zusammenfassung der Ergebnisse der experimentellen Analyse von dynamisch-stochastischen Tourensteuerungsproblemen (dTS)

Das dynamisch-deterministische Tourenplanungsproblem (dTP) erzeugt Tourenpläne auf Basis der Informationen, die zum Planungszeitpunkt vorliegen. Die tatsächlichen Ausprägungen der Fahrzeiten zwischen einzelnen Kunden und zwischen den Kunden und dem Depot können von den Planungen abweichen. Diese Abweichungen

Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
E1 - I	PT	155,74	2,41	11759,78	11549,53
	EX	173,77	2,53	11977,26	11549,53
	ROUTE	182,63	2,41	12002,60	11549,53
	ALT	1861,23	12,47	12695,37	11549,53
E1 - II	PT	99,23	1,59	11777,31	11549,53
	EX	108,78	1,65	11903,79	11549,53
	ROUTE	130,02	2,18	11916,76	11549,53
	ALT	1186,24	9,06	12487,76	11549,53
E1 - III	PT	180,14	2,82	11606,50	11549,53
	EX	199,01	2,94	11906,86	11549,53
	ROUTE	371,46	4,24	11963,89	11549,53
	ALT	1843,44	12,82	12516,54	11549,53
E1 - IV	PT	398,32	4,53	11760,72	11549,53
	EX	405,23	4,53	12106,53	11549,53
	ROUTE	554,43	6,12	12112,86	11549,53
	ALT	2043,77	15,18	12588,18	11549,53
E1 - V	PT	5,67	0,18	11622,74	11549,53
	EX	5,67	0,18	11635,54	11549,53
	ROUTE	5,67	0,18	11635,17	11549,53
	ALT	9,17	0,18	11648,99	11549,53
E1 - VI	PT	1146,87	10,29	12520,33	11549,53
	EX	1238,90	10,76	12650,67	11549,53
	ROUTE	1446,35	11,53	12629,82	11549,53
	ALT	4979,04	24,29	13646,38	11549,53
Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
E2 - I	EX	107,06	2,12	11650,28	11250,41
	PT	123,68	2,29	11454,89	11250,41
	ROUTE	151,42	2,41	11694,41	11250,41
	ALT	1745,01	12,65	12467,96	11250,41
E2 - II	EX	47,05	1,18	11550,94	11250,41
	PT	48,05	1,47	11346,92	11250,41
	ROUTE	53,16	1,71	11575,31	11250,41
	ALT	1295,09	10,76	12258,62	11250,41
E2 - III	EX	108,67	2,06	11566,83	11250,41
	PT	111,39	2,00	11429,30	11250,41
	ROUTE	151,49	2,71	11588,43	11250,41
	ALT	1738,20	11,41	12272,23	11250,41
E2 - IV	PT	261,98	3,47	11636,28	11250,41
	EX	277,07	3,53	11755,99	11250,41
	ROUTE	339,58	4,35	11791,42	11250,41
	ALT	1896,31	12,88	12403,30	11250,41
E2 - V	EX	7,12	0,53	11335,72	11250,41
	PT	9,65	0,53	11232,89	11250,41
	ROUTE	9,65	0,53	11351,18	11250,41
	ALT	14,98	0,59	11383,99	11250,41
E2 - VI	PT	765,45	7,00	12046,87	11250,41
	EX	787,59	7,24	12147,92	11250,41
	ROUTE	943,05	8,47	12181,17	11250,41
	ALT	5504,29	25,24	13555,26	11250,41

