

Diplomarbeit

Integration eines Auftragsvermittlers in ein Agentensystem zur Transportsteuerung

Eingereicht von

Thomas Hirsch, 21706

Erstgutachter

Prof. Dr. Ralf Möller

Zweitgutachter

Prof. Dr. Thorsten Blecker

Eidesstaatliche Erklärung

Ich versichere ehrenwörtlich, dass ich die vorliegende Diplomarbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat. Alle Ausführungen der Arbeit, die wörtlich oder sinngemäß übernommen wurden, sind als solche kenntlich gemacht.

Hamburg, den

Inhaltsverzeichnis

| | | |
|-------|--|----|
| 1 | Einleitung..... | 4 |
| 1.1 | Problemstellung und Motivation..... | 4 |
| 1.1.1 | Transport-Szenario..... | 4 |
| 1.1.2 | Motivation der Akteure..... | 6 |
| 1.2 | Lösungsansatz..... | 8 |
| 2 | Theoretische Grundlagen..... | 11 |
| 2.1 | Reverse-Vickrey-Auktion..... | 11 |
| 2.2 | Routenberechnung..... | 13 |
| 2.2.1 | Die Klasse Route-Node..... | 14 |
| 2.2.2 | Verwendung von Route-Nodes..... | 16 |
| 2.3 | Kostenfunktion und Gebotsberechnung..... | 20 |
| 2.3.1 | Weg- und zeitabhängige Kostenfaktoren..... | 20 |
| 2.3.2 | Kostenfunktion..... | 22 |
| 2.3.3 | Gebotsfunktion..... | 23 |
| 3 | Modellierung der Agenten..... | 27 |
| 3.1 | Container-Agent..... | 27 |
| 3.2 | Transport-Agent..... | 29 |
| 3.2.1 | Attribute des Transportagenten..... | 29 |
| 3.2.2 | Durchführbarkeit von Aufträgen..... | 30 |
| 3.2.3 | Gebotsabgabe..... | 31 |
| 3.2.4 | Planung von Aufträgen..... | 33 |
| 3.2.5 | Implikationen kurzer Auktionsdauer..... | 33 |
| 3.2.6 | Responder-Protokoll..... | 34 |
| 3.3 | Auktionator-Agent..... | 36 |
| 3.3.1 | Container-Vorrat..... | 36 |
| 3.3.2 | Registrierung der Transporter..... | 37 |
| 3.3.3 | Initiator-Protokoll..... | 37 |
| 3.3.4 | Erhebung einer Gebühr..... | 39 |
| 4 | Zusammenfassung und Ergebnisse..... | 41 |
| 4.1 | Entwicklung des Auktionsprotokolls..... | 41 |
| 4.2 | Entwicklung der Agenten..... | 42 |
| 4.3 | Qualitative Ergebnisse..... | 43 |
| 5 | Ausblick..... | 45 |
| 5.1 | Absprachen der Transporter..... | 45 |
| 5.2 | Sinnvolle Gebote und Auftragsdaten..... | 46 |
| 5.3 | Erweiterbarkeit des Systems..... | 46 |
| 5.4 | Transparenz..... | 47 |
| | Literaturverzeichnis..... | 48 |

1 Einleitung

1.1 *Problemstellung und Motivation*

In dieser Diplomarbeit wird ein Mechanismus zur automatisierten Auftragsvergabe in einem Logistik-Szenario entwickelt. Das Problem besteht darin eine Plattform zu entwerfen, die Transporteure und Auftraggeber zusammenbringt, und von deren Nutzung sich beide einen Vorteil versprechen können. Es soll gezeigt werden, welche Motivationen die einzelnen Teilnehmer des System haben, und wie diese erfüllt werden können. Das Verfahren soll stabil sein und damit soll keine (bzw. geringe) Möglichkeit und Motivation zur Manipulation des Systems durch seine Teilnehmer vorhanden sein.

1.1.1 **Transport-Szenario**

Das grundlegende Szenario stammt aus [Rosenfeld 2008]. Für diese Arbeit wurde es dahingehend modifiziert, daß nur der Transport von Containern, beispielsweise 40-Fuß-ISO-Containern, durch entsprechende Sattelzüge betrachtet wird. Das hat zur Folge, daß die Kapazität der Transporter auf einen Container beschränkt ist, und Transportaufträge dadurch nur sequentiell angenommen und abgearbeitet werden.

Innerhalb des Szenarios lassen sich zunächst zwei Akteure beziehungsweise Rollen identifizieren. Zum einen gibt es den Auftraggeber, der durch einen Container bzw. dessen Inhalt repräsentiert ist. Der zweite Akteur ist der (bzw. die) Spediteur(e). Diese repräsentieren einen oder mehrere Lastwagen, durch die sie ihr Transportangebot bereitstellen.

Zusätzlich zu diesen beiden soll nun ein weiterer Akteur in das System eingeführt werden. Dabei handelt es sich um einen Vermittler, der mit den Auftraggebern und den Transporteuren kommuniziert, um Transportaufträge zu vermitteln.

Ein Transportauftrag innerhalb des Szenarios ist gekennzeichnet durch zwei Zeitfenster, die jeweils die frühesten und spätesten Zeitpunkte zum Abholen bzw. Anliefern des Containers darstellen, sowie dem Aufenthaltsort des Containers und seinem Zielort. Ein weiterer Parameter sind die Kosten bzw. der Preis für den Transport.

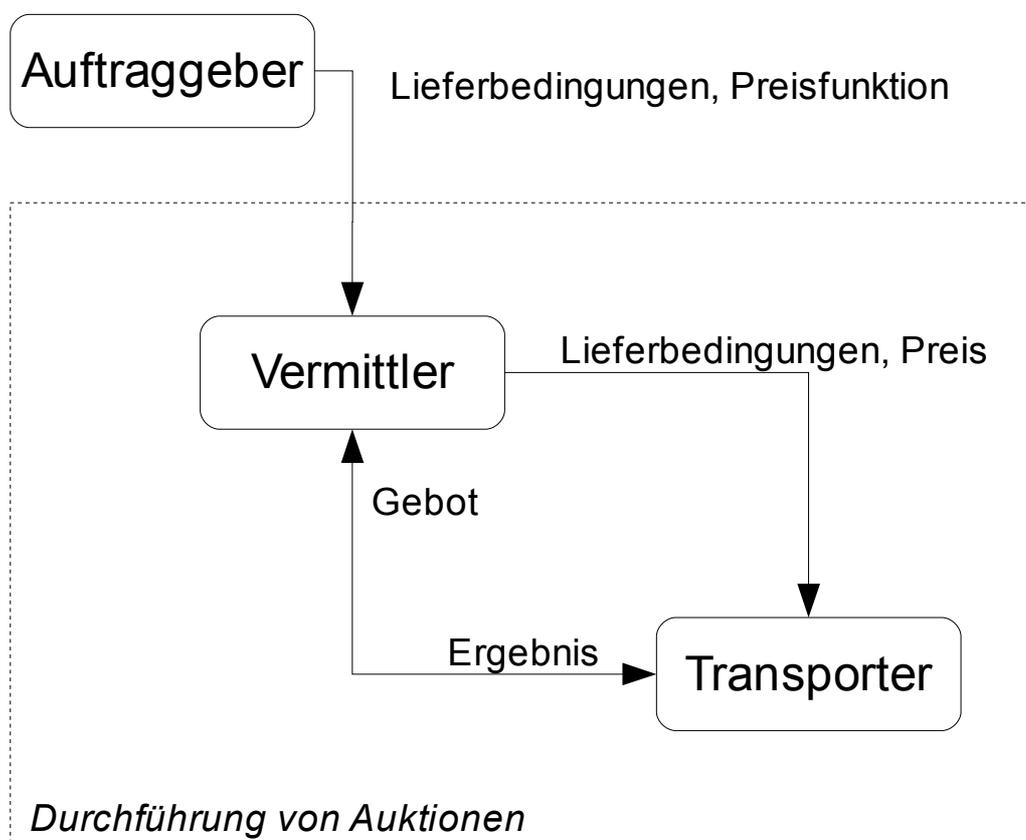


Abbildung 1.1: Akteure im Transport-Szenario

1.1.2 Motivation der Akteure

Wesentlich für die Realisierbarkeit der Auftragsvermittlung ist die individuelle Motivation jedes einzelnen Akteurs innerhalb des System. Die identifizierten Akteure sind Auftraggeber, Transporter und Vermittler. Jeder von ihnen muß motiviert sein, an dem zu entwickelnden System teilzunehmen. Diese Motivation besteht im Rahmen der das System umgebenden Marktwirtschaft im Streben nach größtmöglichem, nichtnegativen monetären Gewinn. Außerdem kann davon ausgegangen werden, daß durch gesteigerte Verfügbarkeit von sowohl Transportangeboten als auch Transportaufträgen, im Vergleich zur Auftragsbeschaffung/-vermittlung durch traditionelle Kommunikationswege, eine grundlegende Motivation aller Teilnehmer besteht, das nach den Vorgaben dieser Arbeit entwickelte System zu nutzen.

Bevor in späteren Kapiteln genauer darauf eingegangen wird, sollen hier zunächst einmal grundsätzliche Annahmen und Feststellungen bezüglich der einzelnen Akteure gemacht werden.

Auftraggeber

Da die Auftraggeber innerhalb des Systems durch ihre Aufträge, und damit letztendlich durch den jeweiligen zu transportierenden Container repräsentiert sind, können Container und Auftraggeber als synonym betrachtet werden.

Ein Container hat das Ziel, innerhalb eines gegebenen Zeitraumes zu seinem Zielort transportiert zu werden. Um dieses Ziel zu erreichen, ist er gezwungen einen gewissen Geldbetrag zu investieren. Ohne den Grund des Transports oder den Inhalt des Containers zu kennen, muß sinnvollerweise angenommen werden, daß der Container eine obere Grenze für den zu investierenden Betrag hat. Die Zielsetzung beinhaltet also das Nichtüberschreiten eines maximalen Preises für den Transport.

Der maximale Preis wird sich für den Container im wesentlichen aus zwei Faktoren ergeben. Zum einen aus den Kosten für den Transport, zum andern aus seiner Dringlichkeit, also der noch verbleibenden Zeit, bis das späteste Lieferdatum zwangsläufig überschritten würde. Der Nutzen seitens des Containers ergibt sich aus seiner Zielerfüllung im Verhältnis zum investierten Betrag ergibt. Daher kann angenommen werden, daß für Zeitpunkte, die verhältnismäßig weit vor dem Datum spätester Ablieferung liegen, der Container weniger bereit ist, den für ihn maximalen Betrag zu zahlen, als zu späterer Zeit,

wenn das Lieferdatum näher bevorsteht.

Die Motivation ist für den Auftraggeber in dem Moment erfüllt, in dem ihm ein Transporter vermittelt wird, der den Container zu den gegebenen Bedingungen befördert, und dabei den vorgegebenen maximalen Preis nicht überschreitet.

Transporter

Der Transporter repräsentiert das Geschäftsmodell, Container entgeltlich zu befördern. Seine Motivation besteht darin, einen Auftrag anzunehmen, den er in der Lage ist zu erfüllen, und für den er als Bezahlung einen Betrag erhält, der mindestens die ihm entstehenden Kosten deckt.

