

Logistische Lösungen von Transportproblemen durch ein Werkzeug der Künstlichen Intelligenz

VON MOUNIR AL-DAAS, DRESDEN

b. fn. a
b. d. s. c
5

1. Einleitung

Wirtschaftswachstum und die Zunahme des Handels fordern ein modernes und leistungsfähiges Transportwesen unter strenger Beachtung des Umweltschutzes. Das wachsende Güteraufkommen verursacht vielfältige Transportprobleme (Stau, Verkehrsbehinderung), weitere Belastungen für die Umwelt (Schadstoffe und Lärm) sowie Gefahrenquellen für Menschen. Deshalb ist es wichtig, aber auch notwendig, die Transporte unter Berücksichtigung moderner Anforderungen an logistische Leistungen zu organisieren.

Bevor die logistische Lösung der Transportprobleme erläutert wird, soll eine kurze Darstellung der klassischen Algorithmen des Operations Research erfolgen. Zur konkreten Lösung dieser Probleme enthält der Text Hinweise auf ausgewählte Literatur.

2. Analyse logistischer Transportprobleme

Die zu erfüllende Aufgabe beim logistischen Transportproblem besteht darin, daß die Vorgaben der von den Kunden bestellten Güter hinsichtlich Lieferservice (Lieferzeit, Liefertreue, ...) bei minimalen Kosten eingehalten werden. Es kommt hinzu, daß die erschöpften Kapazitäten des Straßenverkehrs und die durch den Verkehr verursachten Umweltbelastungen sowie die Verkehrssicherheit berücksichtigt werden müssen. In diesem Sinne soll die Disposition der Kundenaufträge erfolgen.

Um diese ökonomischen, ökologischen und logistischen Anforderungen zu realisieren, muß eine Zielfunktion formuliert werden, die mindestens folgende Terme enthält:

- minimale Kosten,
- minimale Tourenanzahl,
- maximale Auslastung der Fahrzeuge und
- maximale Leistung zur Erfüllung der Kundenaufträge.

Anschrift des Verfassers:
Dipl.-Ing. Mounir AL-Daas
Institut für Transportlogistik
Technische Universität Dresden
Mommensenstraße 13
01062 Dresden

Diese Ziele zeigen, daß die logistischen Transportprobleme die folgenden Probleme enthalten:

1. das klassische Transportproblem,
2. das Standardproblem,
3. das Rucksackproblem,
4. das Tourenplanungsproblem und
5. das Rundreiseproblem.

Operations Research bietet für diese Anforderungen folgende Algorithmen an:

1. Das klassische Transportproblem

Ein häufig auftretendes Modell in der linearen Optimierung ist das klassische Transportproblem. Die zu erfüllende Aufgabe bei diesem Problem besteht darin, daß aus einer gegebenen Anzahl (r) von Lagern (A_i Quellen) mit Vorräten $a_i \geq 0$, $i = 1(1)r$ eine austauschbare Ware an eine bestimmte Anzahl (s) von Verbrauchern (B_j Senken) mit dem jeweiligen Bedarf $b_j \geq 0$, $j(1)s$ mit den dafür minimalen Gesamtaufwand transportiert werden soll.

Beim klassischen Transportproblem werden die Lieferbeziehungen zwischen den Quellen und Senken bestimmt. Hierbei wird zugeordnet, welche Senke wieviel Gutmenge aus welcher Quelle bekommt. Die ungarische Methode ist z.B. ein exaktes Verfahren zur Lösung dieses Problems (vgl. /1/ S. 130 ff.).