Tabelle 5.39: Ergebnistabellen dTS-E1 und dTS-E2

Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
E3 - I	ROUTE	70,45	1,00	10995,62	10710,47
	PT	83,43	1,12	10999,91	10710,47
	EX	84,55	1,12	10994,19	10710,47
	ALT	811,65	6,82	11523,38	10710,47
E3 - II	PT	54,49	0,76	10939,25	10703,29
	EX	58,72	0,82	10938,14	10703,29
	ROUTE	63,23	0,82	10947,47	10703,29
	ALT	510,78	4,82	11400,17	10703,29
E3 - III	PT	87,30	1,12	10999,72	10703,29
	EX	97,94	1,24	11008,83	10703,29
	ROUTE	99,57	1,29	11022,08	10703,29
	ALT	1095,90	8,94	11602,45	10703,29
E3 - IV	PT	35,34	0,59	11040,97	10703,29
	EX	37,94	0,59	11035,98	10703,29
	ROUTE	72,24	1,06	11058,06	10703,29
	ALT	1211,42	9,06	11625,04	10703,29
E3 - V	PT	0,00	0,00	10751,50	10703,29
	EX	0,00	0,00	10753,82	10703,29
	ROUTE	0,00	0,00	10767,80	10703,29
	ALT	0,00	0,00	10784,08	10703,29
E3 - VI	PT	391,79	4,35	11464,46	10703,29
	EX	418,61	4,65	11451,27	10703,29
	ROUTE	496,31	5,24	11475,70	10703,29
	ALT	3850,06	19,47	12621,56	10703,29
Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
F1 - I	EX	1037,52	7,53	11865,60	11448,65
	PT	1146,10	6,94	11842,35	11448,65
	ROUTE	2025,18	12,24	11899,50	11448,65
	ALT	9368,93	28,00	12641,14	11448,65
F1 - II	PT	515,33	4,18	11501,70	11448,65
	EX	529,34	4,71	11736,98	11448,65
	ROUTE	702,84	5,47	11680,62	11448,65
	ALT	4173,16	17,24	12186,12	11448,65
F1 - III	EX	666,21	4,76	11869,72	11448,65
	PT	768,53	6,06	11796,03	11448,65
	ROUTE	1563,26	11,12	11853,66	11448,65
	ALT	8121,91	25,76	12564,19	11448,65
F1 - IV	EX	893,82	6,41	11829,51	11448,65
	PT	1137,18	8,29	11864,66	11448,65
	ROUTE	3218,73	14,94	11986,25	11448,65
	ALT	14533,50	33,53	12858,42	11448,65
F1 - V	PT	10,09	0,18	11460,71	11448,65
	ROUTE	25,48	0,41	11509,94	11448,65
	ALT	33,00	0,47	11515,54	11448,65
	EX	34,01	0,65	11526,69	11448,65
F1 - VI	EX	4757,48	19,59	12424,82	11448,65
	PT	6672,06	23,65	12254,42	11448,65
	ROUTE	12157,35	32,82	12741,55	11448,65
	ALT	24306,42	43,47	13771,74	11448,65

Tabelle 5.40: Ergebnistabellen dTS-E3 und dTS-F1

Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
F2 - I	EX	877,48	5,47	11375,13	11062,65
	PT	905,03	5,94	11365,21	11062,65
	ROUTE	1375,53	8,12	11362,58	11062,65
	ALT	7893,22	24,06	12112,72	11062,65
F2 - II	EX	358,20	3,18	11214,83	11062,65
	PT	398,21	3,71	11228,55	11062,65
	ROUTE	798,62	5,65	11240,03	11062,65
	ALT	4442,16	18,06	11818,42	11062,65
F2 - III	EX	281,05	2,41	11262,06	11062,65
	PT	331,87	3,06	11242,69	11062,65
	ROUTE	783,03	6,29	11269,14	11062,65
	ALT	8428,92	24,94	12078,44	11062,65
F2 - IV	EX	754,10	5,53	11390,68	11062,65
	PT	882,75	6,65	11339,20	11062,65
	ROUTE	1365,79	8,29	11391,21	11062,65
	ALT	14106,62	27,71	12329,03	11062,65
F2 - V	PT	0,00	0,00	11049,04	11062,65
	EX	0,00	0,00	11058,01	11062,65
	ROUTE	20,79	0,41	11117,92	11062,65
	ALT	37,32	0,71	11129,08	11062,65
F2 - VI	EX	3145,52	15,18	11892,38	11062,65
	PT	3476,40	14,41	10542,93	11062,65
	ROUTE	5892,54	20,94	11965,27	11062,65
	ALT	21885,49	39,65	13192,55	11062,65
Problem	Verfahren	mittlere Verspätung	mittlere Anzahl verspätet	mittlere Ist- Belieferungs- zeit	mittlere Plan- Belieferungs- zeit
F3 - I	EX	369,36	2,47	10833,02	10616,06
	PT	559,16	3,59	10417,09	10616,06
	ROUTE	830,79	5,06	10957,06	10616,06
	ALT	3423,76	15,59	11473,72	10616,06
F3 - II	EX	393,07	2,53	10784,90	10616,06
	PT	462,86	3,12	10315,36	10616,06
	ROUTE	689,25	4,29	10900,25	10616,06
	ALT	2440,34	12,35	11329,21	10616,06
F3 - III	EX	127,97	1,41	10786,36	10616,06
	PT	191,15	2,00	10842,02	10616,06
	ROUTE	445,38	3,82	10908,43	10616,06
	ALT	3032,98	13,24	11379,83	10616,06
F3 - IV	EX	276,41	2,47	10933,54	10616,06
	PT	283,64	2,71	10942,12	10616,06
	ROUTE	780,53	5,47	11033,81	10616,06
	ALT	6503,71	22,88	11704,05	10616,06
F3 - V	PT	0,13	0,06	10616,67	10616,06
	EX	3,71	0,12	10607,84	10616,06
	ROUTE	10,42	0,18	10703,17	10616,06
	ALT	10,68	0,18	10709,99	10616,06
F3 - VI	EX	2363,36	12,53	11280,45	10616,06
	PT	2563,07	13,00	11297,29	10616,06
	ROUTE	3882,11	16,71	11442,66	10616,06
	ALT	14610,10	32,65	12383,09	10616,06

