

***ComEx*: Kombinatorische Auktionen zum innerbetrieblichen Austausch von Logistikdienstleistungen**

Oleg Gujo, Michael Schwind, Jens Vykoukal, Kilian Weiß

Institut für Wirtschaftsinformatik
Johann Wolfgang Goethe Universität Frankfurt
60056 Frankfurt am Main
{gujo, schwind, jens.vykoukal, weiss}@wiwi.uni-frankfurt.de

Tim Stockheim, Oliver Wendt

Lehrstuhl für Wirtschaftsinformatik und Operation Research
TU Kaiserslautern
67653 Kaiserslautern
{stockheim,wendt}@wiwi.uni-kl.de

Abstract

Der Austausch von Ladekapazitäten in Transportbörsen ist ein etabliertes Verfahren der logistischen Praxis. Wenige der meist Web-basierten Marktplätze sind jedoch in der Lage, Synergien zu berücksichtigen, die durch geeignete Kombination von Lieferstrecken verschiedener Transportanbieter entstehen. Eine Methode hierfür ist die Verwendung kombinatorischer Auktionen, bei denen auf Bündel von Streckenabschnitten geboten werden kann. Der Beitrag beschreibt eine kombinatorische Auktion zum innerbetrieblichen Austausch von Logistikdienstleistungen. Dabei werden Lieferaufträge für ein in Profitcenterstruktur organisiertes Unternehmen der Nahrungsmittelindustrie zum Outsourcing vom jeweiligen Profitcenter freigegeben, wenn die geografische Lage des Kunden eine günstigere Belieferung durch ein anderes benachbartes Profitcenter zu günstigeren Kosten ermöglicht. Umgekehrt können Profitcenter Aufträge annehmen, wenn dadurch insgesamt niedrigere Gesamtkosten entstehen. Die Kostenberechnung wird durch ein integriertes Routingsystem vorgenommen, das In- und Outsourcing findet mittels des Auktionsmechanismus *ComEx* statt. Die Aufträge verbleiben zum Zwecke der Kundenbindung beim jeweiligen Profitcenter, der Anreiz zum Tausch entsteht durch eine entsprechende Beteiligung der Profitcenter an der Kostenersparnis. Nach der Darstellung der Web-basierten Logistikauktion zusammen mit der Tourenplanungssoftware *DynaRoute* suchen wir die Strategie für das Outsourcing der gebündelten Lieferaufträge, welche die größte Kostenreduzierung erzielt.

1 Einleitung

Seit einigen Jahren gewinnt die Anwendung von kombinatorischen Auktionen (KA) für Beschaffungs- und Ressourcenallokationsprozesse an Bedeutung. Getrieben durch die Entwicklung von Mechanismen für die Vergabe von Frequenzbändern an TK-Unternehmen in England¹, Deutschland und den USA² gelangte die KA in den Fokus des *Electronic Market Engineering* [McMi95]. Der vorliegende Beitrag beschäftigt sich mit einer elektronischen KA zum Austausch von Lieferleistungen in einem mittelständischen Unternehmen, welches in Profitcenterstruktur organisiert ist. Dabei haben die Profitcenter, denen regionale Liefergebiete zugeordnet sind, die Möglichkeit, Lieferaufträge, welche nur zu ungünstigen Kosten beliefert werden können, an benachbarte Profitcenter abzugeben. Die erzielte Kostenersparnis wird zum Zwecke der Anreizkompatibilität nach einem vorher festzulegenden Schlüssel auf die Profitcenter umgelegt. Kernpunkt dieser Arbeit ist die Frage, wie eine Bündelung auszulagernder Lieferaufträge vorgenommen werden kann, so dass das Zusammenwirken von KA und lokaler Routenoptimierung der Profitcenter unter Berücksichtigung der Lieferzeiten kostenoptimal wird.

2 Kombinatorische Auktionen in der Logistik

Für den Bereich der KA in der *Logistik* ist der Ansatz von Caplice [Cap196; CaYo03] für eine mit der Streckenplanung kombinierten Zuweisung von Transportkapazitäten in einem Logistiknetzwerk wegweisend. Dieser verbindet die Bildung der kostengünstigsten Transportkombination durch einzelne Transportanbieter mittels einer KA mit einer einfachen Tourenplanung. Einen ähnlichen Ansatz verfolgen auch Regan und Song [ReSo03]. Sie schlagen einen Spotmarkt für überschüssige bzw. kurzfristig benötigte Transportdienstleistungen vor. Kameshwaran und Narahari [KaNa01] stellen eine sequentielle KA zur Lösung eines Tourenkompositionsproblems vor. In diese Richtung geht auch der Ansatz von Pankratz [Pank03], bei dem die Anreizkompatibilität für Bieter und Nachfrager im Vordergrund steht. Einige Anbieter von Logistiksoftware und Frachtbörsenbetreiber haben bereits einfache KAs in ihre Routenplanungs- und Allokationsverfahren einbezogen³. Im Bereich der Transportdienstleistungen werden seit 1995 die Busrouten der Londoner Region von der *London Regional Transport* per KA versteigert

¹ <http://www.ofcom.org.uk/static/archive/spectrumauctions/index.htm>

² <http://www.fcc.gov>

³ <http://www.nex.com>, <http://www.i2.com>, <http://www.logistics.com>, www.benelog.com, www.combinenet.com

[CaPe05]. Elmaghraby und Keskinocak [ElKe05] dokumentieren die Durchführung einer zwei-stufigen KA zu Beschaffung von Transportkapazitäten durch die US-Baumarktkette *Home Depot* zur Versorgung von über tausend Filialen. Zusammen mit *i2-Technologies* entwickelte das Unternehmen einen flexiblen Auktionsmechanismus, der den Logistikanbietern helfen sollte, ihre individuellen Synergien zwischen Routen mit Hilfe kombinatorischer Gebote auszudrücken. Dabei werden die Bieter durch eine Software bei der Formulierung der Gebote unterstützt, denn die Formulierung der Gebotsbündel und ihre Bewertung stellen ein kombinatorisches Optimierungsproblem dar, das normalerweise nicht manuell gelöst werden kann [Schw05]. Aus diesem Grund ist das hier vorgestellte *ComEx*-System so gestaltet, dass die In- und Outsourcing-Entscheidungen zusammen mit der Gebotsbündelbildung und deren Bewertung mit Hilfe des *DynaRoute*-Tourenoptimierers von *VARLOG*⁴ automatisch durchgeführt werden.