2. Standortproblem

Die Auswahl des richtigen Standortes vom Transit-Terminal (Lager) spielt eine bedeutende Rolle bei der Abwicklung der Kundenaufträge und beeinflusst das ganze Distributionssystem. Für die Errichtung des Transit-Terminals an der Peripherie der Stadt kommen nur bestimmte und begrenzte Orte in Frage. Die folgenden Standortfaktoren bilden die Basis für diese Auswahl:

– Grund und Boden

Die Lage, Größe, Bebauungsvorschriften und Expandierungsmöglichkeiten des vorgeschlagenen Geländes sowie die Erschließung des Gebietes beeinflussen maßgeblich die Auswahl. (Hier wird die Verzahnung zwischen Aufgaben der Stadt- und Verkehrsplaner deutlich.)

– Verkehrsverbindungen

Die durch Transportaktivitäten verursachten Kosten, Staus und Umweltbelastungen müssen minimiert werden. Das Straßennetz bestimmt die Beziehung des Terminals nach innen (zur Stadt), während die Autobahnen, Schienen und gegebenenfalls Flug-, See- und Binnenhafen die Beziehungen nach außen bestimmen.

– Die öffentliche Hand

Die öffentliche Hand kann z.B. durch Subventionen oder Steuervergünstigungen die Transit-Terminal-Betreiber fördern.

– Umweltbedingungen

Die Beachtung ökologischer Gesichtspunkte ist nicht nur gesetzliche Forderung. Durch eine Verknüpfung der stadtplanerischen Aufgaben mit den Belangen des Umweltschutzes ist ein positives Image der Betreiberfirma zu erreichen.

3. Rucksackproblem (Knapsack Problem)

Durch die Wahl verschiedener Objekte soll ein Rucksack so ausgefüllt werden, daß dessen Kapazität nicht überschritten wird und die Bequemlichkeit beim Tragen maximal ist.

In der Literatur wurde das Subset-sum Problem (Problem von Summen der Teilmengen) als praktischer Fall des Rucksackproblems erwähnt und als volumenabhängiges Rucksackproblem (Value Independent Knapsack Problem) genannt (vgl. /2/ S. 105 ff.).

Zur Ausfüllung des Rucksackes ist eine Teilmenge von den Objekten so auszuwählen, daß die Summe der Gewichte aller gewählten Objekte sehr nahe zur Rucksackkapazität liegen muß. Die Kapazität bzw. Tragfähigkeit des Rucksackes (Container oder Fahrzeug) darf nicht überschritten werden. Die Rucksackprobleme können entweder heuristisch oder exakt gelöst werden.

Um die Lösungsverfahren des Rucksackproblems auf die Disposition der Kundenaufträge anwenden zu können, müssen diese modifiziert werden. Zur Disposition der Kundenaufträge werden Fahrzeuge gebraucht, die unterschiedliche Kapazitäten und Konstruktionen besitzen. Darüber hinaus sind die Objekte (hier die zu transportierenden Güter) nicht immer untereinander verträglich. Infolgedessen müssen noch weitere Nebenbedingungen hinzugefügt werden:

- Verträglichkeit der mit einem Fahrzeug zu transportierenden Güter,
- Eignung des Fahrzeuges für das jeweilige Gut und
- Berücksichtigung der verschiedenen Ladekapazitäten der Fahrzeuge.

4. Tourenplanungsproblem

Die Kunden einer Region sollen von einem Auslieferungslager (Depot) aus mit den Gütern versorgt werden. Dafür werden verschiedene Fahrzeuge eingesetzt. Die Kunden bestellen die verschiedenen Güter in unterschiedlichen Mengen und zu unterschiedlicher Zeit.

Die Aufgabe der Tourenplanung besteht nun darin, die Kundenaufträge unter Berücksichtigung sämtlicher Nebenbedingungen auf die Touren (Fahrten) zu verteilen. Dabei sollen alle Kunden bedient und die Fahrstrecke minimiert werden (vgl. /3/ S. 131 ff.).