Damit der Transporter beurteilen kann, ob sich ein gegebener Auftrag für ihn lohnt, muß er in der Lage sein, seine Kosten zu schätzen. Aus den individuellen Kosten ergibt sich der Mindestbetrag, den der Transporter fordern muß. Je nach Erwartungshaltung des Transporteurs wird sich der Mindestbetrag mehr oder weniger weit oberhalb der geschätzten Kosten befinden.

Die Motivation des Transporters ist in dem Moment erfüllt, in dem ihm ein Auftrag vermittelt wird, für dessen Ausführung er einen Betrag erhält, der größer oder gleich dem von ihm geforderten Mindestbetrag für diesen Auftrag ist.

Vermittler

Der Vermittler repräsentiert ebenfalls ein Geschäftsmodell. Seine Motivation ist es, Transportaufträge gegen eine Gebühr an Spediteure zu vermitteln. Er stellt eine Infrastruktur bereit, die die anderen Akteure nutzen können um ihre Ziele zu erfüllen. Zu seiner eigenen Unkostendeckung muß er dafür einen Betrag erhalten, der sich im einfachsten Fall aus der Kostenvorgabe des Auftraggebers ergibt. An späterer Stelle werden verschiedene Berechnungsmethoden mit Vor- und Nachteilen diskutiert.

Für den Vermittler ist die Motivation erfüllt, sofern sich Container und Transporter über die von ihm bereitgestellte Kommunikationsplattform zusammenfinden und einen Vertrag abschließen, der beinhaltet, daß an den Vermittler die von ihm geforderte Gebühr gezahlt wird.

Das Ziel dieser Arbeit ist es, ein System bzw. einen Mechanismus zu entwickeln, der gewährleistet, daß alle individuellen Motivationen der einzelnen Teilnehmer erfüllt werden können. Noch einmal zusammengefaßt bedeutet das:

- Container gibt Beförderungparameter und Maximalpreis vor, das System garantiert, daß dieser Preis eingehalten oder unterschritten wird.
- Transporter gibt Minimalpreis für einen gegebenen Auftrag vor, das System soll garantieren, daß dieser Preis eingehalten oder überschritten wird.
- Vermittler stellt eine Verhandlungsplattform bereit und wird für den Vertragsabschluß zwischen Container und Transporter bezahlt.

1.2 Lösungsansatz

Das Transportszenario aus [Rosenfeld 2008] wurde durch ein Agentensystem realisiert. Da Softwareagenten zahlreiche Eigenschaften besitzen, die eine Umsetzung des Szenarios bzw. die Entwicklung eines vermittelnden Mechanismus begünstigen, wird im Folgenden von ihrer Verwendung ausgegangen. Das bedeutet, daß jede Rolle innerhalb des Systems durch einen Typ von Agent modelliert werden wird. Für die Vorteile und zu berücksichtigenden Kriterien beim Entwurf und Einsatz von Multi-Agentensystemen siehe beispielsweise [Hirsch 2007] oder [Rosenfeld 2008].

Die wesentliche Grundidee ist die Verwirklichung der Anforderungen durch einen auktionsbasierten Mechanismus. Wie in [Rosenfeld 2008] angedacht, ist das Prinzip der Vickrey-Auktion dafür geeignet. Genauer handelt es sich beim im Rahmen dieser Arbeit entwickelten Mechanismus um eine Variante der Reverse-Vickrey-Auktion, einer Zweitpreis-Auktionsform (engl. *Second Price Sealed Bid Reverse Auction*).

Wesentliches Bedingung für die praktische Anwendbarkeit eines nach den Vorgaben dieser Arbeit entwickelten Systems ist die Robustheit bzw. Stabilität des Mechanismus. Dies beinhaltet unter anderem die Existenz einer dominanten Strategie. Im Falle der Vickrey-Auktion besteht diese in wahrheitsgemäßem

Bieten. Wenngleich, wie in [Seifert/Strecker 2003] angedeutet, generell eine Tendenz zu unwahrheitsgemäßem Bieten besteht, kann diese in einer realen Anwendungsumgebung vernachlässigt werden. An später Stelle wird darauf näher eingegangen.

Ein weiteres entscheidendes Kriterium ist die Transparenz des Mechanismus. Es ist naheliegend, daß ein hoher Grad an Transparenz die Motivation der Teilnehmer begünstigt. Dabei muß natürlich sichergestellt sein, daß diese Transparenz mit dem Mechanismus vereinbar ist, unter anderem da es sich beim vorgeschlagenen Mechanismus um eine Geheimgebots-Auktion handelt.

Ein Beispiel für die Notwendigkeit von Transparenz besteht in der Erhebung der Gebühr durch den Auktionator. Das Zustandekommen des Betrages muß im Nachhinein für alle Teilnehmer offensichtlich sein. Außerdem ist es wünschenswert, daß der Mechanismus das Interesse des Auktionators eliminiert, die Auktion zwecks Selbstbereicherung zu manipulieren (beispielsweise durch das Fälschen oder Erzeugen von Geboten).

Weiterhin muß der Mechanismus gewährleisten, daß sowohl Transporter als auch Container zeitnahe Rückmeldung über den Ausgang einer Auktion erhalten. Das bedeutet, daß eine einzelne Auktion zwangsläufig nicht von langer Dauer sein darf. Dieses Kriterium wird offensichtlich wenn man die Bedeutung der Vorkenntnis über auftragsrelevante Zeitpunkte und Zeitfenster in der Logistikbranche betrachtet. Eine Implikation langer Auktionsdauer ist für den Transport-Agenten beispielsweise das Problem, daß er bis zu Kenntnis des Auktionsergebnisses keine weitere Planung vornehmen kann. Planung bedeutet in diesem Fall unter anderem auch die Teilnahme an weiteren Auktionen, beziehungsweise würde die Teilnahme und somit das Abgeben von Geboten schwieriger, wenn das Ergebnis einer laufenden Auktion noch nicht feststeht.

Für den Container-Agenten bedeutet eine lange Auktionsdauer wiederum einen längeren Zeitraum, in dem seinerseits keine Einflußnahme auf den Preis stattfinden kann. Da gemäß den Anforderungen der Container die Möglichkeit haben muß, seinen Maximalpreis anzupassen, so ist ihm dies aufgrund der Vickrey-Auktionsform nur vor der Initiierung der Auktion möglich. Dies impliziert gegebenenfalls mehrfache Versteigerungen mit veränderter Preisvorgabe durch den Container.

Die wesentlichen Ansätze sind zusammengefaßt:

- Modellierung des Systems mit Transport-Agenten, Container-Agenten, und einem Auktionator-Agenten
- Auktionator versteigert Container in Reverse-Vickrey-Auktionen.
- Transporter nehmen an Auktionen teil und bieten auf Container (Transportaufträge).
- Auktionsdauer soll so kurz wie möglich gehalten werden.
- Auktionator erhebt Gebühr bei erfolgreicher Auktion.
- Container kann vor oder nach einer Auktion seinen Preis anpassen um Einfluß auf die Wahrscheinlichkeit einer erfolgreichen Versteigerung zu haben.

2 Theoretische Grundlagen

In diesem Kapitel werden die Voraussetzungen für die Entwicklung der Agenten vermittelt. Dabei werden Ergebnisse anderer Arbeiten verwendet, insbesondere aus [Rosenfeld 2008] (Entscheidung für die Vickrey-Auktion, Ansatz der Routenberechnung).

In dieser Arbeit entwickelte Grundlagen sind insbesondere die Formulierung der Kostenfunktion und Gebotsfunktion, und die auf den Routen basierte Kostenabrechnung und die Preisfunktion des Containers. Die in Abschnitt 2.2 angegebenen (Pseudo-)Code-Beispiele sind insofern als Grundlage zu verstehen, als daß sie ein relativ einfaches Konzept zur Erzeugung der notwendigen Datenstrukturen darstellen. Es kommt nicht darauf an, wie im Detail diese Algorithmen implementiert werden, für besseres Verständnis ist es jedoch vorteilhaft, das Prinzip der Routenerstellung beispielhaft anzugeben.

In den Abschnitten 2.2 und 2.3 sind außerdem wesentliche Querbezüge vorhanden.

2.1 Reverse-Vickrey-Auktion

Die Vickrey-Auktion, siehe [Vickrey 1961], wurde 1961 erfunden und ist nach ihrem Schöpfer William Vickrey benannt, der für seine Erfindung den Wirtschafts-Nobelpreis erhielt. Es handelt sich um eine Zweitpreis-Auktion. Das bedeutet, es gewinnt derjenige, der das höchste Gebot abgegeben hat, allerdings muß er nur den Betrag des zweithöchsten Gebotes zahlen. Alle Gebote werden geheim abgegeben. Außerdem ist es jedem Auktionsteilnehmer nur gestattet ein einzelnes Gebot abzugeben.

Bei einer regulären (Erstpreis-)Auktion neigen Bieter dazu, weniger als den von ihnen geschätzten Wert der zu ersteigerten Ware zu bieten. Im Ergebnis ist die Auktion ineffizient, da das Gewinngebot selber tendenziell unterhalb des Wertes der Ware liegt. Der Verkäufer erhält somit weniger, als der gewinnende Bieter den Wert des Versteigerungsobjekts schätzt.

Die Zweitpreis-Auktion hingegen führt dazu, daß die Bieter tendenziell wahrheitsgemäß bieten, da sie selbst den Preis gar nicht beeinflussen können, den

sie im Falle des Auktionsgewinnes zahlen würden. Die Höhe des Gebots beeinflusst lediglich die Gewinnwahrscheinlichkeit. Das Gebot der aus Sicht des Bieters wahren Wertschätzung des zu ersteigernden Gutes zu bieten, ist die beste Vorgehensweise. Man bezeichnet dies als dominante Strategie.

Dadurch ergeben sich mit der Verwendung der Vickrey-Auktion die folgenden Vorteile:

- Der Bieter kann sicher sein, daß er niemals mehr als den gebotenen Preis bezahlt, sondern in der Regel weniger.
- Für den Verkäufer gilt, daß er in der Regel den größtmöglichen Betrag für die versteigerte Ware erhält, da die Bieter nicht zum Unterbieten neigen.
- Durch Vorgabe eines Mindestbetrages kann der Verkäufer sicher sein, daß er mindestens einen Betrag erhält, den er für angemessen hält.

Die Vickrey-Auktion hat allerdings auch Schwächen. Der wesentliche Nachteil ist, daß das Prinzip verwundbar gegenüber Absprachen der Bieter ist. Eine perfekte Absprache der Bieter würde letztendlich bedeuten, daß alle nur den Mindestpreis bieten. Dies hätte zur Folge, daß auch nur der Mindestpreis gezahlt werden müßte.

Der offensichtliche Nachteil bei dieser Art der Absprache besteht jedoch darin, daß alle Bieter sich im selben Moment, wie sie sich entschließen sich abzusprechen, sich gleichermaßen im Nachteil gegenüber jedem Auktionsteilnehmer befinden, der sich nicht an die Absprache hält. Dies bedeutet, daß sofern es nur einen Bieter gibt, der statt dem Mindestbetrag einen beliebigen, höheren Betrag bietet, er nicht nur die Auktion gewinnt, sondern auch nur den Mindestbetrag, sprich das zweithöchste Gebot, zahlt. Eine derartige Absprache kann also nur erfolgreich sein, wenn sie immer erfolgt, und sich auch alle Teilnehmer immer daran halten.

Ein weiterer Nachteil, ist daß das Prinzip anfällig gegenüber Scheingeboten ist. Wäre es jemandem möglich, für dieselbe Auktion zwei Gebote abzugeben, so könnte er mit dem einen Gebot den Preis senken, und mit dem zweiten Gebot den Gewinn der Auktion sicherstellen. Darauf, wie diesen Nachteilen entgegengewirkt werden kann, wird im Abschnitt über das Design des Auktionators eingegangen werden.