Tabelle 5.41: Ergebnistabellen dTS-F2 und dTS-F3

werden durch Störereignisse modelliert, deren Ausprägungen erst nach dem Beginn der Auslieferung, während der Tourdurchführung bekannt werden. Da die entscheidungsrelevanten Informationen nicht zum Planungszeitpunkt zur Verfügung stehen, ist es nicht möglich ex-ante eine optimale Lösung zum dynamisch-stochastischen Tourensteuerungsproblem (dTTS) zu ermitteln. Erst in dem Moment, in dem keine Handlungsmöglichkeiten mehr bestehen, ist es theoretisch möglich, zu sagen, welche Tourzuordnung, welche Belieferungsreihenfolge und welche Routen optimal gewesen wären.

Werden die geplanten Touren und Routen durchgeführt, ohne auf Störungen zu reagieren (Handlungsalternative ALT), kommt es zu deutlichen Verspätungen und zu höheren Gesamtbediendauern als geplant. Durch eine Anpassung der Routen, unter Beibehaltung der gegebenen Belieferungsreihenfolge (Störungsreaktionsverfahren ROUTE), lässt sich bereits eine deutliche Verringerung der Verspätungen und Ist-Bediendauern erzielen. Es ist in jedem untersuchten Fall sinnvoll, die Möglichkeit Staus zu umfahren, zu berücksichtigen. Zwar können nicht immer alle Verspätungen vermieden werden, dennoch sind die verbleibenden Verspätungen stets geringer als ohne Reaktion auf Störungen.

Die zwei Störungsreaktionsverfahren PT und EX können als Reaktion auf Störereignisse neben den Routen auch die Belieferungsreihenfolge der noch nicht belieferten Kunden auf einer Tour anpassen. Je mehr Störereignisse vorliegen und je größer die Freiheitsgrade bezüglich der Belieferungsreihenfolge ist, desto vorteilhafter sind diese Störungsreaktionsverfahren. EX schneidet dabei in der Regel für die Probleminstanzen mit größeren Freiheitsgraden besser ab als PT, da mit EX die größeren Veränderungen der geplanten Touren ermöglicht werden. Ein Kundenauftrag ohne Zeitfensterrestriktionen kann beispielweise mit PT nur um jeweils eine Position auf der Tour pro Iteration verschoben werden. Mit EX ist es hingegen möglich, die Belieferung eines Kunden ohne Zeitfensterrestriktion sofort ans Ende der Tour zu verlegen und damit andere Kunden (mit Zeitfensterrestriktionen) früher (und damit ggf. noch pünktlich) zu erreichen.

Das relativ gute Abschneiden von PT lässt sich für viele der untersuchten Probleminstanzen dadurch erklären, daß auf jeder Tour nur wenige Kunden beliefert werden und die Kunden enge Zeitfensterrestriktionen aufweisen. Die Plan-Belieferungsreihenfolge auf einer Tour ist dann bereits stark durch die Zeitfenster vorgegeben. Ein paarweiser Tausch von direkt nacheinander belieferten Kunden wird dann häufiger ohne zusätzliche Verspätungen möglich sein (PT), als ein Tausch der Positionen von zwei beliebigen Aufträgen in der Tour (EX). Die EX-Nachbarschaft umfasst die vollständige PT-Nachbarschaft. Ist die gleiche Ausgangssituation gegeben, kann EX immer mindestens genauso gute Lösungen finden wie PT. Allerdings wird

für die Lösungsläufe EX mit einer first-fit Auswahlregel verwendet. Die jeweils ersten gefundenen Verbesserungen mit EX können auch weniger vorteilhaft sein als die jeweils ersten gefundenen Verbesserungen mit PT. Zudem treten in regelmäßigen Abständen neue Störereignisse auf. Wenn EX und PT in einem Reaktionszeitpunkt zu unterschiedlichen Touren gelangen, sind auch für alle späteren Reaktionszeitpunkte die Ausgangslösungen unterschiedlich. Zu späteren Reaktionszeitpunkten können dann, aufgrund von neuen Störereignissen, die relativ stärkeren Änderungen, die EX eventuell vorgenommen hat, unvorteilhaft werden, aber nicht mehr rückgängig gemacht werden.

Bei Touren mit relativ vielen Kunden, ist der Unterschied zu ROUTE besonders deutlich erkennbar. Entgegen der ersten Erwartungen ist EX nicht immer das beste Störungsreaktionsverfahren. ROUTE erweist sich immer genau dann als vorteilhaft, wenn ein Tausch der Belieferungsreihenfolge durchgeführt wird, anschließend weitere Störungen eintreffen und keine Möglichkeit mehr besteht, den (inzwischen unvorteilhaften) Tausch rückgängig zu machen.