Eine andere Optimierungsrichtung ist die Minimierung der gesamten Transportkosten, die durch den Transportmitteleinsatz und die zurückgelegten Strecken entstehen (vgl. /4/ S. 10-11). Im Interesse der City-Logistik ist die Anzahl der Touren (Fahrten) zu minimieren. Mit anderen Worten besteht die Aufgabe der Tourenplanung schlechtweg darin, die Kundenaufträge den Fahrten zuzuordnen. Damit ist die Frage, welcher Kunde in welcher Tour bedient wird, geklärt.

Nachdem die Kunden auf die Touren zugeordnet sind (welcher Kunde wird bei welcher Tour bedient), ist die Reihenfolge der Bedienung der Kunden bei jeder Tour zu ermitteln, mit dem Ziel, die minimale Wegstrecke für jede Tour zu suchen. Jede Rundreise muß alle Kunden einer Tour beinhalten.

Die Tourenbildung (Zuordnung der Kunden zu den Fahrten) und die Routenbildung (Bedienungsreihenfolge der Kunden in einer Tour) sind von den Verbindungen (Straßennetz) zwischen den Kunden abhängig.

Bei N Kunden ergibt sich im allgemeinen die Anzahl der Verbindungen V wie folgt:

$$V = 1 + 2 + \dots + (N-2) + (N-1)$$

$$= \frac{N \cdot (N-1)}{2}$$

Die Anzahl der möglichen Kombinationen für die Tourenbildung K wächst exponentiell zur Anzahl der Verbindungen V

$$K = 2^V$$

Beispielsweise gibt es bei zehn Kunden ($N = 10$) 45 Verbindungen ($V = 45$) und $35 \cdot 10^{12}$ mögliche Kombinationen ($K = 35 \cdot 10^{12}$).

Die enorm große Anzahl der möglichen Kombinationen – sogar bei kleineren Kundenanzahl – macht das Verfahren der vollständigen Enumeration für die Suche nach einer optimalen Lösung unanwendbar, da die dafür notwendige Rechenzeit exponentiell mit der Anzahl der Kunden wächst. Zur Lösung des Tourenplanungsproblems wurden verschiedene heuristische Verfahren wie z.B. das Savingsverfahren nach *Clarck/Wright* /9/ und das Sweep-Verfahren nach *Gillett/Miller* /10/ vorgeschlagen.

5. Rundfahrtproblem (Travelling Salesman Problem)

Das Rundfahrtproblem bzw. Rundreiseproblem ist in der Literatur des Operations Research unter dem Begriff „Travelling Salesman Problem“ bekannt. Der Name kommt daher, daß ein Handelsreisender in einer Rundreise mehrere Orte jeweils nur einmal zu besuchen hat und anschließend zum Ausgangsort zurückkehren muß. Der dabei zurückzulegende Weg soll minimal sein. Bei der Lösung des Rundfahrtproblems können die kürzeste Gesamtentfernung, die kürzeste Gesamtreisezeit oder auch die minimalen Transportkosten angestrebt werden.

In der Literatur des Operations Research wurden diese Probleme im einzelnen und mitunter teilweise kombiniert behandelt. *Engle* /5/ versucht z.B. ein heuristisches Lösungsverfahren für simultane Standort- und Tourenplanung zu entwickeln.

Die Gesamtheit dieser Probleme stellt die Komplexität der logistischen Transportprobleme dar. Die Planung der logistischen Transportprobleme ist außerordentlich komplex und vielseitig. In der Mathematik wird diese Komplexität als NP-harte (nicht deterministisch polynomial harte) Probleme bezeichnet.

Die Lösung der logistischen Transportprobleme soll unter Berücksichtigung weiterer Restriktionen gesucht werden (vgl. /6/ S. 305-310):

1. Güterverträglichkeit
2. Güterpriorität
3. Die begrenzte Anzahl und Ladekapazität der Fahrzeuge
4. Verfügbarkeit der Fahrzeuge
5. Eignung des Fahrzeuges für den Transport der Güter
6. Erreichbarkeit und Zugänglichkeit der Kunden
7. Straßensperrung
8. Zeitfenster der Kunden
9. Maximale Dauer einer Tour

Powell et al. (/7/ S. 69-85) zeigten ein heuristisches Verfahren für die Optimierung der Tourenplanung mit Berücksichtigung des Kundenzeitfensters. Das mathematische Originalmodell in diesem Verfahren setzt jedoch voraus, daß alle Güter untereinander verträglich sind und daß jeder Kunde nur von einem Fahrzeug beliefert werden darf.