Bei der im letztendlich zu entwickelnden System verwendeten Auktionsform handelt es sich um eine Rückwärtsauktion. Das heißt, daß die Rollen von Käufer und Verkäufer vertauscht sind. Dies impliziert wiederum, daß sich die Preisvorgabe und die Preisfindung umkehren. Der Initiator der Auktion wird somit der Käufer (der Transportdienstleistung) sein, der Container, die Bieter sind die Anbieter dieser Dienstleistung, also die Transporter. Dadurch wird der Mindestbetrag, der durch den Initiator, den Käufer, vorgegeben wird, zum Maximalbetrag, den er bereit ist zu bezahlen. Gewinnendes Gebot wird das niedrigste sein, der letztendliche Preis wird dem zweitniedrigsten Gebot entsprechen.

Die Rückwärtsauktion wird auch als Beschaffungsauktion bezeichnet (engl. *sourcing auction*, *procurement auction*). Zur Begründung, warum die Reverse-Vickrey-Auktion für den betrachteten Anwendungsfall geeignet ist, siehe auch [Rosenfeld 2008].

2.2 Routenberechnung

Die Grundlage für die Routenberechnung ist eine Distanzmatrix. Sie enthält für jedes Paar zweier Knotenpunkte einen Eintrag, der der Entfernung zwischen diesen Knoten entspricht. Die Diagonale enthält demzufolge den Wert 0 an jeder Stelle. Ein Eintrag ∞ bedeutet, daß es zwischen den entsprechenden Knoten keine direkte Verbindung gibt.

Die Distanzmatrix muß nicht zwangsläufig symmetrisch sein. Für den Fall, daß die Strecke von Knoten a nach Knoten b in eine Richtung einen anderen Wert hat als in die andere, würden die Einträge $D(a, b)$ und $D(b, a)$ verschiedene Werte aufweisen.

Die Größe der Matrix ist die Genauigkeit, mit der ein gegebenes Wegenetz modelliert wird. Für den realen Anwendungsfall ist davon auszugehen, daß D mindestens eine Größe von $10^2 \cdot 10^2$ Einträgen hat. Die Größe der Distanzmatrix hat wesentlichen Einfluß auf die Laufzeiten von Routenberechnung und Optimierung.

Im folgenden eine beispielhafte Distanzmatrix aus [Rosenfeld 2008], die auch für die Implementation des Prototypen verwendet wurde.

$$D = \begin{pmatrix} 0 & \infty & 147 & \infty & 208 & \infty & 224 & 219 & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & 227 & \infty & \infty & 82 & 93 & \infty & \infty & \infty & \infty & \infty & \infty \\ 147 & \infty & 0 & \infty & 166 & \infty \\ \infty & 227 & \infty & 0 & \infty & \infty & \infty & 215 & \infty & 152 & \infty & \infty & \infty & \infty \\ 208 & \infty & 166 & \infty & 0 & \infty & \infty & 156 & \infty & 201 & \infty & 153 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & \infty & 116 \\ 224 & 82 & \infty & \infty & \infty & \infty & 0 & 119 & 75 & \infty & \infty & \infty & \infty & \infty \\ 219 & 93 & \infty & 215 & 156 & \infty & 119 & 0 & \infty & 252 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 75 & \infty & 0 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 152 & 201 & \infty & \infty & 252 & \infty & 0 & \infty & 188 & \infty & 133 \\ \infty & 0 & 134 & 137 & 165 \\ \infty & \infty & \infty & \infty & 153 & \infty & \infty & \infty & \infty & 188 & 134 & 0 & 175 & 138 \\ \infty & 137 & 175 & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & 116 & \infty & \infty & \infty & 133 & 165 & 138 & \infty & 0 \end{pmatrix}$$

Abbildung 2.1: Distanzmatrix

Die Indizes der Zeilen und Spalten stehen dabei für Städte in Deutschland:

- | | | |
|-----------------------|---------------------|-----------------------|
| <i>1 = Berlin</i> | <i>6 = Freiburg</i> | <i>11 = München</i> |
| <i>2 = Bremen</i> | <i>7 = Hamburg</i> | <i>12 = Nürnberg</i> |
| <i>3 = Dresden</i> | <i>8 = Hannover</i> | <i>13 = Passau</i> |
| <i>4 = Düsseldorf</i> | <i>9 = Kiel</i> | <i>14 = Stuttgart</i> |
| <i>5 = Erfurt</i> | <i>10 = Mainz</i> | |

2.2.1 Die Klasse Route-Node

Das Erstellen einer Route erfolgt auf der Basis von sogenannten Route-Nodes. Eine Route-Node bezeichnet einen Teilabschnitt der Route, und enthält im

wesentlichen Zielort des Teilabschnittes, Länge der Strecke und Ankunftszeitpunkt. Die Route-Node-Klasse aus [Rosenfeld 2008] wurde für diese Arbeit erweitert, um eine einfachere Kostenschätzung und -abrechnung zu ermöglichen. Im folgenden die (auf relevante Felder verkürzte) Java-Klassendefinition:

```
public class RouteNode {

    /* ... */

    // Ankunftszeitpunkt (Datum, Uhrzeit) dieses Teilabschnittes
    Date arrivalTime = null;

    // Index dieses Knotens innerhalb der Route
    int sequenceNumber = 0;

    // Innerhalb dieses Abschnittes zurückgelegte Distanz
    double distance = 0.0;

    // Fahrzeit für diese Teilstrecke in Stunden
    double drivingTime = 0.0;

    // Service-Zeit in Stunden (Be- oder Entladen)
    double serviceTime = 0.0;

    // Pausenzeit innerhalb dieses Abschnittes (neu)
    double breakTime = 0.0;

    // Schlafzeit (neu)
    double sleepingHours = 0.0;

    // reine streckenabhängige Kosten für diesen Abschnitt (neu)
    double cost = 0.0;

    // Veränderung des Ladungszustandes (-1.0, 0.0 oder 1.0)
    double loadChange = 0.0;

    // Name des Zielortes für diesen Abschnitt
    String name = "";

    // benötigte Variablen im Falle des Nichteinhaltens des Zeitplans
    String violationCategory = null;
    String violationAttribute = null;
    String violationValue = null;
}
```

Abbildung 2.2: Codebeispiel Route-Node

2.2.2 Verwendung von Route-Nodes

Soll nun eine Route, daß heißt eine Liste von Route-Nodes, zur Erfüllung eines gegebenen Auftrags erstellt werden, wird zunächst eine Liste der Stationen (Knoten) für den entsprechenden Weg ermittelt. Dafür kann der Dijkstra-Algorithmus zur Bestimmung des kürzesten Pfades zwischen zwei Knoten verwendet werden. Die Grundlage dafür ist die Distanzmatrix.

Zunächst wird der Weg vom Standort des Transportfahrzeuges zum Zeitpunkt des Auftragsantritts zum Ladeort des Containers bestimmt. Daran wird dann der Weg zum Zielort des Containers angehängt. Falls sich der Transporter bereits am selben Ort wie der Container befindet (bzw. befinden wird), entfällt offensichtlich der erste Schritt.

Auf Basis der o. g. Distanzmatrix ergibt sich dies für einen Auftrag von Nürnberg nach Hamburg mit Start in Freiburg folgendermaßen:

Schritt 1 (Standort → Ladeort):

Freiburg → Stuttgart → Nürnberg

Schritt 2 (Ladeort → Zielort):

Nürnberg → Erfurt → Hannover → Hamburg

Die Gesamtroute ergibt sich somit trivialerweise zu:

Freiburg → Stuttgart → Nürnberg → Erfurt → Hannover → Hamburg

Sobald alle Stationen ermittelt sind, die auf der Route liegen, werden anhand dieser Liste die entsprechenden Route-Nodes erstellt. Dazu ist neben der Distanzmatrix die Kenntnis folgender Parameter notwendig:

- Durchschnittsgeschwindigkeit
- Weg-Kostenfaktor
- Zeitbeschränkungen bezüglich (Schlaf-)Pausen
- Zeit zum Be- und Entladen

Der Algorithmus, der aus der Liste der Stationen die Route-Nodes erzeugt, sieht in Pseudo-Code wie folgt aus (vgl. RouteNode-Klassendefinition):

Benötigte Konstanten:

```
// Distanz zwischen zwei Stationen (aus der Distanzmatrix)
Teilstrecke;

// Durchschnittsgeschwindigkeit des Fahrzeugs
Durchschnittsgeschwindigkeit;

// zum Be- oder Entladen eines Containers benötigte Zeit
Ladezeit;

// Weg-Kostenfaktor aus der Kostenfunktion, siehe Abschnitt 2.3
Weg_Kostenfaktor;
```

Benötigte globale Variablen:

```
// Zeitpunkt der Verfügbarkeit für neue Aufträge
Verfügbarkeitsdatum;

// Variablen bzgl. des Fahrerzustandes, bezogen auf den
// Verfügbarkeitszeitpunkt
Fahrzeitbeschränkungen;
```

2 Theoretische Grundlagen

Funktion erstelle_RouteNodes

```
erhält kürzester_Pfad, // z.B. {Kiel, Hamburg, Berlin, Dresden}
        Ladeort, Zielort // z.B. Hamburg, Dresden

gibt zurück Liste von RouteNodes {

Route = leere Liste von RouteNodes;
boolean beladen = falsch;

für i von 1 bis Länge von kürzester_Pfad {

    Node = neue RouteNode;

    // Sequenznummer, Name des Abschnittsziels, Strecke und
    // Fahrzeit setzen
    Node.sequenceNumber = i;
    Node.name = kürzester_Pfad[i];
    Node.distance = Länge_des_Teilabschnittes;
    Node.drivingTime = Node.distance / Durchschnittsgeschw.;

    // ggf. Beladungszustand ändern
    falls Node.name ist Ladeort und nicht beladen {
        Node.loadChange = 1.0;
        Node.serviceTime = Ladezeit;
        beladen = wahr;
    }
    sonst falls Node.name ist Zielort und beladen {
        Node.loadChange = -1.0;
        Node.serviceTime = Ladezeit;
        beladen = falsch;
    }
    // Pausen einplanen
    node.breakTime =
        Fahrzeitbeschränkungen.Pause(Node.drivingTime);
    node.sleepingHours =
        Fahrzeitbeschränkungen.Schlaf(Node.drivingTime);

    // Dauer und Ankunftsdatum für Teilstrecke setzen
```

```

    Dauer = Node.drivingTime + Node.serviceTime +
           Node.breakTime + Node.sleepingHours;

    falls i = 1 Node.arrivalTime = Verfügbarkeitsdatum + Dauer;
    sonst Node.arrivalTime = Route(i - 1).arrivalTime + Dauer;

    // wegabhängige Kosten für diesen Teilabschnitt berechnen
    Node.cost = Weg_Kostenfaktor * Node.distance;

    hänge Node an Route;
}
gib zurück Route;
}

```

Abbildung 2.3: Erstellung von Route-Nodes

Dieser Algorithmus hat also die Aufgabe, aus dem kürzesten Pfad, gegeben als Folge von Knoten, eine Liste von Route-Nodes zu erstellen. Unabhängig von der konkreten Implementation müssen in diesen Route-Nodes folgende Felder mit Werten besetzt sein:

- Name des jeweiligen Zielortes bzw. -knotens
- Distanz
- Distanzkosten
- Fahrzeit
- ggf. die Zeiten für (Schlaf-)Pausen und Service
- Änderung des Beladungszustandes an Start- und Zielort
- Ankunftszeitpunkt

Der Algorithmus muß die Fahrzeitbeschränkungen des Transporters kennen. Dieser Umstand liegt darin begründet, daß die Kostenschätzung und Gebotsabgabe auf Grundlage der Liste von Route-Nodes erfolgt. In dieser Liste

müssen die Pausenzeiten vorhanden sein. Um wiederum die Pausen korrekt planen zu können, müssen die oben genannten Variablen bekannt sein (Zeit bis zur nächsten Pause, Zeit bis zur nächsten Schlafpause. Falls beispielsweise der Fahrer am Ende eines Auftrages vor dem neuen, zu ersteigernden Auftrag nur noch eine dreiviertel Stunde fahren dürfte, bis er eine längere Schlafpause einlegen müßte, so muß dieser Umstand vor der Berechnung der für den neuen Auftrag entstehenden Route bekannt sein, damit zum einen die Durchführbarkeit im Hinblick auf Zeitbeschränkungen des Containers gewährleistet ist, und zum anderen eine gute Kostenschätzung erfolgen kann (siehe Abschnitt 3.2.2).