Nicht immer ist der beste Plan-Tourenplan (mit der geringsten Anzahl an Touren und mit der geringsten Gesamtbediendauer) auch der beste Ist-Tourenplan (mit der geringsten Verspätung). Wenn man jeden Kunden einzeln auf einer Pendeltour bedienen würde, könnte man die erwarteten Verspätungen minimieren. Allerdings bleibt auch dabei noch die Möglichkeit, auf Störereignisse zu reagieren, indem die Routen zum Kunden und zurück ins Depot angepasst werden. Das Problem ist, daß für die Durchführung von Pendeltouren für jeden Kundenauftrag eine eigene Tour geplant werden müsste. Die Anzahl an Touren ist zwar für dTS nicht entscheidungsrelevant ist, für die vorangehende Lösung des dTP, und somit für die Zuordnung von Kundenaufträgen zu Touren, sind die Fahrzeugeinsatzkosten allerdings sehr wohl entscheidungsrelevant.

Eine Betrachtung der Einzelergebnisse hat gezeigt, daß gute Ausgangslösungen tendentiell auch zu geringen Ist-Bediendauern führen. Die Verspätungen sind weitgehend unabhängig von der Lösungsgüte der Ausgangslösungen, es ist nicht erkennbar, daß relativ schlechte Lösungen zu relativ geringeren oder höheren Verspätungen führen.

6 Fazit

In dieser Arbeit wurden dynamisch-deterministische Tourenplanungsprobleme und dynamisch-stochastische Tourensteuerungsprobleme auf Basis von Verkehrswegenetzwerken beschrieben, modelliert und Lösungsverfahren experimentell untersucht.

Zunächst wurde ein neues Modell zur Abbildung von dynamisch-deterministischen Tourenplanungsproblemen mit im Zeitablauf variablen Fahrzeiten auf Basis von Verkehrswegenetzwerken vorgestellt. Dabei werden die schnellsten Wege im Verkehrswegenetzwerk nicht vor Beginn der Planung festgelegt, sondern in Abhängigkeit vom Abfahrtszeitpunkt bei jedem Kunden neu ermittelt. Neben den zu beliefernden Kunden und dem Depot enthält das Verkehrswegenetzwerk zusätzliche Knoten, die Kreuzungen, die dazu dienen, alternative Wege im Zeitablauf zwischen einzelnen Kunden und zwischen den Kunden und dem Depot zu modellieren. Für die experimentelle Analyse wurde ein Netzwerkgenerator entworfen und damit neue Probleminstanzen zu dynamisch-deterministischen Tourenplanungsproblemen auf Basis von Netzwerken mit den wesentlichen Eigenschaften von realen Verkehrswegenetzwerken generiert.

Ein neuer Ameisenkoloniealgorithmus (dynACO) zur Lösung von dynamisch-deterministischen Tourenplanungsproblemen mit Zeitfensterrestriktionen wurde entworfen und beschrieben. Er unterscheidet sich von herkömmlichen Ameisenkoloniealgorithmen unter anderem dadurch, daß das Pheromonupdate multiplikativ und nicht additiv erfolgt. Auf Basis der neuen Probleminstanzen zu dynamisch-deterministischen Tourenplanungsproblemen wurden vier heuristische Lösungsverfahren aus der Literatur (ein Verfahren des besten Nachfolgers, eine Einfügeheuristik, das Cross-Exchange-Verbesserungsverfahren und ein Ameisenkoloniealgorithmus) mit dem neuen Ameisenkoloniealgorithmus (dynACO) experimentell verglichen. Dabei hat sich gezeigt, daß dynACO besser als alle anderen untersuchten Verfahren dazu geeignet ist, die generierten Probleminstanzen zu dynamisch-deterministischen Tourenplanungsproblemen zu lösen.

Aufbauend auf dem Tourenplanungsproblem wurde in dieser Arbeit erstmals das Tourensteuerungsproblem formuliert. Stochastische Fahrzeiten, deren Ausprägungen im Zeitablauf erst bekannt werden, werden durch Störereignisse abgebildet. Prognosen

über unterschiedliche Fahrzeiten in Abhängigkeit von der Tageszeit werden bereits in die Planung einbezogen, während Störereignisse nur die von der Planung abweichenden Ausprägungen der Fahrzeiten berücksichtigen. Die Tourensteuerung setzt in dem Moment an, in dem die Tourenplanung abgeschlossen wurde und die Transportvorgänge beginnen. Als Reaktion auf Störereignisse können die geplanten Touren und Routen noch während der Auslieferung von Gütern angepasst werden. Vier Störungsreaktionsverfahren wurden zu diesem Zweck untersucht, darunter zwei Verfahren, die die Belieferungsreihenfolge der noch zu beliefernden Kunden anpassen. Es zeigt sich, daß die Ergebnisse der Tourenplanung bei starken und zahlreichen Störungen nur eine grobe Approximation für die tatsächlich benötigten Auslieferungszeiten geben.