Die Einhaltung dieser Nebenbedingungen zeigt, daß das logistische Transportproblem mehr als eine Kostenoptimierung ist. Im Gegenteil zum klassischen Transportproblem sind die Transportkosten für bestimmte Zwecke (Eilbedürftigkeit, Notfälle, ...) nicht das erste Kriterium, sondern das Sicherstellen eines Lieferzeitpunktes bekannt als: „Koste es, was es wolle“. Die Dynamik der Logistikkosten in Abhängigkeit von der Qualität des Lieferservices soll untersucht werden.

3. Anwendung eines Werkzeuges der Künstlichen Intelligenz zur Lösung logistischer Transportprobleme

Mit dem Umfang der oben genannten Teilprobleme und einzuhaltenden Nebenbedingungen erreichen die logistischen Transportprobleme eine so hohe (harte) Komplexität, daß keines der vorgeschlagenen Lösungsverfahren ohne Modifikation anwendbar ist.

Schematisch läßt sich das logistische Transportproblem wie folgt darstellen:

Gegeben seien:

- eine endliche Menge von Gütern, Kunden, Kundenaufträgen und Transportmitteln,
- die einer Reihe von Nebenbedingungen (Verträglichkeit, Lkw-Kapazität, Beladbarkeit, Erreichbarkeit, ...) unterworfen sind.

Gesucht wird eine Lösung,

- welche sämtliche Nebenbedingungen erfüllt und
- dabei noch die Zielfunktion (Anzahl der einzusetzenden Transportmittel) minimiert (optimiert).

Diese Aufgabenstellung scheint auf den ersten Blick einfach zu sein, verbirgt jedoch eine enorme kombinatorische Vielfalt der dabei auftretenden Situationen, welche ihre Lösung

außerordentlich schwierig gestaltet. Der große Umfang der einzuhaltenden Nebenbedingungen im logistischen Transportproblem zwingt dazu, daß die Suche nach der Lösung solcher Probleme in einer anderen Herangehensweise erfolgen sollte.

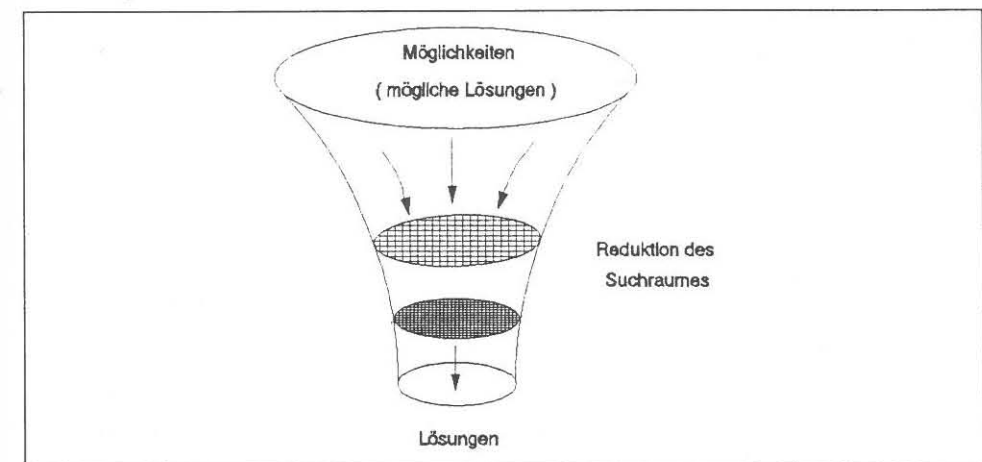
Anhand eines Werkzeuges der Künstlichen Intelligenz werden die Nebenbedingungen ausgenutzt, um das Untersuchungsgebiet einzuschränken.