3 Funktionsprinzip und Systembeschreibung von *ComEx*

Im Folgenden geben wir einen Überblick über das Prinzip der KA und stellen die *ComEx*-Systemarchitektur und den Ablauf der Auktions- bzw. Optimierungsprozesse dar. Abb. 1 skizziert die Vorgehensweise für den Austausch von Dienstleistungen zwischen 2 Profitcentern:

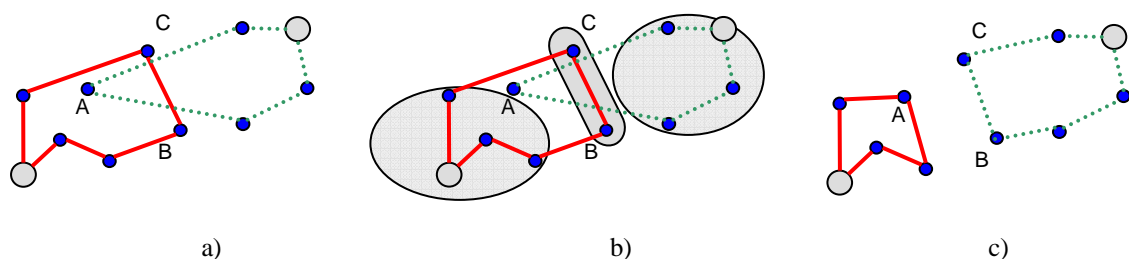


Abb. 1: Austausch von Kunden zwischen benachbarten Profitcentern

Zur Vereinfachung wird angenommen, dass jedes Profitcenter über eine Teilmenge von Kunden verfügt, die bereits einer Lieferroute zugeordnet ist (Abb. 1a). Für jedes Profitcenter wird basierend auf der geografischen Nähe zum Profitcenter entschieden, welche der Kunden weiterhin beliefert werden sollen. Diese Kunden werden dem so genannten Fixbereich zugeordnet (graue Kreise in Abb. 1b). Die übrigen Kunden werden unter Berücksichtigung der geografischen Lage und der Lieferzeiten zu Clustern zusammengeführt (Abb. 1b Punkte B und C). Für jedes Cluster werden die Kosten ermittelt, die das Profitcenter im Falle der Abgabe aller Kunden dieses Clusters an ein anderes Profitcenter einsparen würde. Es findet ein Informationsaustausch zwi-

⁴ <http://www.varlog.de>

schen den Profitcentern statt, bei dem die Outsourcing-Kandidaten benachbarter Profitcenter als Insourcing-Kandidaten des jeweiligen zentralen Profitcenters deklariert werden. Für jedes Cluster wird sodann von den jeweiligen Profitcentern untersucht, ob die Insourcing-Kandidaten zeitlich und geografisch zu der bestehenden Kundenmenge passen. Ist dies der Fall, werden die Kosten für die Belieferung der neuen Kundenmenge ermittelt. Diese Kosten werden als Gebotspreise für die *ComEx*-Auktion gesetzt. Nachdem jedes Profitcenter seine Gebote abgegeben hat, findet eine KA statt, um eine optimale Allokation von Geboten mit minimalen Lieferkosten im Gesamtsystem zu finden. Nach der Auktion erfährt jedes Profitcenter, welche Kunden von ihm beliefert werden. Die Lieferrouten der Profitcenter werden neu ermittelt (Abb. 1c).

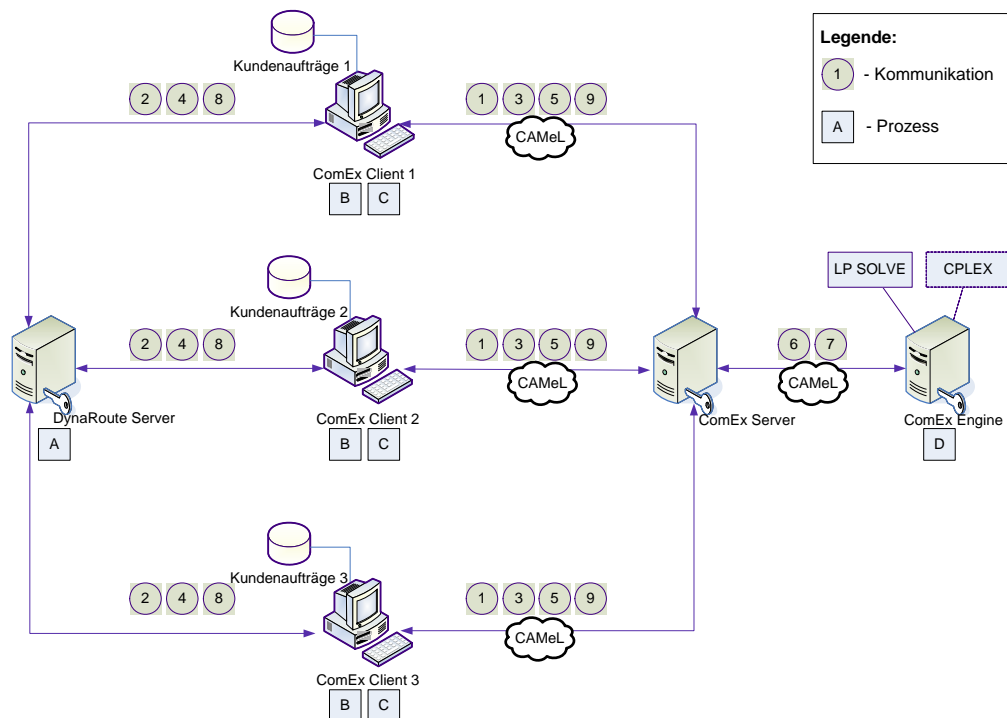


Abb. 2: Kommunikation und Prozessablauf im ComEx-System

Das *ComEx*-System besteht aus den folgenden Komponenten:

- *ComEx*-Client: Jedem Profitcenter steht ein *ComEx*-Client zur Verfügung, der alle Kunden im jeweiligen Liefergebiet verwaltet und mit Hilfe der Kostendaten vom *DynaRoute*-Server festlegt, welche Kunden in den In- und Outsourcing-Prozess einbezogen werden (in Abb. 2 sind das die *ComEx*-Clients 1 bis 3). Zudem wird die Bündelung von Kundenaufträgen zu kombinatorischen Geboten vom *ComEx*-Client durchgeführt. Das Clustering wird in Abschnitt 3.2.1 genauer erläutert.
- *DynaRoute*-Server: Er ist für die Zusammenstellung der Touren und die Schätzung der damit verbundenen Lieferkosten innerhalb eines Profitcenters zuständig.

- *ComEx*-Server: Dieser steuert den Ablauf des gesamten Auktionsprozesses. Dabei sammelt der *ComEx*-Server die In- und Outsourcing-Gebote aller Profitcenter und stößt die Berechnung der optimalen Allokation durch die *ComEx*-Engine an.
- *ComEx*-Engine: Sie ist für die Berechnung der optimalen Allokation zuständig.

Im Folgenden wird der Ablauf einer *ComEx*-Auktion im Zusammenspiel mit den beteiligten Systemkomponenten dargestellt. Insgesamt sind vier Hauptphasen zu unterscheiden⁵:

Initialisierungsphase:

- In der Phase werden Einstellungen wie das Auktionsformat, Anmeldungs- und Zulassungsdaten der Teilnehmer usw. übermittelt ^①. Dabei kommt ein spezielles XML-Format *CAMeL* für die Übertragung zum Einsatz (siehe auch Abschnitt 3.3).