Für eine praktische Umsetzung der entwickelten Störungsreaktionsverfahren existieren inzwischen alle wesentlichen technischen Voraussetzungen. Aktuelle Daten über die aktuelle Verkehrssituation auf den entscheidungsrelevanten Straßen sind erhältlich, genauso wie Prognosen über Beeinträchtigungen der Verkehrswegenetzwerke. Für den praktischen Einsatz sind allerdings deutlich bessere und schnelle Verfahren zur Ermittlung oder zur geeigneten Approximation von schnellsten Wegen im Verkehrswegenetzwerk erforderlich, damit die Rechenzeiten für reale Problemstellungen handhabbar bleiben. Alternativ ist auch denkbar, bei den Lösungsalgorithmen die Suche nach einer vorgegebenen Zeitspanne abubrechen und die bis dahin beste bekannte Lösung zu verwenden. In jedem Fall sind vor einer praktischen Umsetzung weitergehende experimentelle Analysen unerlässlich, um eine für den konkreten Einsatzzweck geeignete Parameterkonstellation zu ermitteln.

Literaturverzeichnis

Ahn und Shin (1991):

Ahn, B.-H.; Shin, J.-Y.: Vehicle-routing with Time Windows and Time-varying Congestion. In: Journal of the Operational Research Society 42 (1991), Nr. 5, S. 393-400

Antes und Derigs (1996):

Antes, J.; Derigs, U.: A New Parallel Tour Construction Algorithm for the Vehicle Routing Problem with Time Windows. In: Kleinschmidt, P.; Bachem, A.; Derigs, U.; Fischer, D.; Leopold-Wildburger, U.; Möhring, R. (Hrsg.): Operations Research Proceedings 1995, Springer, Berlin u.a. (1996), S. 400-404

Badeau et al. (1997):

Badeau, P.; Guertin, F.; Gendreau, M.; Potvin, J.-Y.; Taillard, É.: A parallel tabu search heuristic for the vehicle routing problem with time windows. In: Transportation Research 3/5 (1997), No. 2, S. 109-122

Balakrishnan (1993):

Balakrishnan, N.: Simple Heuristics for the Vehicle Routing Problem with Soft Time Windows. In: Journal of the Operational Research Society 44 (1993), Nr. 3, S. 279-287

Bent und Van Hentenryck (2004):

Bent, R.; Van Hentenryck, P.: A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows. In: Transportation Science 38 (2004), Nr. 4, S. 515-530

Berger et al. (2001):

Berger, J.; Barkaoui, M.; Bräysy, O.: A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows. In: Defense Research Establishment, Valcartier, Canada (2001) - Working Paper

Bertsimas und van Ryzin (1991):

Bertsimas, D. J.; van Ryzin, G.: A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane. In: Operations Research 39 (1991), Nr. 4, S. 601-615

Bock (2004):

Bock, S.: Echtzeitfähige Steuerung von Speditionsnetzwerken. Dt. Univ.-Verl., Gabler, Wiesbaden, 2004. - Zugl.: Universität Paderborn, Habilitationsschrift, 2003, u.d.T.: Echtzeitfähige Steuerung von Transportnetzwerken

Bräysy und Gendreau (2005a):

Bräysy, O.; Gendreau, M.: Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. In: *Transportation Science* 39 (2005), Nr. 1, S. 104-118

Bräysy und Gendreau (2005b):

Bräysy, O.; Gendreau, M.: Vehicle Routing Problem with Time Windows, Part II: Metaheuristics. In: *Transportation Science* 39 (2005), Nr. 1, S. 119-139

Chiang und Russell (2004):

Chiang, W. C.; Russell, R. A.: A metaheuristic for the vehicle-routing problem with soft time windows. In: *Journal of the Operational Research Society* 55 (2004), S. 1298-1310

Chiang und Russell (1996):

Chiang, W. C.; Russell, R. A.: Simulated annealing metaheuristics for the vehicle routing problem with time windows. In: *Annals of Operations Research* 63 (1996), S. 3-27

Chiang und Russell (1997):

Chiang, W. C.; Russell, R. A.: A Reactive Tabu Search Metaheuristic for the Vehicle Routing Problem with Time Windows. In: *INFORMS Journal on Computing* 9 (1997), Nr. 4, S. 417-430

Clarke und Wright (1964):

Clarke, G.; Wright, J. W.: Scheduling Of Vehicles From A Central Depot To A Number Of Delivery Points. In: *Operations Research* 12 (1964), S. 568-574

Cook und Rich (1999):

Cook, W.; Rich, J. L.: A parallel cutting-plane algorithm for the vehicle routing problem with time windows. Tech. Report. Computational and Applied Mathematics, Rice University, Houston (1999)

Cordeau et al. (2001):

Cordeau, J.-F.; Laporte, G.; Mercier, A.: A unified tabu search heuristic for vehicle routing problems with time windows. In: *Journal of the Operational Research Society* 52 (2001), S. 928-936