3.1 Die KI-Software *Charme* – Einführung

Im Rahmen der Künstlichen Intelligenz wurde eine auf Zusatzbedingungen basierende Programmiersprache (constraints-based Programming) entwickelt. *Charme* ist eine constraints-basierte Programmiersprache, die 1990 von der BULL AG entwickelt wurde. In dieser Programmiersprache wird die im Operations Research bekannte Technik der *Branch-and-Bound Methode* angewandt. Dabei werden die Zusatzbedingungen (Constraints) dafür eingesetzt, um die „unfruchtbaren“ Zweige des Entscheidungsbaumes abzuschneiden und somit das Suchgebiet zu reduzieren. Die permanente Überprüfung auf Einhaltung der Nebenbedingungen wird durch einen Mechanismus gewährleistet, der in der Fachsprache als *constraint satisfaction* oder *constraint propagation* bezeichnet wird. Ehe eine der eingesetzten Variablen einen neuen Wert annehmen kann, wird überprüft, ob dieser mit den Nebenbedingungen verträglich ist. Wenn es bei der Suche nach der Lösung zu einer „Sackgasse“ kommt, benutzt *Charme* die Technik der Rückwärtsverfolgung (Backtracking). *Charme* kombiniert die Effizienz von Algorithmen des Operations Research mit der Anwendung von Methoden der künstlichen Intelligenz (vgl. /8/).

Das Durchmustern des Suchraumes beruht auf einem Reduktionsmechanismus: Mittels der Nebenbedingungen wird der anfängliche Suchraum immer weiter eingeeengt, indem aus den einzelnen Domänebereichen jene Variablenwerte, die bei der Lösung nicht auftreten können, sukzessive entfernt werden. Dieses Verfahren wird solange wiederholt, bis alle Variablen mit einem Wert belegt sind und diese eine Lösung darstellen (siehe Abb. 1).

Abbildung 1: Reduktion des Suchraumes



Die Suche erfolgt nicht-deterministisch. Die verschiedenen Instanziierungen der Variablen werden nacheinander ausprobiert. Zur Verdeutlichung soll das Verfahren mittels eines physikalischen Modells beschrieben werden:

Jede Nebenbedingung spielt die Rolle eines „Siebes“ mit einer bestimmten Feinheit. Die unterschiedlichen Nebenbedingungen stellen verschiedene Siebe mit verschiedenen Feinheiten dar. Die Reihung der Siebe in einem Siebungsprozeß beeinflusst maßgeblich den Durchsatz und die Arbeitszeit (siehe Abb. 1). Genauso stark beeinflusst die Reihenfolge der Nebenbedingungen die Rechenzeit. Aufgrund dessen muß die Einordnung der Nebenbedingungen sorgfältig gewählt werden.

Bei der Suche nach der „optimalen“ (minimalen) Lösung bietet *Charme* zwei Möglichkeiten:

1. Tiefensuche im Entscheidungsbaum (minimize Variante)

Bei dieser Variante, begonnen von der Wurzel, sucht *Charme* in einem (z.B. linken) Zweig (Teilbaum) nach der ersten Lösung. Wenn eine zulässige Lösung gefunden wird, wird der Wert der Zielfunktion als Obergrenze betrachtet. Durch Iterationen in demselben (aktuellen) Zweig wird nach einem besseren (kleineren) Wert für die Zielfunktion gesucht. Wenn ein kleinerer Wert gefunden wurde, wird dieser als neue Obergrenze betrachtet usw.. Wenn der aktuelle Zweig durchmustert ist, prüft *Charme* den nächsten Zweig usw. Kommt es dabei zu einer „Sackgasse“, wird durch Backtracking zum nächsten Verzweigungspunkt (Choice-point) in demselben Zweig zurückgegangen (eingeschränkte Form des Branch-and-Bound).