Outsourcingphase:

- Für jedes Profitcenter werden die Kunden unter Berücksichtigung der zeitlichen (ob die Belieferung der Kunden nacheinander möglich ist) und räumlichen (ob die Kunden sich nah zueinander befinden) Bedingungen in Cluster zusammengeführt. Die Outsourcingkandidaten werden, basierend auf ihrer Entfernung vom Zentrum des Profitcenters, ermittelt ^{□B}. Hierauf wird im Abschnitt 3.2.1 näher eingegangen.
- Für jedes der so erzeugten Cluster wird von den Clients eine Anfrage ^② an den *DynaRoute*-Server gestellt, um die Kosten zu ermitteln ^{□A}, die sich durch die Auslagerung einer Kundengruppe einsparen lassen (siehe auch Abschnitt 3.1).
- Die als Outsourcing-Kandidaten bestimmten Cluster werden von jedem Profitcenter an den *ComEx*-Server weitergeleitet. Die Profitcenter erhalten jeweils vom *ComEx*-Server eine Liste aller durch die benachbarten Profitcenter zum Outsourcing freigegebenen Kundencluster, da diese Insourcing-Kandidaten darstellen ^③.

Insourcingphase:

- Anhand der erzeugten Kandidatenliste wird für jedes Profitcenter untersucht, ob sich die jeweiligen Kunden unter Berücksichtigung der zeitlichen und räumlichen Bedingungen zu eigenen Kunden im Outsourcingbereich hinzufügen lassen ^{□C}. Ist

⁵ Das Zusammenspiel zwischen *ComEx* und *DynaRoute* ist ein kritischer Faktor für die Gesamtperformance und sollte bei der Weiterentwicklung des Systems adressiert werden.

dies der Fall, wird eine weitere Anfrage an den *DynaRoute*-Server gestellt ^④, um herauszufinden, zu welchen Konditionen die Belieferung der neuen Kunden erfolgen kann [Ⓐ]. Diese Kosteninformation wird als Gebotspreis des Clusters gesetzt und das entstehende Insourcing-Gebot wird an den *ComEx*-Server weitergeleitet ^⑤.

- Der *ComEx*-Server sammelt die Gebote aller Profitcenter und initiiert die KA durch Aufruf der *ComEx*-Engine ^⑥. Eine optimale Allokation wird ermittelt [Ⓓ] und die Ergebnisse werden den Profitcentern mitgeteilt ^⑦.

Endbewertungsphase:

- Für alle Profitcenter werden die Routenpläne basierend auf der neuen Kundenumenge aktualisiert ^⑧ [Ⓐ], und die Ersparnis im Unternehmen wird ermittelt ^⑨.

3.1 *DynaRoute*-Server

Die Tourenberechnung bei den Clients findet mittels der Software *DynaRoute* [WeSW05] statt und löst Erweiterungen des „*Time Dependent-Stochastic Capacitated Vehicle Routing Problem with Time Windows*“ (TD-CVRP-TW). Die hohe Komplexität des Problems entsteht aufgrund der Ergänzung des bereits NP-vollständigen „*Vehicle Routing Problems*“ um zeitabhängige Fahrzeiten und der Einführung von Zeitfensterkategorien. Die Zeitfenster sind in die vier Kategorien niedrig, mittel, hoch und strikt eingeteilt, aus denen eine Bewertungsfunktion für die prognostizierte Nichteinhaltung der Zeitfenster abgeleitet wird.

Grundlage des Lösungsverfahrens ist die von [Wend95] vorgeschlagene Kombination der Metaheuristik Simulated Annealing (SA) und Genetischer Algorithmen (GA). Cooperative Simulated Annealing (COSA) übernimmt dabei die Populationsidee und die Idee des Informationsaustausches zwischen einzelnen Individuen von GAs und kombiniert diese mit der SA-Heuristik. Diese steuert lokale Suchoperatoren entsprechend stochastischer Annahmeregeln für neue Lösungen. Im Laufe der letzten Jahre wurde das Lösungsverfahren an die Anforderungen der Praxis angepasst und es wurden folgende Ergänzungen eingeführt:

- *Lösungsraumkompression*

Viele Teillösungen können mit hoher Wahrscheinlichkeit als nicht optimal identifiziert werden, und ihr Ausschluss ermöglicht eine Komplexitätsreduktion. Während entsprechende Lösungsbereiche bei bekannten Optimierungskriterien und Restriktionen häufig mit Hilfe einfacher Verfahren ausgeschlossen werden können, ist dies hier nicht

möglich. *DynaRoute* nutzt auf Ant-Systemen [BuHS99] basierende Verfahren zur Identifikation stochastischer Muster, die eine effiziente Steuerung der Suche ermöglichen. Weiterhin können im Rahmen der Restriktionsspezifikation Abbruchkriterien für die Lösungsbewertung definiert werden, die den Lösungsraum weiter einschränken.

- *Tabu-Listen*

Tabu-Listen [Glov86] erlauben die temporäre Sperrung bereits analysierter Lösungen, bzw. Lösungsregionen und damit die gezielte Diversifikation des Suchprozesses. Die Nutzung von Tabu-Listen im Rahmen der SA-Suche hat sich als ein außerordentlich effizienter Ansatz erwiesen [LiLi02].

- *Approximationsverfahren zur Lösungsbewertung*

Die exakte Bewertung einer neuen Lösung ist häufig nicht notwendig. Entsprechend helfen Approximationsverfahren, die Rechenzeit zu reduzieren, ohne die Lösungsqualität zu senken. *DynaRoute* nutzt dynamische Approximationsverfahren, die in Abhängigkeit des Suchverlaufs den Detaillierungsgrad der Berechnungen intensivieren.

Basierend auf den einzelnen Lösungsbewertungen durch die Heuristik werden Bewertungen der von *ComEx-Client* spezifizierten Teilmengen vorgenommen. Der skizzierte Lösungsansatz für das Tourenplanungsproblem bietet die Möglichkeit, nach der geringfügigen Veränderung eines Tourenplans diesen durch kurzes Reaktivieren des Berechnungsprozesses schnell zu optimieren. Ist dies geschehen, ermöglicht ein einfacher Vergleich der Kosten der Ausgangslösung und der durch Hinzufügen neuer Kunden veränderten Lösung eine Aussage über die Kosten der Integration neuer Kunden. Das Risiko einer Fehlbewertung durch den Vergleich suboptimaler Lösungen ist auf Grund des populationsbasierten Algorithmus gering, da der Vergleich über die k -besten Lösungen der beiden Populationen einzelne Fehler relativiert.

3.2 *ComEx-Client*

Jedes Profitcenter entscheidet, welche Kunden es auf jeden Fall selbst beliefern will und welche von anderen Profitcentern übernommen werden können. Im ersten Fall geht es um die Kunden, die geografisch nah am Profitcenter i liegen. Solche Kunden bilden, wie bereits oben beschrieben, den Fixbereich P_i^f . Alle anderen Kunden werden dem Outsourcingbereich P_i^o zugeordnet.