Czech und Czarnas (2002):

Czech, Z. J.; Czarnas, P.: A Parallel Simulated Annealing for the Vehicle Routing Problem with Time Windows. In: *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, IEEE Computer Society, Canary Islands (2002), S. 376-383

Dantzig und Ramser (1959):

Dantzig, G. B.; Ramser, J. H.: The Truck Dispatching Problem. In: *Management Science* 6 (1959), S. 80-91

de Backer et al. (2000):

de Backer, B.; Furnon, V.; Shaw, P.; Kilby, P.; Prosser, P.: Solving Vehicle Routing Problems Using Constraint Programming and Metaheuristics. In: Journal of Heuristics 6 (2000), S. 501-523

Desrochers et al. (1992):

Desrochers, M.; Desrosiers, J.; Solomon, M.: A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows. In: Operations Research 40 (1992), Nr. 2, S. 342-354

Dijkstra (1959):

Dijkstra, E. W.: A Note on Two Problems in Connexion with Graphs. In: Numerische Mathematik 1 (1959), S. 269-271

Domschke (1997):

Domschke, W.: Logistik: Rundreisen und Touren. 4. Auflage. R. Oldenbourg Verlag, München, Wien (1997)

Donati et al. (2003a):

Donati, A. V.; Montemanni, R.; Casagrande, N.; Rizzoli, A. E.; Gambardella, L. M.: Time Dependent Vehicle Routing Problem with a Multi Ant Colony System. Techn. Report IDSIA-17-03, Lugano (2003)

Donati et al. (2003b):

Donati, A. V.; Montemanni, R.; Gambardella, L. M.; Rizzoli, A. E.: Integration of a Robust Shortest Path Algorithm with a Time Dependent Vehicle Routing Model and Applications. CIMSIA 2003 - International Symposium on Computational Intelligence for Measurement Systems and Applications, Lugano (2003)

Dorigo (1992):

Dorigo, M.: Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale (Optimization, learning and natural algorithms). Dipartimento di Elettronica, Politecnico di Milano, Italien, Dissertation, 1992 - italienisch

Dorigo und Di Caro (1999):

Dorigo, M.; Di Caro, G.: The Ant Colony Metaheuristic. In: Corne, D.; Dorigo, M.; Glover, F. (Hrsg.): New Ideas in Optimization. McGrawHill, London, (1999), S. 11-32

Dorigo und Gambardella (1999):

Dorigo, M., Gambardella, L. M.: Ant Algorithms for Discrete Optimization. In: Artificial Life 5 (1999), Nr. 3, S. 137-172

Dorigo und Stützle (2004):

Dorigo, M.; Stützle, T.: Ant Colony Optimization. The MIT Press, London, England, Cambridge, Massachusetts (2004)

Dror und Powell (1993):

Dror, M.; Powell, W.: Stochastic and Dynamic Models in Transportation. In: Operations Research 41 (1993), Nr. 1, S. 11-14

Dullaert et al. (2002):

Dullaert, W.; Janssens, G. K.; Sörensen, K.; Vernimmen, B.: New heuristics for the mixed Fleet Size and Mix Vehicle Routing Problem with Time Windows. In: Journal of the Operational Research Society 53 (2002), S. 1232-1238

Fisher et al. (1997):

Fisher, M. L.; Jörnsten, K. O.; Madsen, O. B. G.: Vehicle Routing with Time Windows: Two Optimization Algorithms. In: Operations Research 45 (1997), Nr. 3, S. 488-492

Fleischmann (1997):

Fleischmann, B.: Logistik: Gestaltung von Logistiksystemen. 2. überarbeitete und erweiterte Auflage. Verlag Moderne Industrie, Landsberg / Lech (1997)

Fleischmann et al. (2004):

Fleischmann, B.; Gietz, M.; Gnutzmann, S.: Time-Varying Travel Times in Vehicle Routing. In: Transportation Science 38 (2004), Nr. 2, S. 160-173

Floyd (1962):

Floyd, R. W.: Algorithm 97 Shortest Path. In: Communications of the ACM 5 (1962), Nr. 1, S. 345

Fu et al. (2005):

Fu, Z.; Eglese, R.; Li, L. Y. O.: A tabu search heuristic for the open vehicle routing problem. In: Journal of the Operational Research Society 56 (2005), Nr. 3, S. 267-274

Gambardella et al. (1999):

Gambardella, L. M.; Taillard, É.; Agazzi, G.: MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In: Corne, D.; Dorigo, M.; Glover, F. (Hrsg.): New Ideas in Optimization. McGrawHill, London, (1999), S. 63-76

Gietz (1994):

Gietz, M.: Computergestützte Tourenplanung mit zeitkritischen Restriktionen. Physica Verlag, Heidelberg, 1994. - Zugl.: Universität Augsburg, Dissertation, 1993

Homberger (2000):