2. Breitensuche (minim Variante)

Hier wird eine erste Lösung in einem Zweig gesucht. Dann wird in die Wurzel zurückgegangen und in einem anderen Zweig nach einer besseren Lösung gesucht. Dabei wird durch das Backtracking die Blockanweisung bei jedem neuen Durchlauf komplett ausgeführt. Dies wird wiederholt, bis die gesuchte (optimale) Lösung gefunden wird.

Die Varianten *minim* und *minimize* liefern die gleichen Ergebnisse, aber unterscheiden sich in der Herangehensweise bei der Suche nach der minimalen Lösung. Da *minim* Variante manchmal zu Beginn der Suche ganze Zweige des Suchbaumes abschneidet, ist *minim* schneller als *minimize*. Jedoch bei komplexen Berechnungen innerhalb der Blockanweisung kann *minimize* die Lösungsfindung beschleunigen.

3.2 Der Aufbau des Algorithmus

Basierend auf dieser Software wird ein Programm zur Lösung der logistischen Transportprobleme entwickelt. Im folgenden wird der Algorithmus dieses Programmes analysiert:

1. In einem Feld (Matrix) sind die Kundenaufträge, die in dem Transit-Terminal eingetroffen sind, einzuordnen (siehe Tab. 1).

Normalerweise beinhalten die Kundenaufträge die folgenden Informationen: Kundennummer, das bestellte Gut, die bestellte Menge, Beginn und Ende des Zeitfensters, in dem die Lieferung beim Kunden ankommen soll. Anhand der bestellten Gutmenge wird die notwendige Zeit für den Umschlag ermittelt.

Tabelle 1: Einordnung der Kundenaufträge in einem Feld

Kunden-auftrag	Kunde	Gut	Menge	Umschlag-zeit	Zeitfenster-beginn	Zeitfenster-ende
1	3	4	2	8	540	660
2	7	1	3	15	600	720
3	3	1	1	6	540	660
.
.
N	k	g	m	t	b	e

Zur Normierung der Zeiteinheiten für die Umschlagzeit, Fahrzeit sowie den Beginn und das Ende des Zeitfensters wird in diesem Modell die Zeiteinheit [Minute] eingeführt. Der Tag beginnt 0.00 Uhr (0 Minute) und endet 24.00 Uhr (1440 Minute). Z.B. der Beginn des Zeitfensters um 10.00 Uhr bedeutet Minute 600, 13.00 Uhr heißt 780. Minute usw.

Ein Kunde kann mehrere Aufträge erteilen. Beispielsweise gehören in (Tab. 1) die Aufträge 1 und 3 demselben Kunden (3). Die symbolische Darstellung der Kundenaufträge ist wie folgt:

$$A_i = [k, g, m, u, b, e] ::$$

$$[Kunde, Gut, Menge, Umschlagzeit, Zeitfensterbeginn, -ende]$$

Dabei sind:

A_i der Kundenauftrag Nummer i ,

$$\begin{aligned} \forall i \in \{1, \dots, N\}; & \quad N = \text{Anzahl der Kundenaufträge,} \\ \forall k \in \{2, \dots, K+1\}; & \quad k = 1 \text{ ist das Lager; } K = \text{Anzahl der Kunden,} \\ \forall g \in \{1, \dots, G\}; & \quad G = \text{Anzahl der Güter,} \\ \forall m \in \{1, \dots, M\}; & \quad M = \text{Anzahl der Ladeeinheiten,} \\ \forall u \in \{1, \dots, U\}; & \quad U = \text{Zeiteinheiten für den Umschlag,} \\ \forall b \in \{1, \dots, B\}; & \quad B = \text{Beginn des Zeitfensters (Zeiteinheit),} \\ \forall e \in \{1, \dots, E\}; & \quad E = \text{Ende des Zeitfensters (Zeiteinheit).} \end{aligned}$$

2. In einem Feld (Matrix) sind die Entfernungen zwischen den Kunden untereinander und dem Transit-Terminal einzuordnen. Wird die Entfernungsmatrix durch die durchschnittliche Fahrgeschwindigkeit auf der jeweiligen Strecke dividiert, so ergibt sich die entsprechende Fahrzeitmatrix (siehe Tab. 2).