3.2.1 *Clustering und Outsourcing*

Das Ziel des Clusterings besteht darin, die Komplexität des vorliegenden KA-Problems zu verringern und Synergieeffekte zwischen einzelnen Lieferaufträgen zu identifizieren. Um dies zu erreichen, werden die Kunden nach bestimmten zeitlichen, räumlichen und kapazitiven Kriterien bewertet und anschließend zu *Clustern* $C_i^y \subseteq C_i$, $i, y \in \mathbb{N}$ gruppiert, wobei C_i die Menge aller Cluster des Profitcenters i darstellt. Ein Cluster wird dabei als ein ungerichteter Pfad verstanden, der ein Teilstück einer Route bilden kann, welche durch die spätere Routenplanung bestimmt wird. Das Clustering verläuft also unabhängig von der eingesetzten Routenplanung und hat die Aufgabe, geografisch nahe gelegene Kunden so zusammenzufassen, dass sie unter Einhaltung von oben erwähnten Kriterien mit einem Fahrzeug beliefert werden können.

Zu Beginn des eingesetzten Clustering-Verfahrens enthält ein Pfad jeweils einen Kunden. Um möglichst viele Pfade zu kombinieren, werden zwei verschiedene Strategien eingesetzt, die in aufeinander folgenden Phasen stattfinden. Während in der ersten Phase versucht wird, Pfade durch eine Verbindung an deren Anfang oder deren Ende mit einem anderen Pfad zu verlängern, wird in der zweiten Phase überprüft, ob sich Pfade in andere eingliedern lassen.

Sei E die Menge der Kanten, V die Menge der Knoten in einem ungerichteten Graphen G , $e_{new} = \{a_k, a_l\}$ mit $a_k, a_l \in V, k \neq l$ ein Kundenpaar mit Distanz $|e_{new}| \in \mathbb{R}^+$ zwischen den Kunden in einem Profitcenter i . Um den Algorithmus der ersten Phase zu veranschaulichen, ist in Abb. 3 dessen Pseudocode dargestellt:

ClusteringInclude

- 1 $E' := \{e_{new} = \{a_k, b_l\} \mid a_k \in C_i^a \wedge b_l \in C_i^b \wedge e_{new} \in E \wedge |e_{new}| \leq r\};$
- 2 Sortiere E' nach steigenden Distanzen $|e_{new}|$;
- 3 **for all** $\{a_k, b_l\} \in E'$ (*in sortierter Reihenfolge*) **do**
- 4 **if** $((a_k \text{ Endknoten} \wedge b_l \text{ Endknoten}) \wedge (a_k \text{ und } b_l \text{ nicht im selben Cluster}))$
- 5 Verbinde Cluster C_i^a und C_i^b durch e_{new} ;
- 6 **if** (Kombiniertes Cluster valide)
- 7 Speichere kombiniertes Cluster C_i^y ;

Abb. 3: Clustering-Algorithmus zum Zusammenlegen von Streckenabschnitten (Phase 1)

Zunächst wird die Menge E' aller möglichen Verbindungen zwischen zwei Kunden a_k und a_l eines Profitcenters bestimmt. Dabei werden nur die Kunden berücksichtigt, deren Distanz zueinander kleiner ist als eine maximale Entfernung r (Zeile 1). Die Kundenpaare werden nach aufsteigender Entfernung sortiert (Zeile 2) und der Reihe nach betrachtet (Zeile 3). Falls beide Kunden a_k und b_l Cluster-Endpunkte sind und sich zugleich nicht im selben Cluster befinden

(Zeile 4), wird ein kombiniertes Cluster erzeugt, indem zwischen a_k und b_l eine Kante e_{new} eingefügt wird (Zeile 5). Anschließend wird die Validität des neu erstellten Clusters überprüft, die genau dann gegeben ist, wenn die im Cluster enthaltenen Kunden in mindestens einer der beiden Reihenfolgen unter Einhaltung der Lieferzeitfenster von einem Fahrzeug beliefert werden können. Falls Validität vorliegt, wird das Cluster C_i^y abgespeichert (Zeilen 6, 7).

Die zweite Phase zielt darauf ab, bestehende Cluster in andere einzugliedern. Dies bedeutet, dass der Pfad innerhalb eines Clusters unterbrochen wird, um die Kunden eines anderen Clusters zu beliefern. Erst wenn dies geschehen ist, wird die Abarbeitung des ursprünglichen Pfades fortgesetzt. Für diese Eingliederung eines Clusters in ein anderes sind zwei verschiedene Varianten denkbar, die in Abb. 4 dargestellt sind. Zunächst wird eine Verknüpfung zwischen einem Endknoten des ersten Clusters mit dem inneren Knoten des zweiten Clusters festgelegt (Abb. 4b). Nun wird der Pfad des zweiten Clusters zwischen a_2 und a_3 (Abb. 4a) bzw. a_1 und a_2 (Abb. 4c) unterbrochen und zwei mögliche Verbindungen werden bestimmt, die die Eingliederung des ersten Clusters ($\{b_1, b_2\}, \{b_2, b_3\}$) in das zweite ermöglichen. Analog zur ersten Phase dürfen auch hier die neu entstehenden Verbindungen die Kantenlänge r nicht übersteigen.

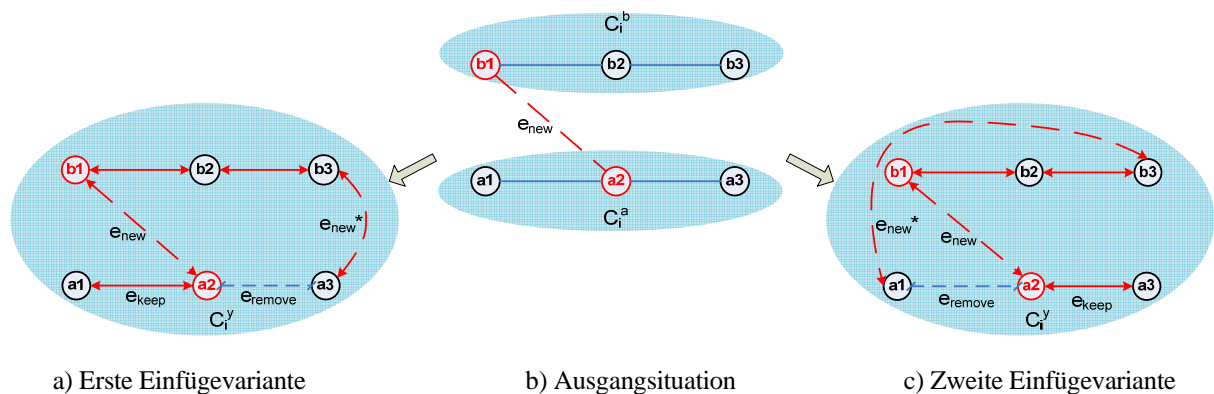


Abb. 4: Zwei Varianten für die Clusterzusammenlegung

In Abb. 5 wird der Algorithmus der zweiten Phase dargestellt. Zunächst wird eine Menge E'' generiert, die anfangs leer ist (Zeile 1). Dafür wird jedes Element der Menge E' daraufhin geprüft, ob es als erste der beiden Verbindungen zwischen zwei verschiedenen Clustern in Frage kommt (Zeilen 2 und 3). Dies ist genau dann der Fall, wenn sich die beiden Kunden a_k und b_l nicht bereits im selben Cluster befinden und wenn einer der beiden Kunden einen Cluster-Endpunkt darstellt, der andere jedoch nicht (Zeile 4). Nach der Bestimmung der beiden Varianten für die zweite Verbindung werden die Kantentupel zur Menge E'' hinzugefügt (Zeilen 5, 6).