Homberger, J.: Verteilt-parallele Metaheuristiken zur Tourenplanung. Dt. Univ.-Verl., Gabler, Wiesbaden, 2000. - Zugl.: FernUniversität-Gesamthochschule-Hagen, Dissertation, 2000

Homberger und Gehring (1999a):

Homberger, J.; Gehring, H.: Ein Simulated-Annealing-Verfahren für das Standardproblem der Tourenplanung mit Zeitfensterrestriktionen. In: Kall, P.; Lüthi, H.-J. (Hrsg.): Operations Research Proceedings 1998, Springer, Berlin u.a. (1999), S. 483-492

Homberger und Gehring (1999b):

Homberger, J.; Gehring, H.: Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows. In: INFOR 37 (1999), S. 297-318

Homberger und Gehring (2005):

Homberger, J.; Gehring, H.: A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. In: European Journal of Operational Research 162 (2005), S. 220-238

Ibaraki et al. (2001):

Ibaraki, T.; Kubo, M.; Masuda, T.; Uno, T.; Yagiura, M.: Effective Local Search Algorithms for the Vehicle Routing Problem with General Time Window Constraints. In: MIC'2001 - 4th Metaheuristics International Conference July 16-20, Porto (2001), S. 293-297

Ichoua et al. (2000):

Ichoua, S.; Gendreau, M.; Potvin, J.-Y.: Diversion Issues in Real-Time Vehicle Dispatching. In: Transportation Science 34 (2000), Nr. 4, S. 426-438

Ichoua et al. (2003):

Ichoua, S.; Gendreau, M.; Potvin, J.-Y.: Vehicle Dispatching with Time-dependent Travel Times. In: European Journal of Operational Research 144 (2003), S. 379-396

Ioannou et al. (2001):

Ioannou, G.; Kritikos, M.; Prastacos, G.: A greedy look-ahead heuristic for the vehicle routing problem with time windows. In: Journal of the Operational Research Society 52 (2001), S. 523-537

Keynon und Morton (2003):

Keynon, A. S.; Morton, D. P.: Stochastic Vehicle Routing with Random Travel Times. In: Transportation Science 37 (2003), Nr. 1, S. 69-82

Kilby et al. (1997):

Kilby, P.; Prosser, P.; Shaw, P.: Guided Local Search for the Vehicle Routing Problem. In: 2nd International Conference on Metaheuristics - MIC97, Sophia-Antipolis, France (1997), S. 1-10

Kohl und Madsen (1997):

Kohl, N.; Madsen, O. B. G.: An Optimization Algorithm for the Vehicle Routing Problem with Time Windows based on Lagrangian Relaxation. In: *Operations Research* 45 (1997), Nr. 3, S. 395-406

Kruskal (1956):

Kruskal, J. B.: On the shortest spanning subtree of a graph and the travelling salesman problem. In: *Proceedings of the American Mathematical Society* 7 (1956), S. 48-50

Larsen et al. (2002):

Larsen, A.; Madsen, O.; Solomon, M.: Partially dynamic vehicle routing - models and algorithms. In: *Journal of the Operational Research Society* 53 (2002), S. 637-646

Larsen (2000):

Larsen, A.: The Dynamic Vehicle Routing Problem. Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Dissertation, 2000 (IMM-PHD-2000-73)

Lau et al. (2003):

Lau, H. C.; Sim, M.; Teo, K. M.: Vehicle routing problem with time windows and a limited number of vehicles. In: *European Journal of Operational Research* 148 (2003), S. 559-569

Le Bouthillier und Crainic (2005):

Le Bouthillier, A.; Crainic, T. G.: A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. In: *Computers & Operations Research* 32 (2005), S. 1685-1708

Lenstra und Rinnooy Kan (1981):

Lenstra, J. K.; Rinnooy Kan, A. H. G.: Complexity of Vehicle Routing and Scheduling Problems. In: *Networks* 11 (1981), Nr. 22, S. 12-27

Li und Lim (2003):

Li, H.; Lim, A.: Local search with annealing-like restarts to solve the VRPTW. In: *European Journal of Operational Research* 150 (2003), S. 115-127

Liu und Shen (1999):

Liu, F.-H. F.; Shen, S.-Y.: A route-neighborhood-based metaheuristic for vehicle routing problem with time windows. In: *European Journal of Operational Research* 118 (1999), S. 485-504

Miller-Hooks und Mahmassani (2000):

Miller-Hooks, E. D.; Mahmassani, H. S.: Least Expected Time Paths in Stochastic, Time-Varying Transportation Networks. In: *Transportation Science* 34 (2000), Nr. 2, S. 198-215

Nagy und Salhi (2005):

Nagy, G.; Salhi, S.: Heuristic algorithms for single and multiple depot vehicle routing problems with pickup and deliveries. In: *European Journal of Operational Research* 162 (2005), S. 126-141

Ombuki et al. (2002):