2.3 Kostenfunktion und Gebotsberechnung

Damit die Transport-Agenten selbständig auf Angebote reagieren können ist es essentiell, daß sie in der Lage sind, ihre Kosten realistisch zu schätzen. Dies geschieht durch eine individuelle Kostenfunktion. Prinzipiell können die Kostenfunktionen der Transportagenten beliebig komplex sein. Jedoch lassen sich mit sehr einfachen Annahmen und Formeln bereits relativ gute Schätzungen erzielen. Im folgenden werden eine Kostenfunktion und eine Gebotsfunktion konstruiert. Letztere enthält dabei strategieabhängige Parameter, die den Planungshorizont des Transport-Agenten berücksichtigen.

2.3.1 Weg- und zeitabhängige Kostenfaktoren

Eine grundsätzliche Annahme bezüglich der Kostenfunktion eines Transport-Agenten ist, daß sich seine Kosten aus Zeit-Kosten und Weg-Kosten zusammensetzen. Dies erscheint intuitiv und entspricht der gängigen Praxis bei der Datenverarbeitung in der Logistik-Branche (vgl. zum Beispiel [www.frachtprofi.at]).

Für bestimmte Kostenfaktoren ist die Einordnung in eine der beiden Kategorien nicht offensichtlich. Ein Beispiel dafür sind die Anschaffungskosten eines LKW. Der Wertverlust für die Nutzungsperiode eines Fahrzeugs kann sowohl auf die Summe der gefahrenen Kilometer innerhalb des Zeitraumes, als auch auf den

Zeitraum selber bezogen werden. Für die letztendliche Berechnung der Kosten ist dies allerdings von relativ geringer Bedeutung, sofern die relevanten Größen hinreichend gut geschätzt werden können.

Im folgenden eine beispielhafte Auflistung von Kenngrößen beider Kategorien:

Zeitabhängige Kostenfaktoren

- Wertverlust des Fahrzeugs innerhalb seines Nutzungszeitraums
- Reparatur- und Wartungskosten
- Versicherungskosten
- KFZ-Steuer
- Fahrerabhängige Kosten (Stundenlohn)

Wegabhängige Kostenfaktoren

- Öl- und Treibstoffverbrauch
- Reifen (Anzahl, Laufleistung, Kosten)
- Mautgebühren
- (Wartungskosten)

(Diese können als Weg- oder Zeit-Kostenfaktor interpretiert werden, je nach unternehmensabhängigem Berechnungsmodell, z. B. „dreimal pro Jahr“, oder „alle 50.000km“.)

Um einige der zeitabhängigen Kostenfaktoren in die Kostenberechnung für einen konkreten Auftrag einbeziehen zu können, müssen diese zunächst normiert werden. Das bedeutet daß diese Faktoren zunächst auf den gleichen Zeitraum bezogen werden, beispielsweise ein Jahr. Daraufhin muß ermittelt werden, wie häufig beziehungsweise wie lange das jeweilige Fahrzeug innerhalb dieses Zeitraumes eingesetzt wird (zum Beispiel 250 Tage pro Jahr für 10 Stunden pro Tag). So läßt sich dann beispielsweise die Versicherungsgebühr für ein Jahr auf die Einsatzdauer eines Auftrages umrechnen.

Die wegabhängigen Kosten lassen sich dagegen sehr einfach und zudem sehr genau für eine Schätzung verwenden, da sie sich unmittelbar auf die Wegstrecke beziehen, die durch einen Auftrag vorgegeben ist. Unter diesem Gesichtspunkt scheint es naheliegend, möglichst viele Kostenfaktoren eher auf den Weg als auf die Zeit zu beziehen, da mit der Normierung unter Umständen eine weitere Schätzung verbunden ist (siehe oben, Wartungskosten).

2.3.2 Kostenfunktion

Die Kostenfunktion gibt die (bestmögliche Schätzung der) realen Kosten einer gegebenen Route ohne Berücksichtigung des Rückwegs an. Als wesentliche Parameter wurden die Faktoren der Kosten pro zurückgelegter Strecke und die Kosten pro Einsatzdauer eines LKW ermittelt. Diese lassen sich, normiert, als Kosten pro Kilometer beziehungsweise Kosten pro Stunde ausdrücken. Diese Parameter sind unternehmensabhängig aber strategie-unabhängig.

Die Route hat die Form einer Sequenz von Route-Nodes (siehe 2.2.1). :

$$\begin{aligned} \text{Route: } r &= (x_1, x_2, x_3 \dots x_n) \\ \text{mit } x_i &= \text{RouteNode } i \end{aligned}$$

Für jeden Teilabschnitt sind Zeitaufwand (ggf. abzüglich Schlafzeit) und Distanz bekannt. Dabei sind:

$$\begin{aligned} c_d &= \text{Distanz-Kostenfaktor} & d(x_i) &= \text{Distanz von RouteNode } i \\ c_t &= \text{Zeit-Kostenfaktor} & t(x_i) &= \text{Zeitaufwand für RouteNode } i \end{aligned}$$

Die Kosten für die Route ergeben sich damit zu:

$$\text{Kosten: } k_r(r) = \sum_{i=1}^{i=n} c_d \cdot d(x_i) + c_t \cdot t(x_i)$$

2.3.3 Gebotsfunktion

Die Gebotsfunktion muß für einen gegebenen Auftrag eine zuverlässige Schätzung für ein angemessenes Gebot liefern. Das Gebot soll dabei der wahren Wertschätzung des Auftrages durch den Transporter entsprechen. Wie es gelingen kann, anhand der mit der Kostenfunktion gegebenen realen Kosten für einen Auftrag zu einem sinnvollen Wert gelangen, soll im folgenden aufgezeigt werden.

Planung des Rückwegs

Die durch den Rückweg anfallenden Kosten müssen in die Schätzung mit einbezogen werden. Das generelle Problem dabei ist die Unsicherheit, ob am Zielort weitere Aufträge verfügbar sind, oder ob es zu einer Leerfahrt kommt. Wenn es sich beim Zielort eines Auftrages um einen logistischen Knotenpunkt handelt (Seehafen o. ä.), ist die Wahrscheinlichkeit größer, dort einen Folgeauftrag annehmen zu können, als dies beispielsweise für abgelegene Regionen der Fall wäre.

Eine effektive Möglichkeit, diesen Umstand in die Kostenschätzung mit einzubeziehen besteht darin, für Zielorte oder Zielregionen die Auftragshäufigkeit bzw. -wahrscheinlichkeit zu ermitteln. Die für den Rückweg (Leerfahrt) anfallenden Kosten würden dann nur soweit in das Gebot mit einbezogen, wie es wahrscheinlich ist, daß am Zielort kein weiterer Auftrag verfügbar ist. Da Folgeaufträge mit einer gewissen Wahrscheinlichkeit wieder in Richtung Heimatort des Transporters führen, sowie die Wahrscheinlichkeit von

Leerfahrten verringern, wirkt sich dies positiv auf die zu erwarteten Folgekosten aus, die durch die Gebotsfunktion ausgedrückt werden sollen. Geringere Gebote wiederum bedeuten höhere Gewinnwahrscheinlichkeit und damit mehr Planungssicherheit. Inwiefern die Verfügbarkeit von Folgeaufträgen sich auf die Höhe des Gebots auswirken soll ist strategieabhängig.

Einbeziehung nichtmonetärer Faktoren

Hier soll insbesondere der „Rückkehr-Faktor“ in Betracht gezogen werden. Damit ist der Umstand gemeint, daß (abhängig von Arbeitsverhältnis, Geschäftsmodell etc.) ein Anspruch des Fernfahrers besteht, regelmäßig nach hause zurückkehren zu können. Prinzipiell ließe sich dies in einen monetären Faktor umrechnen. Besteht eine diesbezügliche Vereinbarung zwischen Unternehmer und Fahrer, so wird der Faktor explizit oder implizit darin enthalten sein. Außerdem kann durch Einführung eines entfernungsabhängigen Faktors die Bildung von kreisförmigen Routen begünstigt werden. Das würde dazu führen, daß der Preis für Aufträge, die den Transporter in Richtung seines Heimatortes führen, geringer ausfiele. Das wiederum bedeutet, daß sich die Wahrscheinlichkeit erhöht, einen Folgeauftrag zu erhalten, der eine Rückwegskomponente enthält. Der Faktor ist strategieabhängig.

Gewinnerwartung

Der letzte Aspekt der Gebotsfunktion ist das Einbeziehen einer Gewinnerwartung. Offensichtlich wird in einem realen Anwendungsszenario ein Anbieter von Transportdienstleistungen nicht zum Selbstkostenpreis arbeiten. Dieser Umstand ist nicht zwangsläufig durch das Streben nach Gewinn begründet, sondern spielen dabei auch Sicherheitsaspekte eine Rolle. Der Unternehmer muß die Sicherheit haben, daß im Falle einer Panne oder sonstiger Unwägbarkeiten genug (monetäre) Reserven vorhanden sind, um eventuell entstehende zusätzliche Kosten kompensieren zu können.

Der Parameter der Gewinnerwartung wird im Verhältnis zu den tatsächlichen Kosten ausgedrückt. Er stellt einen Kompromis aus Gewinn und Sicherheit entgegen Auftragswahrscheinlichkeit dar und ist damit strategieabhängig. Durch Veränderung der Gewinnerwartung kann auch Unsicherheit bezüglich der anderen Parameter entgegengewirkt werden.

In mathematischen Formeln ausgedrückt hat die Gebots- bzw. Kostenfunktion folgende Gestalt (Der Beladungszustand ist hier vernachlässigt worden, er ließe sich durch das Wählen zweier verschiedener Distanzkostenfaktoren berücksichtigen):

Mathematische Formulierung

Die realen Kosten der Route sind durch $k_r(r)$ gegeben.

Die x_i bezeichnen hier die Knoten entlang der Route. Außerdem sei x_0 der Aufenthaltsort zu Beginn des Auftrags und x_h der Heimatort.