ClusteringInclude2

```
1    $E'' := \emptyset$  ;
2    $E' := \{e_{new} = \{a_k, b_l\} \mid e_{new} \in E \wedge a_k \in C_i^a \wedge b_l \in C_i^b \wedge |e_{new}| \leq r\}$  ;
3   for all  $e_{new} = \{a_k, b_l\} \in E'$  do
4       if  $((a_k \text{ Endknoten} \wedge b_l \neg \text{Endknoten}) \wedge (a_k \text{ und } b_l \text{ nicht im selben Cluster}))$ 
5           for all  $e_{keep} = \{b_l, b_{l^*}\}$  mit  $l \neq l^* \wedge b_l \in e_{new}$  do
6               Bestimme  $e_{new}^*, e_{remove}, \Delta = |e_{new}| + |e_{new}^*| - |e_{remove}|$ ,  $E'' \leftarrow (e_{new}, e_{keep})$  ;
7       Sortiere  $(e_{new}, e_{keep}) \in E''$  nach steigendem  $\Delta$  ;
8       for all  $(e_{new}, e_{keep}) = (\{a_k, b_l\}, \{b_l, b_{l^*}\}) \in E''$  (*in sortierter Reihenfolge*) do
9           if  $(a_k, b_l, b_{l^*}$  nicht im selben Cluster)
10              Gliedere  $C_i^a$  gemäß der Kanten  $e_{new}, e_{new}^*$  und  $e_{keep}$  in  $C_i^b$  ein;
11              if (Kombiniertes Cluster valide)
12                  Speichere kombiniertes Cluster  $C_i^y$  ;
```

Abb. 5 Clustering-Algorithmus zum Eingliedern von Streckenabschnitten (Phase 2)

Anschließend werden die Elemente der Menge E'' nach steigenden Distanzsummen sortiert (Zeile 7) und es wird der Reihe nach geprüft, ob die beiden repräsentierten Verbindungen zu einer validen Eingliederung eines Clusters in ein anderes Cluster führen (Zeilen 8 bis 10). Wenn die Validität des kombinierten Clusters gegeben ist, wird dieses Cluster abgespeichert (Zeilen 11 und 12).

Sei $I \in \mathbb{N}$ die Anzahl von Profitcentern im Unternehmen. Für jedes $i \in I$ werden die Kunden entsprechend ihrer Entfernung vom Zentrum des Profitcenters in zwei Gruppen eingeteilt: Ein bestimmter Prozentsatz der weiter entfernten Kunden wird dem Outsourcingbereich P_i^o zugeordnet, die übrigen näher am Zentrum liegenden Kunden bilden den Fixbereich P_i^f . Innerhalb jedes Profitcenters i wird die Menge C_i gebildet, die alle im Profitcenter erstellten kombinierten Cluster in P_i^o enthält. Des Weiteren wird C_i an *ComEx*-Server weitergeleitet und dort zu $C^* = \bigcup_{i=1}^I C_i$ gruppiert. Anschließend erhalten alle Profitcenter die Menge C^* .

3.2.2 Gebotserstellung und Insourcing

In der Literatur sind verschiedene Gebotssprachen und damit verbundene Arten der Gebotslogik beschrieben [Nisa99]. Das *ComEx*-System benutzt eine OR-of-XOR Gebotslogik. Dies bedeutet, dass die XOR-Gebotsbündel jeweils eine Auswahl eines der alternativen Gebote erlauben, während bei den OR-Gebotsbündeln mehrere Gebote gleichzeitig befriedigt werden können.

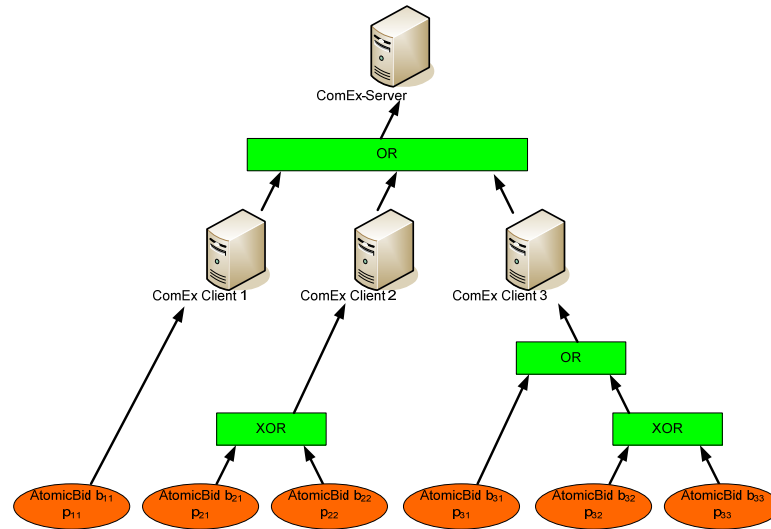


Abb. 6 Gebotslogik in ComEx

Auf Grund seiner Größe wird C^* von jedem Profitcenter anhand eines von uns entwickelten Verfahrens zu C_i^* verkleinert, indem die am weitesten entfernt liegenden kombinierten Cluster anderer Profitcenter eliminiert werden. Die Menge C_i^* wird von Profitcentern für die Erstellung der Kombinationen aus eigenen kombinierten Clustern mit kombinierten Clustern anderer Profitcenter (nach dem gleichem Verfahren wie in 3.2.1) verwendet. Anschließend werden die Gebote wie folgt gebildet: Sei $M_i \in \mathbb{N}$ die Anzahl der für Profitcenter $i \leq I$ zur Verfügung stehenden kombinierten Cluster $C_i^y \in C_i^*$, dann ist $b_{ij} = C_i^\alpha \wedge C_i^\beta \wedge \dots \wedge C_i^\delta$, $\alpha \neq \beta \neq \delta \wedge \alpha, \beta, \delta \leq M_i$ ein *AtomicBid* vom Profitcenter i . Für jedes b_{ij} wird eine Anfrage an den *DynaRoute*-Server gestellt, um die Kosten p_{ij} für die Belieferung aller Kunden in b_{ij} festzustellen. Hier kann zunächst auf die spieltheoretische Diskussion über wahrheitsgemäßes Bieten verzichtet werden, da die berechneten Lieferkosten direkt als Gebotspreise übernommen werden und durch Verwendung identischer *ComEx*-Clients bei allen Profitcentern nicht manipulierbar sind [Schw05].⁶ Sei B_i eine Zusammensetzung aller Gebote vom Profitcenter i , die mit Hilfe logischer Operationen *OR* bzw. *XOR* verknüpft sind. Verschiedene *AtomicBids* können dieselben kombinierten Cluster C_i^y enthalten, z.B. *AtomicBids* b_{21} und b_{22} in Abb. 6. In diesem Fall werden die beiden Gebote mit *XOR* verknüpft: $B_2 = b_{21} \oplus b_{22}$. Sind die Teilmengen an den in *AtomicBids* vorkommenden Clustern disjunkt, so wird *OR*-Operation angewendet, z.B. $B_3 = b_{31} \vee b_{32} \oplus b_{33}$.