Ombuki, B. M.; Nakamura, M.; Osamu, M.: A Hybrid Search Based on Genetic Algorithms and Tabu Search for Vehicle Routing. Tech. Report. Department of Computer Science, Brock University, St. Catharines, Ontario, Canada (2002)

Potvin und Bengio (1996):

Potvin, J.-Y.; Bengio, S.: The Vehicle Routing Problem with Time Windows - Part II Genetic Search. In: *INFORMS Journal on Computing* 8 (1996), Nr. 2, S. 165-172

Potvin et al. (1996):

Potvin, J.-Y.; Kervahut, T.; Garcia, B.-L.; Rousseau, J.-M.: The Vehicle Routing Problem with Time Windows - Part I Tabu Search. In: *INFORMS Journal on Computing* 8 (1996), Nr. 2, S. 158-164

Potvin und Rousseau (1993):

Potvin, J.-Y.; Rousseau, J.-M.: A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. In: *European Journal of Operational Research* 66 (1993), S. 331-340

Potvin und Rousseau (1995):

Potvin, J.-Y.; Rousseau, J.-M.: An Exchange Heuristic for Routeing Problems with Time Windows. In: *Journal of the Operational Research Society* 46 (1995), S. 1433-1446

Rochat und Taillard (1995):

Rochat, Y.; Taillard, É. D.: Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. In: *Journal of Heuristics* 1 (1995), S. 147-167

Russell (1995):

Russell, R. A.: Hybrid Heuristics for the Vehicle Routing Problem with Time Windows. In: *Transportation Science* 29 (1995), Nr. 2, S. 156-166

Sariklis und Powell (2000):

Sariklis, D.; Powell, S.: A heuristic method for the open vehicle routing problem. In: *Journal of the Operational Research Society* 51 (2000), S. 564-573

Savelsbergh (1992):

Savelsbergh, M. W. P.: The Vehicle Routing Problem with Time Windows: Minimizing Route Duration. In: *ORSA Journal on Computing* 4 (1992), Nr. 2, S. 146-154

Schrimpf et al. (2000):

Schrimpf, G.; Schneider, J.; Stamm-Wilbrandt, H.; Dueck, G.: Record Breaking Optimization Results Using the Ruin and Recreate Principle. In: *Journal of Computational Physics* 159 (2000), S. 139-171

Schulze und Fahle (1999):

Schulze, J.; Fahle, T.: A parallel algorithm for the vehicle routing problem with time window constraints. In: *Annals of Operations Research* 86 (1999), S. 585-607

Shaw (1998):

Shaw, P.: Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In: Mahler, M. und Puget, J.-F. (Hrsg.): *Proceedings: Principles and Practice of Constraint Programming*, Springer, Berlin u.a. (1998), S. 417-431

Shaw (1997):

Shaw, P.: A new local search algorithm providing high quality solutions to vehicle routing problems. In: *University of Strathclyde, Glasgow, Scotland (1997) - Working Paper*

Solomon (1987):

Solomon, M.: Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints. In: *Operations Research* 35 (1987) Nr. 2, S. 254-265

Stützle (1998):

Stützle, T. G.: *Local Search Algorithms for Combinatorial Problems - Analysis, Improvements, and New Applications*. TU Darmstadt, Dissertation, 1998

Taillard et al. (1997):

Taillard, É.; Badeau, P.; Gendreau, M.; Guertin, F.; Potvin, J.-Y.: A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. In: *Transportation Science* 31 (1997), Nr. 2, S. 170-186

Taillard et al. (1995):

Taillard, É. ; Laporte, G.; Gendreau, M.: *Vehicle Routing with Multiple Use of Vehicles*. Tech. Report CRT-95-19. Centre de recherches sur les transports, Montreal (1995)

Tarantilis et al. (2003):

Tarantilis, C. D.; Kiranoudis, C. T.; Vassiliadis, V. S.: A list based threshold accepting metaheuristic for the heterogeneous mixed fleet vehicle routing problem. In: *Journal of the Operational Research Society* 54 (2003), S. 65-71

van Landeghem (1988):

van Landeghem, H. R. G.: A bi-criteria heuristic for the vehicle routing problem with time windows. In: *European Journal of Operational Research* 36 (1988), S. 217-226

Wendt (1995):

Wendt, O.: Tourenplanung durch Einsatz naturanaloger Verfahren: Integration von Genetischen Algorithmen und Simulated Annealing. Dt. Univ.-Verl., Gabler, Wiesbaden, 1995. - Zugl.: Universität Frankfurt (Main), Dissertation, 1994, u.d.T.: Naturanaloge Verfahren zur approximativen Lösung kombinatorischer Optimierungsprobleme

Zhan und Noon (1998):

Zhan, F. B.; Noon, C. E.: Shortest Path Algorithms: An Evaluation using Real Road Networks. In: Transportation Science 32 (1998) Nr. 1, S. 65-73