Die Parameter sind:

Rückwegfaktor, mit

$$p(x_i) = \text{geschätzte Auftragswahrscheinlichkeit an } x_i$$

$$R_p(r) = k_r(x_n \dots x_h) \cdot (1 - p(x_n))$$

Entfernungs- bzw. Rückkehrfaktor:

$$E_f(r) = f \cdot (k_r(x_n \dots x_h) - k_r(x_0 \dots x_n)) \quad , \quad 0 < f \ll 1$$

Der Wert von f gewichtet dabei, wie es sich auf das Gebot auswirken soll, ob durch Erledigung des Auftrags der Transporter seinem Heimatort näher kommt, ausgehen vom Standort vor Antritt des Auftrages. Da f klein ist, können die k_r in der obigen Formel grob geschätzt werden.

$$m = \text{Gewinnerwartung} \quad , \quad 0 < m \ll 1$$

Das Gebot auf die Durchführung des Auftrags, der durch r vorgegeben ist:

$$\text{Gebot } g_{f,p,m}(r) = (k_r(r) + R_p(r) + E_f(r)) \cdot (1 + m)$$

Es setzt sich aus den realen Kosten der Route k_r (ohne Rückweg) und einem strategieabhängigen Teil zusammen. Zur Bedeutung und Wahl der verschiedenen Parameter siehe auch Abschnitt 3.2.3.

3 Modellierung der Agenten

In diesem Kapitel geht es darum, wie die in Kapitel 2 ermittelten Grundlagen als Bausteine bei der Entwicklung von Agenten verwendet werden. Der Container-Agent ist vergleichsweise einfach gebaut und durch seine Transportbedingungen und seine Preisfunktion definiert. Der Transport-Agent hat die Aufgabe, auf die Durchführung von Aufträgen zu bieten und diese zu planen. Um diese Aufgaben zu erfüllen, muß er die in Kapitel 2 erarbeiteten Mechanismen zur Routenberechnung und Gebotskonstruktion implementieren. Der Auktionator vermittelt Container in Auktionen an die Transporter, wobei er die wesentliche Kommunikation übernimmt. Er ermittelt den Gewinner der Auktion und den Preis, und erhebt eine kleine Gebühr.

3.1 Container-Agent

Der Auftraggeber eines Transports ist im zu entwickelnden System durch den zu transportierenden Container repräsentiert. Für eine reale Anwendung kann von der Existenz von Container-Agenten ausgegangen werden (siehe Kapitel 5), die autonom ihre eigene Beförderung planen. Dazu ist ihnen ein Konto gegeben, von dem sie die Kosten bezahlen. Wieviel ein Container für seinen Transport bereit ist zu investieren, richtet sich beispielsweise nach folgenden Faktoren:

- Entfernung zum Transportziel
- verbleibende Zeit bis zum spätesten Auslieferungstermin, bzw. spätestem Abholungstermin
- anfallende Kosten bei nicht zeitgerechter Auslieferung
- frachtspezifische Parameter (Gefahrgutklasse, Kühlung etc.)
- Verfügbarkeit von (geeigneten) Transportern am momentanen Standort

Der Container-Agent wird alle diese Faktoren intern verwalten. Auf Anfrage des Auktionator wird er sein Transportziel und seinen momentanten maximalen Preis nennen, den er für die gegebene Strecke bereit ist zu bezahlen. Das Ergebnis dieser Anfrage, über mehrere Anfragezeitpunkte (Auktionen) aufgezeichnet, ist die Preisfunktion des Containers. Sie ist von den Informationen des Containers über seine Umwelt beeinflusst.

Wäre dem Container zum Beispiel bekannt, daß sich an seinem Standort in der Regel nur wenige Transporter und wenige Container aufhalten, so würde er bei fern in der Zukunft liegendem spätesten Abholungstermin zunächst einen geringen Maximalpreis veranschlagen, der aber umso steiler ansteigen würde, je näher dieser Termin rückt. Befindet er sich hingegen an einem Knotenpunkt mit hoher Verfügbarkeit von Transportern aber wenigen Containern, so wird die Preisfunktion tendenziell zu Beginn höher ansetzen, aber flacher steigen. Die Funktion wird normalerweise monoton steigend sein.

In der folgenden Abbildung sind mögliche Preisfunktionen beispielhaft dargestellt.

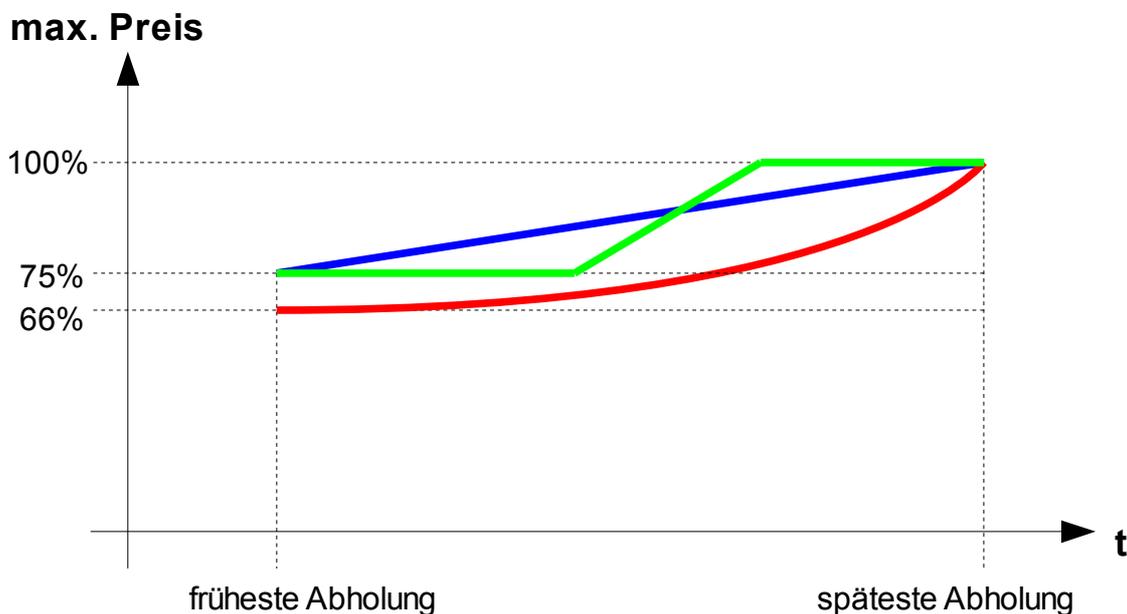


Abbildung 3.1: beispielhafte Preisfunktionen für Container

Der Container-Agent ist zunächst sehr einfach strukturiert. Seine Modellierung als Agent zielt in erster Linie auf zukünftige Anwendungen. Es wäre beispielsweise denkbar, daß der Container häufig Informationen aus seiner Umwelt aufnimmt oder mit seinem Auftraggeber kommuniziert, um gegebenenfalls seine Preisfunktion zu optimieren oder Lieferbedingungen anzupassen. Bei solchem autonomen Verhalten ist die Modellierung als Agent vorteilhaft. Im Prototyp des Systems ist der Container als einfaches Objekt mit seinen charakteristischen Parametern vertreten.

3.2 Transport-Agent

Der Transport-Agent ist durch seine Kostenfunktion, seine Verfügbarkeit und die Fähigkeit definiert, Transportaufträge für Container zu erwerben. Dazu ist er in der Lage, anhand seiner (ggf. zukünftigen) Verfügbarkeit und der Lieferzeit-Beschränkungen des Containers zu prüfen, ob eine Durchführung des Transports möglich ist. Falls dem so ist, wird er dem Auktionator ein Gebot zu übermitteln, das sich aus der strategieabhängigen Gebotsfunktion ergibt. In diese gehen die Kosten für die durch den Auftrag implizierte Route ein.

Das Verhalten des Agenten ist durch viele Parameter bestimmt. Einige davon kann man als konstant ansehen, zum Beispiel die Kostenfunktion und die Fahrzeitbeschränkungen. Andere wird der Agent laufend seinem Status und seiner Planung anpassen. Dazu zählen Verfügbarkeit und strategieabhängige Faktoren für weitere Gebote. Im folgenden eine Beschreibung aller Parameter gegeben, die den Agenten beeinflussen.

3.2.1 Attribute des Transportagenten

Die Kostenfunktion des Agenten ist ihm mit den Faktoren c_d und c_t gegeben. Weitere Parameter, die als konstant angenommen werden, sind der Faktor der Gewinnerwartung $m \approx 0,05$, die Durchschnittsgeschwindigkeit v_d und die Fahrzeitbeschränkungen, das heißt Dauer und Frequenz von Pausen und Länge von Arbeitstages und Übernachtung.

Parametern, die der Agent seiner Planung anpaßt sind der Zeitpunkt seiner Verfügbarkeit, der Ort seiner Verfügbarkeit bzw. die Liste von Orten an denen er bereit ist Aufträge anzunehmen, sowie die strategieabhängigen Parameter der Gebotsfunktion, E_f bzw. f und $p(x_i)$. Außerdem hält er seine momentane Auftrags- und Routenplanung in Form einer Liste von Route-Nodes fest. Aus dieser Planung ergeben sich weitere Verfügbarkeit, Erholtheit des Fahrers, und ggf. eine Anpassung von f oder $p(x_i)$.

3.2.2 Durchführbarkeit von Aufträgen

Erhält der Transporter ein Auftragsangebot, muß er zunächst prüfen, ob er den Auftrag überhaupt durchführen kann. Dazu muß er von seiner aktuellen Planung ausgehend eine Route berechnen, die den Auftrag erfüllt. Die Route ist gültig wenn alle Zeitvorgaben eingehalten werden können, und der Auftrag ist dann durchführbar. Die Berechnung der Route erfolgt mit dem Algorithmus aus 2.2.2.

Grundsätzlich kann, bevor oder nachdem die Route auf Gültigkeit geprüft wurde, eine Optimierung über die bereits geplante Route und die neu berechnete Route erfolgen. Im Falle des Container-Szenarios wird dieser Schritt jedoch trivial, da Aufträge immer sequentiell abgearbeitet werden.

Es bestünde allerdings die Möglichkeit, Aufträge zu suchen, die unmittelbar Leerfahrten kompensieren könnten. Dazu würde der Transporter, Auftragsangebote an Orten entlang (bevorzugt zu Beginn) seiner Leerfahrt entgegennehmen, und bei Erhalt eines entsprechenden Angebotes prüfen, ob der Auftrag eingeplant werden kann, unter anderem ob der Zielort vor Beendigung der Leerfahrt liegt. Der Algorithmus aus 2.2.2 kann das nicht leisten, er müßte dazu verkompliziert werden, oder es müßte die oben genannte Optimierung erfolgen. Dies ist im Hinblick auf komplexere Szenarien (beispielsweise größeren Kapazitäten) zwangsläufig gegeben. Laufzeit der Optimierung könnte sich als problematisch erweisen, da gemäß der Anforderungen das Gebot zügig erfolgen muß.

Im vorliegenden Fall verhält es sich dagegen relativ einfach. Der Agent erhält ein Angebot, konstruiert die Route, und falls diese gültig ist, wird der Agent die Berechnung der Kosten vornehmen und ein Gebot formulieren.

3.2.3 Gebotsabgabe

Sobald der Transportagent für einen gegebenen Auftrag eine gültige Route berechnet hat, kann er die realen Kosten direkt berechnen. Um ein Gebot abzugeben, muß er jedoch weitere Dinge berücksichtigen. Hier noch einmal die Gebotsfunktion und ihre Anwendung anhand eines Beispiels:

$$g_{f,p,m}(r) = (k_r(r) + R_p(r) + E_f(r)) \cdot (1 + m)$$

Die Kosten für eine gegebene Route seien $k_r(r) = 580,00 \text{ €}$.