⁶ Da *DynaRoute* ein stochastisches Verfahren ist, stellen die Lieferkosten eine Verteilung dar, so dass Anreizkompatibilität nur näherungsweise gewährleistet ist und eine weitere Diskussion dieses Punktes erfolgen sollte.

3.2.3 Benutzeroberfläche

Für die Kundenverwaltung der Profitcenter und die Ergebnisdarstellung wurde eine grafische Benutzeroberfläche mit Hilfe der Java Swing Bibliothek in J2SE 5.0 erstellt (Abb. 7).

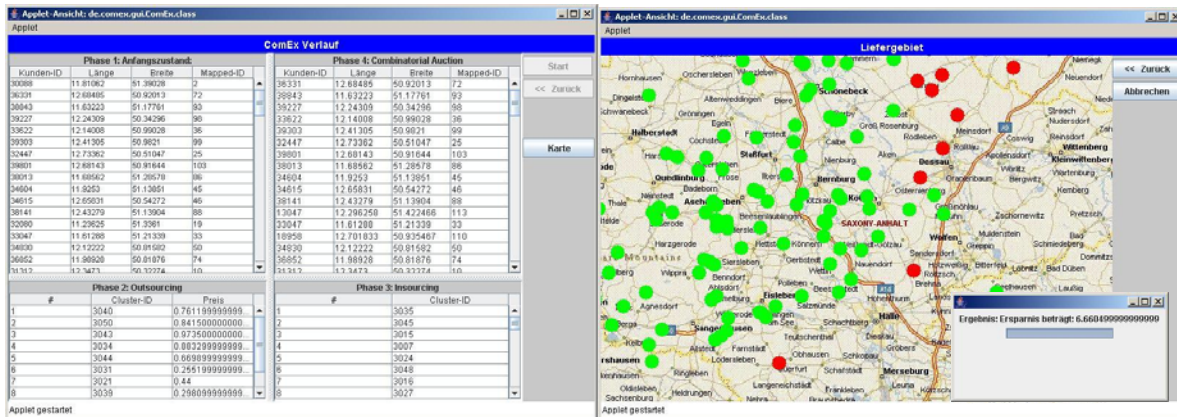


Abb. 7: Grafische Benutzeroberfläche von ComEx

Das ComEx-Client Programm, das in jedem Profitcenter installiert wird, enthält eine Kundentabelle mit den notwendigen Informationen, z.B. geografische Position eines Kunden, Kunden-ID usw. Kunden, die auf jeden Fall vom Profitcenter behalten werden sollen (bspw. auf Grund hoher Umsätze), können ausgewählt und somit dem Fixbereich zugeordnet werden. Nach dem Initiieren der Routenoptimierung werden die Ergebnisse der einzelnen Optimierungsschritte (Auslagerung freigegebener Kunden, die errechneten Gebotspreise, von anderen Profitcentern übernommene Kunden) nacheinander vom System angezeigt. Nach der Optimierung wird gezeigt, ob und wie viel durch den Einsatz der KA an Lieferkosten im Unternehmen gegenüber dem alleinigen Einsatz von DynaRoute eingespart werden konnte. Zur Veranschaulichung wird das Liefergebiet mit den zu beliefernden Kunden, die sowohl vom eigenen Profitcenter als auch von anderen Profitcentern übernommen wurden, auf einer Karte eingeblendet.

3.3 ComEx-Server

Die Aufgabe des ComEx-Servers ist es, den Ablauf der Auktion zu steuern. Die von den Profitcentern erstellten Anfragen werden von einem Servlet im ComEx-Webserver angenommen.⁷ Zwischen den Clients und dem ComEx-Server findet ein Informationsaustausch in Form von SOAP-Nachrichten statt. Unser System erlaubt die Übertragung von Nachrichten in der XML-basierten Sprache CAMEL, die für eine standardisierte Gebotsabgabe in KAs entworfen wurde [ScWS04]. Die Gebote aller Profitcenter werden im CAMEL-Format in einer Datenstruktur

⁷Als Webcontainer-Umgebung wurde die Apache Tomcat 5.5 Software gewählt: <http://tomcat.apache.org>

$B^* = B_1 \vee B_2 \vee \dots \vee B_I$ abgespeichert (Abb. 6). Alle SOAP-Nachrichten sind bei der Web-Übertragung per SSL-Technologie verschlüsselt. Nachdem die Gebote aller Profitcenter in der Datenstruktur B^* gespeichert worden sind, werden sie an die *ComEx*-Engine weitergeleitet, und die KA wird initiiert. Die optimale Allokation wird mittels der *ComEx*-Engine ermittelt und über den *ComEx*-Server an alle Clients übermittelt. Des Weiteren ist der *ComEx*-Server für die Berechnung der durch die Auktion im Gesamtsystem erzielten Ersparnisse zuständig.

3.4 *ComEx*-Engine

Wie im vorherigen Abschnitt beschrieben, werden die vom *ComEx*-Server gesammelten Gebote B_i von allen *ComEx*-Clients i in Form von *CAMEL*-Nachrichten an die *ComEx*-Engine weitergeleitet. Diese muss, um die optimale Allokation zu finden, das Combinatorial Auction Problem (CAP) [dVVo01] lösen. Seien z die Lieferkosten im Unternehmen, M die Menge aller kombinierten Cluster im System, J_i die Anzahl der Gebote vom Profitcenter i und $b_{ij}(C_i^y) = 1$ wenn das AtomicBid b_{ij} das Cluster C_i^y enthält, dann lautet das CAP für die Tourenoptimierung:

$$\min z = \sum_{i=1}^I \sum_{j=1}^{J_i} p_{ij} x_{ij} \quad (1)$$

u.d.N.
$$\sum_{i=1}^I \sum_{j=1}^{J_i} b_{ij}(C_i^y) \cdot x_{ij} = 1 \quad \forall C_i^y \in C^* \wedge x_{ij} \in \{0,1\} \quad (2)$$

$$b_{ij}(C_i^y) = \begin{cases} 1, & \text{wenn } C_i^y \text{ in } b_{ij} \text{ vorkommt} \\ 0, & \text{sonst} \end{cases} \quad \forall C_i^y \in C^* \quad (3)$$

Die Akzeptanzvariable x_{ij} zeigt an, ob ein Gebot den Zuschlag erhalten soll. Gleichzeitig definiert sie die hier notwendige Ganzzahligkeitsbedingung. Mit Gleichung (2) wird sichergestellt, dass alle kombinierten Cluster in der Lösungsallokation vorkommen. Für die Berechnung des CAP wird das Software-Paket *LP_SOLVE* 5.5⁸ verwendet. Das Paket liefert in akzeptabler Zeit eine exakte Lösung des hier beschriebenen *NP*-harten Problems bei bis zu einigen Hundert Geboten. Für Problemstellungen mit höherer Komplexität ist ein Umstieg auf Heuristiken sinnvoll. In [ScGu05] wurden daher heuristische Verfahren für das CAP [ScSR03] auf ihre Effizienz (Qualität der Lösung) und Schnelligkeit im Vergleich zu exakten Verfahren getestet.