Zunächst wird der Rückweg einbezogen. Der Faktor $R_p(r)$ gibt die für den Rückweg zum Heimatort des Transporters einzuplanenden Kosten in Abhängigkeit der Auftragswahrscheinlichkeit am Endpunkt der Route an. Je höher die Wahrscheinlichkeit für einen Folgeauftrag ist, desto kleiner ist der Wert von $R_p(r)$.

Sei für eine Auftragswahrscheinlichkeit von 40% am Ziel der Route und Kosten von 98,00 € für 224 km Heimweg $R_p(r) = 85,80 \text{ €}$.

Dann wird der entfernungsabhängige Faktor $E_f(r)$ ermittelt. Angenommen, der Auftrag würde den Transporter, von seinem momentanen Aufenthaltsort aus gesehen, 450 km in Richtung seines Heimatortes führen. Er beziffere die Kostendifferenz bzw. Ersparnis auf $-175,00 \text{ €}$ (Er müßte die 450 km sonst ggf. leer fahren). Dies sei gewichtet mit $f = 0.2$ und damit ist $E_f = -35,00 \text{ €}$.

Zuletzt wird der Parameter m eingerechnet. Dabei handelt es sich um die Gewinnerwartung/Sicherheit, die auch angepaßt werden könnte, um Unsicherheit bei der Bestimmung der anderen Parameter auszugleichen.

Sei $m = 5/100 = 0,05$.

Insgesamt wäre sein Gebot:

$$g_{f,p,m}(r) = (580,00 \text{ €} + 85,80 \text{ €} - 35,00 \text{ €}) \cdot 1,05 = 662,34 \text{ €}$$

Der Sinn, die Gebotsfunktion durch die in Kapitel 2 eingeführten Parameter, zu beschreiben, ergibt sich aus der Betrachtung einiger Extremfälle.

Sei bezüglich $R_p(r)$ angenommen, die Auftragswahrscheinlichkeit am Zielort wäre 100%. Dies wäre z. B. der Fall, wenn der Agent mit dem Auftrag r eine Leerfahrt füllt. Dann ist $R_p(r) = 0$, was bedeutet, daß sich eventuelle Rückwegskosten zunächst nicht auf das Gebot auswirken. Findet der Agent später (für Zeiten nach Ende seiner Planung) einen Auftrag, der ihn nachhause führt, und war es im bis dahin gegeben, für alle Teilaufträge seiner Planung von einer hohen Folgeauftrags-Wahrscheinlichkeit auszugehen, so wären letztendlich die Rückwegskosten nie in Form einer (eingepflanzten) Leerfahrt angefallen. Dementsprechend wären sie auch nur zu einem geringen Teil in die Gebote mit eingegangen.

Für Routen mit geringer Folgeauftrags-Wahrscheinlichkeit und großer Entfernung vom Heimatort erhöht sich das Risiko des Agenten. Dies würde durch einen entsprechend größeren Wert von $R_p(r)$ ausgedrückt, der gegen die realen Kosten für den Heimweg konvergiert, und zu einem höheren Preis für den Auftrag führt.

Der Wert von $E_f(r)$ ergibt sich in Abhängigkeit der Veränderung der Entfernung vom Heimatort durch die Route r . Der Wert wird durch den Parameter f gewichtet. Wenn der Agent noch keine Route geplant hat und sich an seinem Heimatort befindet, dann muß f sehr klein gewählt werden, da ihn ohnehin jeder Auftrag von zuhause wegführt, und damit zunächst keine Anpassung des Gebots erfolgen soll. Ist der Transporter weit vom Heimatort entfernt (und naht ggf. das Datum, zu dem der Fahrer wieder zuhause sein muß), sollten für f größere Werte gewählt werden. Das hat zur Folge, daß der Agent Aufträge, die ihn zu seinen Heimatort oder in dessen Richtung führen, deutlich gegenüber anderen bevorzugt, indem er mit $E_f(r) < 0$ geringere Gebote abgibt.

Es ist festzustellen, daß der Auftrag und die Gebotsfunktion Parameter enthalten, die den Nutzen des Agenten beschreiben. Kosten- und Gebotsfunktion rechnen diesen abstrakten Wert des Nutzens (*engl.* utility) in einen monetären Wert um, das Gebot. Wie die sinnvolle Gewichtung der Faktoren erfolgt, beruht auf menschlicher Erfahrung und maschinellem Lernen.

3.2.4 Planung von Aufträgen

Geht der Transporter als Sieger aus einer Auktion hervor, so wird er die für den neuen Auftrag berechnete Route in seine Planung aufnehmen. Im vorliegenden Fall bedeutet das für den Transporter, daß er die für den Auftrag berechnete Route an seine momentan geplante Route anhängt (sofern denn eine vorhanden ist). Prinzipiell besteht die Möglichkeit der Optimierung der Route. Das heißt, daß ein Agent die Reihenfolge der Stationen auf seiner Route verändert, und damit ggf. die Reihenfolge von Be- und Entladungen (siehe z. B. [Dorer/Calisti 2005]). Da in dieser Arbeit als Fracht nur Container und Transporter mit der Kapazität betrachtet werden, maximal einen Container zur Zeit zu transportieren, ist das in diesem Fall zu vernachlässigen.

Eine Konsequenz aus der Planung von Aufträgen ist, daß der Agent für weitere, zukünftige Aufträge an einem anderen Ort verfügbar sein wird, nämlich dem Zielort des zuletzt geplanten Auftrages. Das meldet er wiederum dem Auktionator, so daß dieser es beim Angebot weiterer Aufträge berücksichtigen kann.

Weiterhin muß der Agent unter Umständen seine Gebotsfunktion anpassen, um am Zielort bevorzugt solche Aufträge anzunehmen, die ihm in der jeweiligen Situation den größten Nutzen verschaffen (siehe oben, Abschnitt 3.2.3).

3.2.5 Implikationen kurzer Auktionsdauer

Ein wesentlicher Grund, warum (insbesondere im Hinblick auf ein anderes Szenario bezüglich Fracht und Kapazität der Transporter) die Routenplanung und Kostenschätzung nicht anhand einer Optimierung über den gesamten Planungsraum des Transporters erfolgt (d. h. den momentanen Zustand und bereits angenommene, zukünftige Aufträge), ist, daß gemäß der Anforderungen aus Kapitel 1.2, ein schneller Ablauf der Auktionen gewährleistet sein muß. Das heißt, daß die Gebotsabgabe relativ schnell nach Erhalt des Auftragsangebots erfolgen muß.

Es erscheint realistisch, für den realen Anwendungsfall von einer Größenordnung von 10 bis 10^2 Transportern auszugehen, die gleichzeitig an einer Auktion teilnehmen. Im Hinblick auf etwaige Parallelisierung von Auktionen (siehe

Kapitel 3.3), erscheint die Durchführbarkeit einer Größenordnung von 10^2 bis 10^3 Auktionen pro Stunde durch einen einzelnen Auktionator möglich.

Die Planungssicherheit aus Sicht des Transporters ergibt sich nicht aus der gleichzeitigen Verfügbarkeit vieler möglicher Aufträge, sondern aus der hohen Frequenz, mit der diese angeboten werden. Durch die Reduktion der Auftragscharakteristika auf einen Zahl, das Gebot, wird die (sich aus Planungsumständen ergebende) Attraktivität des Auftrages in eben dieser Zahl ausgedrückt, siehe dazu Kapitel 2.3. Aufträge, die dem Transporter „unbequem“ erscheinen, würden somit automatisch teurer bzw. besser bezahlt werden. Im optimalen Fall bedeutet das, daß die durch die „Unbequemlichkeit“ entstehenden Mehrkosten durch einen höheren Preis im Falle des Zuschlages mindestens kompensiert würden. Eine gute Kostenschätzung nach Kapitel 2.3 ist dafür die Voraussetzung. Für den hier betrachteten Fall der streng sequentiellen Auftragsabarbeitung scheinen diesbezüglich sehr genaue Schätzungen möglich.

3.2.6 Responder-Protokoll

Es folgt nun noch eine Beschreibung der Kommunikation des Transporters mit dem Auktionator. Grundlage dafür ist das Contract-Net-Protokoll (siehe [www.fipa.org]), das entsprechend erweitert wurde. Die Durchführung des Protokolls bestimmt maßgeblich das Verhalten des Agenten.

Benötigte Variablen und Konstanten:

```
Aufenthaltort;  
Kostenfunktion;
```

Verhalten Responder_Protokoll {

```
empfangen Auftragsangebot; // nicht blockierendes Warten  
  
Start = Auftragsangebot.Start;  
Ziel = Auftragsangebot.Ziel;  
Preis = Auftragsangebot.Maximalpreis;  
  
Knoten = finde_kürzesten_Pfad(Aufenthaltort, Start, Ziel);
```

```
// siehe Kapitel 2.2.2
Route = erstelle_RouteNodes(Knoten, Start, Ziel);

// siehe Kapitel 3.2.2
falls prüfe_Gültigkeit(Route) ist wahr {

    // siehe Kapitel 3.2.3
    Gebot = berechne_Gebot(Route, Kostenfunktion);

    falls Gebot <= Preis sende Gebot;
}

empfange Auktionsergebnis;

// fall das Gebot erfolgreich war, gewinnt der Agent den Auftrag
falls Auktionsergebnis.PERFORMATIVE ist ACCEPT_PROPOSAL {

    // siehe Kapitel 3.2.4, neuer Auftrag wird hinzugefügt
    update_Route(Route);

    // siehe Kapitel 3.2.3
    update_Gebotsfunktion(Aufenthaltort, Ziel)

    // Rückmeldung beim Auktionator für weitere Aufträge
    registriere_Aufenthaltort(Ziel);

} sonst {
    // ggf. update_Gebotsfunktion etc.
}
// Neustart des Verhaltens
reset Responder_Protokoll;
}
```

Abbildung 3.1: Responder-Verhalten

3.3 Auktionator-Agent

Der Auktionator ist der Auftragsvermittler im System. Die Transportagenten müssen sich bei ihm registrieren. Dabei übermitteln sie in erster Linie die Liste von Orten, an denen sie bereit sind Aufträge anzunehmen. Der Auktionator verwaltet zudem einen Vorrat von Containern, die auf der Suche nach Transportmitteln sind. Soll für einen Container eine Auktion durchgeführt werden, stellt der Auktionator den momentanen maximalen Preis und die Zeitvorgaben für die Lieferung fest. Diese Informationen übermittelt er an diejenigen Transporter, die am Startpunkt des Containers registriert sind, so daß diese ein Preisangebot machen können.

Die von den Transportern abgegebenen Gebote werden dann gesammelt und der Gewinner wird nach den Vorgaben aus Kapitel 2.1 ermittelt und benachrichtigt. Bei erfolgreicher Vermittlung eines Containers erhebt der Auktionator eine Gebühr. Sofern der Container zum gegebenen Zeitpunkt kein Transportmittel findet, das ein gültiges Angebot für die Beförderung machen kann, geht der Container zurück in den Vorrat, und wird sobald wie möglich erneut versteigert, ggf. zu einem höheren Maximalpreis. Im Anwendungsfall würde der Auktionator über eine Schnittstelle verfügen, über die Transportaufträge in Form von Container-Daten bzw. -Agenten in das System eingegeben werden können.

3.3.1 Container-Vorrat

Der Auktionator verwaltet die Container, bzw. die Transportaufträge, die ihm zur Versteigerung übermittelt wurden. Dabei hat er das Ziel jeden Container so häufig wie möglich zu versteigern ohne dabei andere zu vernachlässigen. Die naheliegendste Lösung besteht darin, die Container in einer FIFO-Liste zu halten. Im Falle einer Nichtversteigerung würde der entsprechende Auftrag dann einfach wieder ans Ende der Liste angehängt.

Um den Prozess zu optimieren könnte sich der Auktionator zunutze machen, daß sich die Transportagenten bei ihm mit möglichen Startorten für Aufträge registrieren (siehe nächster Abschnitt). Der Auktionator würde dann für die unmittelbar zur Auktion bevorstehenden Container ermitteln, ob es Agenten gibt, die auf mehrere Aufträge bieten würden. Sollte dies nicht der Fall sein, könnten die entsprechenden Auktionen parallel ablaufen. Dadurch würde die relative

Versteigerungshäufigkeit der Container erhöht.

Wird ein Container dem Vorrat entnommen, um angeboten zu werden, so wird der Auktionator dessen Lieferbedingungen und seinen momentanen Preis ermitteln, daraus ein Auftragsangebot formulieren und dieses an die entsprechenden Transporter verschicken.

3.3.2 Registrierung der Transporter

Prinzipiell könnte der Auktionator jeden Auftrag immer an jeden Transporter übermitteln. Es macht jedoch Sinn hier zu filtern. Aus Sicht der Transporter ist dabei das wesentliche Kriterium der Ort, an dem sie bereit sind Aufträge anzunehmen. Es hält einen Transporter nichts davon ab, sich beispielsweise für alle Städte zu registrieren. In dem Fall wäre eine Parallelisierung von Auktionen, wie sie im vorhergehenden Abschnitt angesprochen wurde, nicht möglich, da ein Transporter, da alle Auktionen zum gegebenen Zeitpunkt einen gemeinsamen Teilnehmer hätten, und damit streng sequentiell ablaufen müssten. Aus praktischer Sicht erscheint es sowohl aus Sicht des Transporters als auch aus Sicht des Auktionators sinnvoll, hier eine Grenze einzuführen. Die Registrierung für entfernte Orte bzw. Annahme von Aufträgen an denselben entspricht (zunächst) der Einplanung von Leerfahrten, was den Nutzen für den Transporter verringert. Der Transportagent würde also beispielsweise nur Aufträge annehmen, deren Startort maximal einen Knoten von seinem Verfügbarkeitsort entfernt ist.

Die Registrierung trägt ganz wesentlich zur Reduzierung von unnötiger Kommunikation bei. Dies ist im Hinblick auf die hohe Anzahl von Auktionen (mind. 10^2 pro Stunde) erforderlich.

3.3.3 Initiator-Protokoll

Der Auktionator ist für den größten Teil der Kommunikation verantwortlich, die für die Durchführung von Auktionen erforderlich ist. Er implementiert die Initiatorseite des angepassten Contract-Net-Protokolls (vgl. 3.2.6).

Benötigte Variablen und Konstanten:

```
Container_Vorrat;      // Liste mit Containern
Transporter_Register  // Transporter und deren Standorte
```

Verhalten Initiator_Protokoll {

```
    // entnimm einen Container aus der Liste
    Auftragsangebot = erstelle_Angebot(Container_Vorrat);

    // finde geeignete Transporter
    Auftragsangebot.Empfänger =
        finde_Transporter(Transporter_Register);

    sende Auftragsangebot;

    empfange Gebote;

    // ermittle den Gewinner der Auktion aus allen Geboten
    Auktionsergebnis = bestimme_Gewinner(Auftragsangebot, Gebote);

    // Benachrichtigung von Gewinner und Verlierer
    sende Auktionsergebnis;

    // Gibt es einen Gewinner, so wird der Container aus dem Vorrat
    // entfernt.
    falls Auktionsergebnis.Gewinner ist nicht null {
        Container_Vorrat.enferne(Auftragsangebot);
        verdiene_Gebühr(Auftragsangebot);
    }

    // Neustart des Verhaltens
    reset Initiator_Protokoll;
}
```

Abbildung 3.4: Initiator-Verhalten

3.3.4 Erhebung einer Gebühr

Das Geschäftsmodell des Auktionators besteht darin, die Partnerfindung zwischen Containern und Transportern zu begünstigen, um durch den sich durch automatisierte Auftragsvergabe erhöhten Nutzen auf beiden Seiten, einen Gewinn in Form einer kleinen Gebühr zu erwirtschaften. Für die Berechnung dieser Gebühr kann es verschiedenen Ansätze geben.

Das Problem, welches bei der Berechnung der Gebühr auftritt, ist, daß sofern sie sich in Abhängigkeit der abgegebenen Gebote berechnet, dies die Motivation zur Absprache zwischen den Transportern erhöht. Ergäbe sich die Gebühr beispielsweise aus der Differenz zwischen bestem und zweitbestem Gebot, wären die Transporter bestrebt, eng beieinander liegende Beträge zu bieten, da dies die Gebühr verringern würde. Natürlich muß man dabei berücksichtigen, daß die Absprache selbst ebenfalls Kosten verursacht, was gegebenenfalls dazu führt, daß sie sich nicht lohnt, insbesondere für Gebühren, die im Vergleich zu den Auftragskosten sehr gering sind, beispielsweise etwa 1 / 100.

Der naheliegendste Ansatz, der auch im Prototyp implementiert wurde, ist derjenige, die Gebühr vom aktuellen Maximalpreis für den Auftrag abhängig zu machen. Das heißt, der Auktionator zieht die Gebühr implizit vor der Versteigerung ab, indem er in der Formulierung des Angebots einen entsprechend niedrigeren Maximalpreis vorgibt. Für den Fall, daß der Container zu diesem Preis versteigert wird, ist die Gebühr für den Auktionator trotzdem gesichert. Für den (wesentlich häufigeren) Fall, daß der Preis unterhalb des Maximalpreises liegt, kann die Gebühr ebenfalls aufgeschlagen werden, da der sich ergebende Gesamtpreis inklusive Gebühr ebenfalls unter dem Maximalpreis befindet. In diesem Fall zahlt der Auftraggeber die Gebühr. Selbstverständlich wird die Gebühr nur bei erfolgreicher Versteigerung erhoben, d. h. Vertragsabschluß zwischen Container und Transporter. Im folgenden eine Abbildung, die qualitativ visualisiert, wie sich für den Auktionator und den Transporter eine Gewinnsituation ergeben kann.

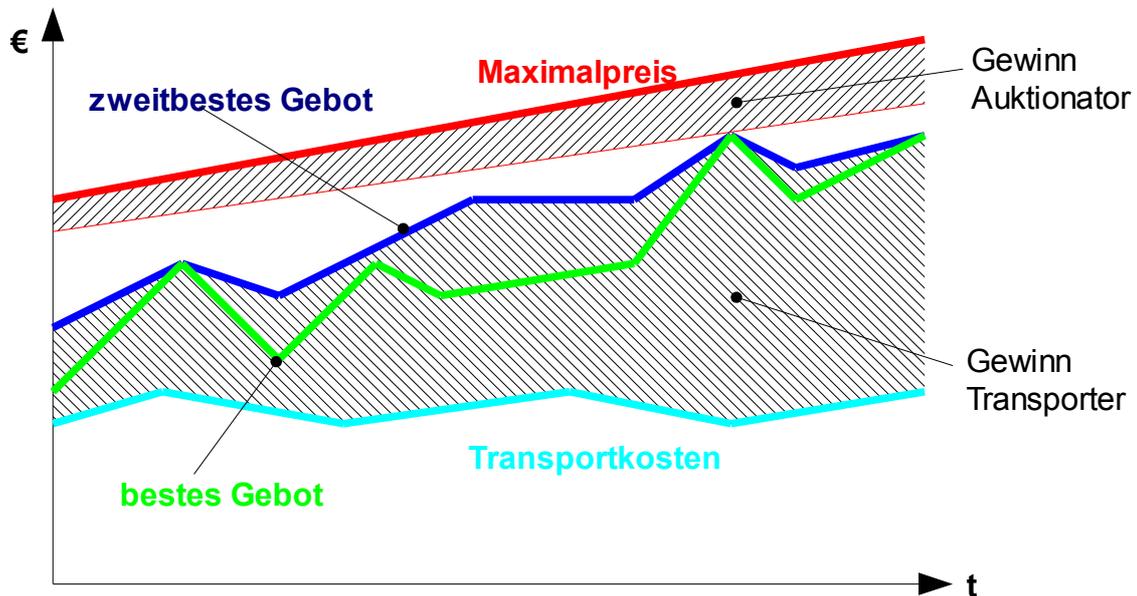


Abbildung 3.3: Preis, Gebote, Gewinn (qualitativ)

Die obige Abbildung stellt die hypothetische Abgabe von Geboten durch verschiedene Transport-Agenten im Laufe mehrerer Auktionen eines Containers dar. Eine Versteigerung findet zum Zeitpunkt t statt, mit gegebenen Werten für die Kosten, das beste und zweitbeste Gebot, sowie Maximalpreis und sich daraus ergebende Gebühr. Dabei kann es nur zu einer Versteigerung kommen, falls vorher keine erfolgreich war. Die im Laufe der Zeit steigenden Werte für die Gebote ergeben sich aus den Strategiefaktoren der Gebotsfunktion, von denen man annehmen kann, daß sie das Gebot erhöhen, je kurzfristiger die Planung ist. Dabei kann es allerdings Ausreißer nach oben oder unten geben. Ausreißer nach oben, d. h. bestes Gebot liegt nah am Maximalpreis, ergeben sich wenn es z. B. nur einen Bieter gibt, der „versucht“ zum Maximalpreis zu gewinnen. Ausreißer nach unten ergeben sich, wenn der Transporter durch Strategiefaktoren bedingt niedrig bietet (z. B. wenn er es günstig machen kann, da er sonst eine Leerfahrt hätte), wobei diese Ausreißer, bei denen das Gebot nahe der realen Kosten liegt, durch das zweitbeste Gebot „gedämpft“ werden (der Transporter erhält für niedrige Gebote in der Regel mehr als er bietet, bedingt durch das Vickrey-Prinzip).

4 Zusammenfassung und Ergebnisse

In diesem Kapitel werden noch einmal die Arbeitsschritte gezeigt, die zur Einführung des Auktionsmechanismus bzw. des Auktionators erforderlich waren. Dann wird ein qualitatives Ergebnis formuliert, sowohl anhand des Gesamtsystems, als auch im Hinblick auf die Erweiterbarkeit des Transportagenten, der mit seinen Eigenschaften die Grundvoraussetzung für intelligentes, zielorientiertes Verhalten besitzt.