⁸ <http://www.geocities.com/lpsolve>

4 Der Einfluss des Clustering-Prozesses auf die Gesamtlieferkosten

4.1 Ziel der Simulation und Simulationseinstellungen

Ziel des Einsatzes von *ComEx* ist die Reduktion der Gesamtlieferkosten im Unternehmen, die durch die Verkürzung der Gesamtlieferstrecke erzielt wird. Dieses Ziel kann nur erreicht werden, wenn eine geeignete Strategie zur Erstellung von Geboten für die Übernahme und Abgabe der Streckenabschnitte durch die Profitcenter angewendet wird. Im Rahmen unseres Systems kann dieses durch zwei Faktoren gesteuert werden: zum einen durch die Begrenzung des maximal zulässigen Abstandes r zwischen den Kunden innerhalb eines Clusters, zum zweiten durch die Festlegung des prozentualen Anteils a der Kunden, die dem Fixbereich P_f eines Profitcenters zugeordnet werden. Neben dem Einfluss dieser Faktoren wurde erforscht, wie sich die Änderung der Maximalzahl der zulässigen Gebote pro Profitcenter auf die Gesamtlieferkosten auswirkt.

Es wurde folgendes Testszenario gewählt: Die Simulationen basieren bezüglich geografischer Lage und Lieferzeiten auf Realdaten unseres Partnerunternehmens aus der Nahrungsmittellindustrie, wobei 3 Profitcenter mit je ca. 100 Kunden für die Auswertung zu Grunde gelegt wurden. Das CAP wurde mittels der *ComEx*-Engine basierend auf LP_SOLVE exakt gelöst. Der *DynaRoute*-Server lieferte die Routen und Kostenschätzungen basierend auf der oben erwähnten *COSA*-Heuristik⁹.

Für die Simulation wurden die relevanten Parameter innerhalb folgender Schranken variiert:

- Maximalabstand r zwischen den Kunden eines Clusters: 2–20 km (2 km Schritte)
- Anteil a der Kunden im Outsourcingbereich: 20% - 100% (20% Schritte)

Die Berechnung wurde mehrfach wiederholt, um die stochastischen Effekte zu reduzieren, die durch den Einsatz des heuristischen Verfahrens beim *DynaRoute*-Server entstanden.

4.2 Ergebnisse

Wie Abb. 8 zeigt, lässt sich bei einem Maximalabstand von 10 km zwischen den Kunden eines Clusters und bei einem Anteil der Kunden im Outsourcingbereich von 40% die Gesamtlieferstrecke am stärksten verkürzen¹⁰. Ein zu gering angesetzter Maximalabstand bei der Gebotsbil-

⁹ Es wurden folgende Einstellungen für das Cooperative Simulated Annealing Verfahren verwendet: Starttemperatur 1.000.000, 1% Abkühlrate und eine Populationsgröße von 20 Elementen.

¹⁰ Hier wird zur Vereinfachung die Lieferstrecke als Schätzer für die Lieferkosten verwendet.

ung führt zur Erstellung von Clustern, die aus nur wenigen Kunden bestehen. Dies bedeutet, dass die Abgabe bzw. Übernahme von solchen Clustern keine Synergieeffekte bewirken kann.

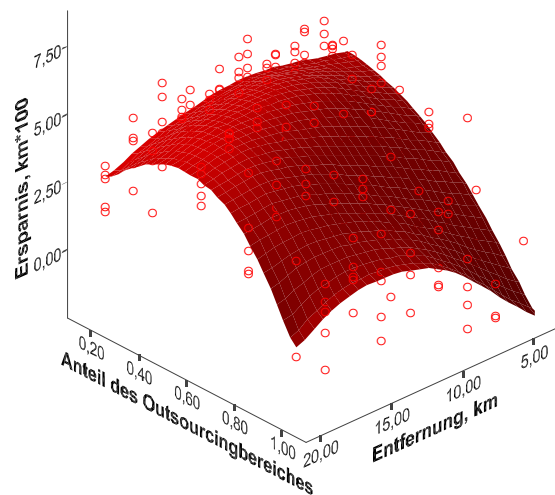


Abb. 8: Lieferkosteneinsparung durch den Einsatz von *ComEx*

Wird andererseits ein zu hoch angesetzter Maximalabstand r bei der Clustererstellung erlaubt, werden sehr komplexe Teilrouten für das Outsourcing erstellt. Solche größeren Cluster lassen sich schwer in die Touren anderer Profitcenter integrieren.

Ein zu großer Anteil von Kunden im Outsourcingbereich bei der Gebotsbildung führt zur Auslagerung vieler Kunden, die nahe zum eigenen Profitcenter liegen und eigentlich zu günstigen Konditionen von diesem beliefert werden können. Ein zu kleiner Anteil von Kunden eines Profitcenters im Outsourcingbereich verhindert die Freigabe zur Auslagerung geeigneter Kunden. In einer weiteren Simulation wurde beispielhaft untersucht, ob sich die zuvor identifizierten minimalen Lieferkosten (bei $r=10$ km, $a=40\%$) durch eine Änderung der maximal erlaubten Anzahl von Geboten weiter reduzieren lassen. Hierfür zeigt Abb. 9 die Gesamtlieferstrecke in Abhängigkeit zur maximal erlaubten Anzahl von Geboten pro Profitcenter:

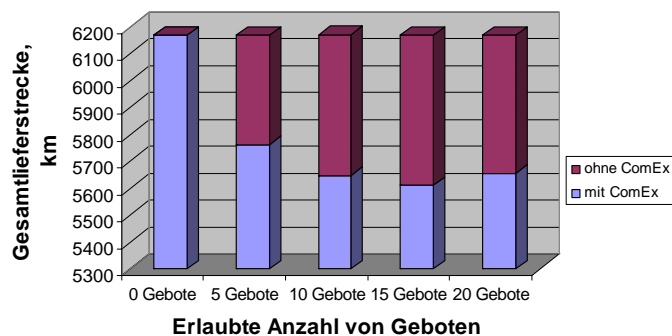


Abb. 9: Gesamtlieferstrecke in Abhängigkeit zu erlaubter Anzahl von Geboten

Die Erstellung einer höheren Anzahl von Geboten pro Profitcenter und somit die Erhöhung des Rechenaufwandes, der bei der *ComEx*-Engine anfällt, hebt das Einsparpotential im Unterneh-

men bis zur maximal zulässigen Anzahl von fünfzehn Geboten pro Profitcenter. Eine weitere Erhöhung hat jedoch einen negativen Effekt auf die Einsparungen.