4.1 Entwicklung des Auktionsprotokolls

In das System nach [Rosenfeld 2008] wurde ein Auktionator eingeführt, der Transportaufträge in Auktionen an Transportagenten vermittelt. Um dies zu ermöglichen wurde zunächst die Vickrey-Auktion als Protokoll implementiert. Dieses Protokoll setzt sich aus Initiator- und Responderseite zusammen. Für beide Seiten wurden Funktionen identifiziert, die zur Durchführung des Protokolls vorhanden sein müssen:

Funktionen des Initiators (Auktionator)

- Auswahl eines Containers aus dem Vorrat nach gegebenen Regeln (beispielsweise first-in-first-out)
- Ermitteln der Transportbedingungen inklusive Preis
- Suchen passender (entsprechend registrierter) Transportagenten
- Bestimmung des Gewinners anhand abgegebener Gebote

Funktionen des Responders (Transporter)

- Ermitteln von Verfügbarkeitsdatum und Erholtheit des Fahrers
- Ermitteln des Verfügbarkeitsortes
- Registrierung beim Auktionator
- Prüfung der Durchführbarkeit eines gegebenen Auftrags
(z. B. anhand einer erstellten Route)
- Kostenberechnung für einen gegebenen Auftrag
- Formulierung eines Gebotes, das sich aus dem Nutzen des Auftrags für den Agenten ergibt
- Planung eines Auftrags (nach Auktionsgewinn)

Das Protokoll kann für beide Seiten durch die o. g. Funktionen ausgedrückt werden. Es ist somit generisch und unabhängig von der jeweiligen Realisierung der Funktionen durch die das Protokoll verwendenden Agenten.

4.2 Entwicklung der Agenten

Durch Identifikation der für das Protokoll notwendigen Funktion, war es möglich die Agenten entsprechend zu implementieren bzw. zu erweitern.

Besondere Aufmerksamkeit kam dabei dem Transport-Agenten zu. Es war notwendig, Annahmen bezüglich der Gestalt seiner Kosten- und Gebotsfunktion zu machen. Dazu wurden Parameter eingeführt, die den Nutzen eines gegebenen Auftrags unter verschiedenen Gesichtspunkten in einen monetären Wert umwandeln. Beispielsweise wurde das Auftragsrisiko als Kombination aus Entfernung vom Heimatort und Folgeauftrags-Wahrscheinlichkeit ausgedrückt.

Um die Kosten überhaupt schätzen zu können, muß der Agent in der Lage sein eine Route zu ermitteln, die den Auftrag erfüllt. Dabei wurde festgestellt, daß

dazu gegebenenfalls Optimierung erfolgen kann, die jedoch vor allem für das Container-Szenario entfallen kann.

Mit JADE [jade.tilab.com], einem Java-basierten Rahmenwerk zur agentenorientierten Softwareentwicklung, wurde ein Prototyp des System entwickelt. Die Architektur ermöglicht es dabei, Änderungen bzgl. des Verhaltens der Agenten vorzunehmen, indem nur Funktionen wie beispielsweise die Gebotskonstruktion angepaßt werden, ohne dabei in den Mechanismus, d. h. die Protokolle, einzugreifen.

4.3 Qualitative Ergebnisse

Der Mechanismus wurde ausgehend von den individuellen Motivationen der im System identifizierten Akteure entworfen. Die durch den Mechanismus bedingte Preisfindung gewährleistet, daß die Motivation jedes Teilnehmers prinzipiell erfüllt werden kann.

Die Erfüllung der Motivation für den Auktionator ergibt sich in der Möglichkeit, daß er von jeder erfolgreich vorgenommenen Auktionen einen kleinen Anteil des Preises als Gebühr abführen kann. Sofern langfristig die erfolgreiche Vermittlung von Aufträgen gewährleistet ist, kann er auf dieser Grundlage einen Gewinn erwirtschaften.

Die Gebote für einen Auftrag durch die Transport-Agenten stellen ihre Wertschätzung für den Auftrag dar. Die Wahrscheinlichkeit zu gewinnen wird abhängig von der Höhe des Gebots sein. Da durch die Auktionsform gewährleistet ist, daß der Transporter mindestens den Betrag erhält, den er bietet, muß sein Ziel sein, den zu erhaltenden Betrag, also sein Gebot, immer so zu formulieren, daß die entstehenden Kosten bei Durchführung des Auftrages kleiner oder gleich dem Gebot sind. Er muß in der Lage sein seine (Folge-)Kosten gut schätzen zu können, wobei Unsicherheitsfaktoren eine Rolle spielen. Durch die Gestalt der Gebotsfunktion, können durch bestimmte Parameter die Auswirkungen der Unsicherheitsfaktoren gesteuert werden. Es ist auch möglich durch entsprechende Wahl der Parameter die Gebotsfunktion beispielsweise einer konstanten, entfernungsabhängigen Funktion anzunähern, was dem gegenwärtigen Geschäftsmodell von Logistik-Anbietern entspricht. Je mehr Erfahrung (beispielsweise durch Zuhilfenahme von Methoden des maschinellen Lernens) dem Transporter zu Verfügung steht, desto mehr wird er

in der Lage sein, seine Unsicherheitsfaktoren zu beziffern und bessere Gebote abzugeben.

Außerdem kann als Ergebnis festgehalten werden, daß der Transporter unter Umständen eine Motivation hat Informationen über andere Transporter und bevorstehende Auktionen zu erlangen. Mehr dazu findet sich in Kapitel 5.

5 Ausblick

5.1 Absprachen der Transporter

Der Auktionsmechanismus bewirkt, daß die am System teilnehmenden Transportagenten unter Umständen die Motivation haben, sich abzusprechen, um die Preisfindung durch das System zu beeinflussen. Die einfachsten Möglichkeiten zur Absprache bestehen darin, daß entweder alle Transporter immer den Maximalpreis bieten, oder daß sie sich einigen, daß nur einer an einer jeweiligen Auktion teilnimmt (und ggf. auch den Maximalpreis bietet). Die Konsequenz daraus wäre, daß immer nur maximale Preise gezahlt würden, was ein Vorteil für die Transporter ist.

Der wesentliche Grund, warum in einer realen Anwendung diese Art der Absprache kein Problem darstellen muß, liegt in der Natur der Zweitpreis-Auktion. Denn sobald es einen (einzigen) Transporter gibt, der sich nicht an die Absprache hält, so wird dieser in der Regel seine Konkurrenten unterbieten und, da diese auch noch den Maximalpreis geboten haben, einen besonders guten Preis bekommen, und zudem mit 100%-iger Sicherheit die Auktion gewinnen. Das bedeutet wiederum, in dem Moment, in dem sich alle Transporter zu einer Absprache entschlossen haben, das Nichteinhalten der Absprache den größeren Erwartungswert hat, als sich daran zu halten. Ein weitere Aspekt ist es, daß die Absprache ebenfalls Kosten verursacht, insbesondere in Bezug auf das Gewinnaufteilungs-Problem. Um das in dieser Arbeit entwickelte System für reale Aufträge verwenden zu können, muß gewährleistet sein, daß der erwartende durchschnittliche Gewinn durch die automatische, nutzenbasierte Vermittlung von Aufträgen den durch Manipulationsversuchen entstehendem Gewinn deutlich übertrifft.

5.2 Sinnvolle Gebote und Auftragsdaten

Wie bereits erwähnt liegt die Motivation auf Seiten des Transporters darin, Aufträge zu einem Preis durchzuführen, der immer oberhalb seiner Kosten liegt. Dazu müssen die Kosten unter Berücksichtigung von Unsicherheitsfaktoren geschätzt werden. Es gilt zu ermitteln wie sich die Faktoren der Gebotsfunktion die diese Unwägbarkeiten ausdrücken gewählt und angepaßt werden können, damit die Gebote sinnvoll sind. Es könnte auch eine andere Gebotsfunktion gewählt werden, die andere Faktoren berücksichtigt. Der Grundgedanke ist jedoch der, daß gegebenenfalls nicht-monetäre Faktoren in das Gebot einbezogen werden.

Um sinnvolle Gebotsfaktoren zu ermitteln ist es notwendig, daß realistische Daten bezüglich Kosten von Transport und Preise für Aufträge ermittelt werden, und verschiedene Strategien erprobt werden. Die Strategie besteht in der Wahl der Faktoren für die Gebotsfunktion und aus Regeln, wie diese im Fall von Gewinn oder Verlust von Auktionen, sowie dem momentanen Standort und der zukünftigen Planung angepaßt werden

5.3 Erweiterbarkeit des Systems

Die Architektur der Agenten ist auf Erweiterbarkeit ausgelegt. Unabhängig vom Kommunikationsprotokoll können die im Protokoll verwendeten Funktionen innerhalb der Agenten erweitert werden. Das würde zu anderem Verhalten führen. Ein Beispiel sei die Registrierung beim Auktionator durch die Transport-Agenten. Einfache Möglichkeiten bestehen in der Registrierung mit nur einem Ort, dem momentanen Aufenthaltsort. Oder es werden zusätzlich alle Nachbarorte angegeben. Fortgeschrittenere Methoden bestehen beispielsweise darin Orte für die Auftragsannahme entfernungsabhängig anzugeben (d. h. alle Nachbarorte, sofern sie nicht weiter entfernt sind als x km), oder die Auftragshäufigkeit mit einzubeziehen. Sofern man sich für die generelle Verwendung des Protokolls entscheidet, hat man die Möglichkeit, die Auswirkungen von verschiedenen Implementierungen der identifizierten Teilfunktionen zu prüfen.

5.4 *Transparenz*

Einer letztlich Frage, die für praktische Anwendungen von großer Bedeutung ist, ist die der Transparenz. Damit ist gemeint, welche der innerhalb des Systems verwendeten Datenstrukturen welchen Teilnehmern verfügbar gemacht werden. Beispielsweise beruht die Gebotsabgabe eines Transport-Agenten auf Unkenntnis über andere Transporter und deren Aufenthaltsorte und Planung. Würde ein Transporter über die Registrierungsdaten des Auktionators verfügen, so könnte er seine Gebote anpassen und Rückschlüsse über Routenplanung und sogar die Kostenfunktion der anderen Transporter ermitteln. Eine Frage in diesem Zusammenhang ist beispielsweise, ob es sinnvoll sein könnte, die entsprechenden Daten allen Transportern zugänglich zu machen, ohne dabei die Interessen der anderen Systemteilnehmer zu kompromittieren.

Grundsätzlich ist aber immer die Frage inwieweit, aus Sicht der Transport-Agenten, es sinnvoll ist alle zur Verfügung stehenden Daten der anderen Teilnehmer einbeziehen zu wollen. Denn dies stellt zweifellos eine optimierungstechnisch sehr anspruchsvolle Aufgabe dar. Die Kosten für die laufende Auswertung und Optimierung der Daten müssen dabei durch den womöglich entstehenden Vorteil wieder kompensiert werden. Unter Umständen würde der Transporter es sogar vorziehen, sich bei seiner Planung und Gebotsberechnung lediglich die eigene Situation und Einschätzung von Kosten und Nutzen beziehen.

Literaturverzeichnis

[Rosenfeld 2008]

Entwurf und Realisierung eines autonomen Agentensystems zur
Transportsteuerung

Technische Universität Hamburg-Harburg, 2008

[Dorer/Calisti 2005]

An Adaptive Solution to Dynamic Transport Optimization

Whitestein Technologies AG, 2005

[Vickrey 1961]

Counterspeculation, Auctions, and Competitive Sealed Tenders

The Journal of Finance, Vol. 16, No. 1, 1961

[Seifert/Strecker 2003]

A Note on over- and underbidding in Vickrey auctions: Do we need a new
theory?

Universität Karlsruhe, 2003

[Hirsch 2007]

Entwicklung eines Multi-Agenten-Systems

Technische Universität Hamburg-Harburg, 2007

[www.frachtprofi.at]

Online Dokumentation zum Frachtprofi-Softwaresystem

Phönix Software, 2008

[www.fipa.org]

IEEE Foundation for Intelligent Physical Agents, 2009

[jade.tilab.com]

Java Agent DEvelopment Framework

Telecom Italia Lab, 2009