5 Fazit

Das vorgestellte Softwaresystem *ComEx* wurde zum auktionenbasierten innerbetrieblichen Austausch von Transportdienstleistungen in Verbindung mit der Tourenoptimierungssoftware *DynaRoute* eingesetzt. Der Fokus der vorgestellten Untersuchungen lag auf der Ermittlung einer optimalen Strategie zur Erstellung von Geboten für die Übernahme bzw. Abgabe von Streckenabschnitten zwischen benachbarten Profitcentern, so dass für das gesamte Unternehmen Kostenreduktion entsteht. Anhand realer Lieferdaten eines Unternehmens aus der Nahrungsmittelindustrie wurden ein optimaler Maximalabstand von 10 km zwischen den Kunden und ein optimaler Anteil der Kunden im Fixbereich von 40% bei der Bildung optimaler In- bzw. Outsourcing-Gebote für die Berechnung der optimalen Allokation mit Hilfe der KA ermittelt. Eine höhere Bedeutung als im innerbetrieblichen Einsatz ist der Verwendung des *ComEx*-Verfahrens im zwischenbetrieblichen Bereich beizumessen, z.B. als allgemeine kombinatorische Transportbörse. Eine solche Realisierung eines Logistikmarktplatzes zwischen eigenständigen Unternehmen wirft jedoch in erhöhtem Maße die Frage nach anreizkompatiblen Gebotsbewertungsmechanismen, wie bspw. dem *Vickrey-Clarke-Groves*-Mechanismus, auf. Diese Problematik wurde im Zusammenhang mit dem hier diskutierten Anwendungsfall durch den Einsatz gleichartiger automatisierter Gebotsmechanismen bei allen Teilnehmern umgangen. Offen bleibt jedoch auch hier die Frage nach der gerechten und anreizkompatiblen Ersparnisverteilung zwischen Profitcentern. Diese wird zusammen mit dem Einsatz des *ComEx*-Systems als Transportbörse bei einem größeren Logistikdienstleister den nächsten Forschungsgegenstand darstellen.

Literaturverzeichnis

- [BuHS99] *Bullnheimer, B.; Hartl, R. F.; Strauss, C.:* An improved Ant System Algorithm for the Vehicle Routing Problem. In: *Annals of Operations Research* 89 (1999), S. 319-328.

- [CaPe05] *Cantillon, Estelle; Pesendorfer, Martin: Auctioning Bus Routes: The London Experience.* In: *Steinberg, R.; Shoham, Y.; Cramton, P. (Hrsg): Combinatorial Auctions.* MIT-Press: Cambridge, Massachusetts 2005. S. 573-592.
- [Capl96] *Caplice, Christopher: An Optimization Based Bidding Process: A New Framework for Shipper-Carrier Relationships.* Dissertation, Massachusetts Institute of Technology 1996.
- [CaSh03] *Caplice, Christopher; Sheffi, Yossi: Optimization-Based Procurement for Transportation Services.* In: *Journal of Business Logistics* 24 (2003) 2, S. 109-128.
- [dVVo01] *de Vries, Sven; Vohra, Rakesh: Combinatorial Auctions: A Survey.* In: *INFORMS Journal on Computing* 15 (2001) 3, S. 284-309.
- [ElKe05] *Elmaghraby, Wedad; Keskinocak, Pinar: Combinatorial Auctions in Procurement.* In: *Harrison, T. P.; Lee, H. L.; Neale, J. J. (Hrsg): The Practice of Supply Chain Management: Where Theory and Application Converge.* Springer, Berlin 2003, S. 245-258.
- [Glov86] *Glover, Fred: Future Paths for Integer Programming and Links to Artificial Intelligence.* In: *Computers and Operations Research* 13 (1986) 5, S. 533-549.
- [KaNa01] *Kameshwaran, S; Narahari, Y.: Auction Algorithms for Achieving Efficiencies in Logistics Marketplaces.* In: *Proceedings of the International Conference on Energy, Automation and Information Technology, 2001.*
- [LiLi02] *Li, Haibing; Lim, Andrew: Local search with annealing-like restarts to solve the vehicle routing problem with time windows.* ACM Press: Madrid, Spain, S. 560-565, 2002.
- [McMi95] *McMillan, John: Why Auction the Spectrum?* In: *Telecommunications Policy* 19 (1995), S. 191-199.
- [Nisa99] *Nisan, Noam: Bidding and Allocation in Combinatorial Auctions.* In: *Proceedings of the 2nd ACM Conference on Electronic Commerce (ACM-EC99), 1999.*

- [Pank03] *Pankratz, Giselher*: Zweiseitige kombinatorische Auktionen in elektronischen Transportmärkten – Potenziale und Probleme. Working Paper: FernUniversität Hagen, Fachbereich Wirtschaftswissenschaft: ftp://ftp.fernuni-hagen.de/pub/fachb/wiwi/win/forschung/publi/gp_p6.pdf, 2003, Abruf am: 30.6.2006.
- [ReSo03] *Regan, Amelia; Song, Jiongjiong*: An Auction Based Collaborative Carrier Network. Unter Begutachtung in: Transportation Research Part E: Logistics and Transportation Review 2003.
- [Schw05] *Schwind, Michael*: Design of Combinatorial Auctions for Allocation and Procurement Processes. In: Proceedings of the 7th International IEEE Conference on E-Commerce Technology 2005, München 2005, S. 391-395.
- [ScGu05] *Schwind, Michael; Gujo, Oleg*: An Agent-based Simulation Environment for a Combinatorial Grid Scheduler. Working Paper: Institute of Information Systems, Johann Wolfgang Goethe University: Frankfurt, Germany, <http://www.wiiw.de/publikationen/AnAgentBasedSimulationEnviron1531.pdf>, 2005, Abruf am: 30.6.2006.
- [ScSR03] *Schwind, Michael; Stockheim, Tim; Rothlauf, Franz*: Optimization Heuristics for the Combinatorial Auction Problem. In: Proceedings of the Congress on Evolutionary Computation CEC 2003, Canberra 2003, S. 1588-1595.
- [ScWS04] *Schwind, Michael; Weiss, Kilian; Stockheim, Tim*: CAMEL - Eine Meta-Sprache für Kombinatorische Auktionen. Working Paper: Institut für Wirtschaftsinformatik, Johann Wolfgang Goethe Universität: Frankfurt, <http://www.wiiw.de/publikationen/CAMELEineMetaSprachefuerKombin1068.pdf>, 2004, Abruf am: 30.6.2006.
- [Wend95] *Wendt, Oliver*: Tourenplanung durch Einsatz naturanaloger Verfahren. Gabler Verlag, Wiesbaden 1995.
- [WeSW05] *Wendt, Oliver; Stockheim, Tim; Weiss, Kilian*: Intelligente Tourenplanung mit DynaRoute. In: WIRTSCHAFTSINFORMATIK 47 (2005) 2, S. 135-140.