Tabelle 2: Die Fahrzeitmatrix

Kunden → ↓ Kunden	1 (Lager)	2	3	. k .	K+1
1	0	τ_{12}	τ_{13}	...	τ_{1K+1}
2	τ_{21}	0	τ_{23}	...	τ_{2K+1}
.
l	.	.	.	τ_{lk}	.
.
K+1	τ_{K+11}	τ_{K+12}	τ_{K+13}	τ_{K+1k}	0

Hier wird die Fahrzeit für alle im Fuhrpark befindlichen Fahrzeuge vereinheitlicht. Befinden sich im Fuhrpark unterschiedliche Fahrzeuge, so kann die Fahrzeit für das jeweilige Fahrzeug mit einem Faktor korrigiert werden. Symbolisch kann die Fahrzeitmatrix wie folgt dargestellt werden:

$$\text{Fahrzeit}_{kl} = [\tau_{kl}] :: [1..K+1, 1..K+1]$$

$\forall k, l \in \{1, \dots, K+1\}$; K = Anzahl der Kunden,
k = 1 = 1 steht für das Terminal bzw. das Lager.

τ_{kl} ist die Fahrzeit zwischen dem Kunden bzw. dem Terminal (k) und dem Kunden (l).

Die hier anzuwendenden Zeiteinheiten müssen mit den Zeiteinheiten der Umladezeit und dem Beginn und Ende des Zeitfensters kompatibel sein.

3. In einem Feld (Matrix) ist die Verträglichkeit der Güter untereinander zu bestimmen. Diese Matrix ist symmetrisch und enthält $G \cdot G$ binäre Variablen (siehe Tab. 3).

Tabelle 3: Verträglichkeitsmatrix

Güter → ↓ Güter	1	2	. h .	G
1	1	V_{12}	...	V_{1G}
2	V_{21}	1	...	V_{2G}
.
g	.	.	V_{gh}	.
.
G	V_{G1}	V_{G2}	...	1

Die symbolische Schreibweise der Verträglichkeitsmatrix ist wie folgt:

$$\text{Verträglichkeit}_{gh} = [V_{gh}] :: [1..G, 1..G]$$

$\forall g, h \in \{1, \dots, G\}$; G = Anzahl der Güter

V_{gh} ist die Verträglichkeit zwischen dem Gut g und dem Gut h.

$V_{gh} = \begin{cases} 1 & \text{Falls Gut g mit dem Gut h verträglich ist.} \\ 0 & \text{sonst.} \end{cases}$

Auf diese Matrix wird zurückgegriffen, um die Zusammenladeverbote der unverträglichen Güter auf ein und demselben Fahrzeug zu garantieren.

4. In einem Feld (Matrix) ist die Beladbarkeit der Güter auf den im Fuhrpark zur Verfügung stehenden Fahrzeugen zu bestimmen. Diese Matrix enthält $G \cdot F$ binäre Variablen. Symbolisch wird die Beladbarkeit wie folgt:

$$\text{Beladbarkeit}_{gf} = [B_{gf}] :: [1..G, 1..F]$$

$\forall g \in \{1, \dots, G\}$; G = Anzahl der Güter

$\forall f \in \{1, \dots, F\}$; F = Anzahl der Fahrzeuge

B_{gf} ist die Beladbarkeit zwischen dem Gut g und dem Fahrzeug f.

$B_{gf} = \begin{cases} 1 & \text{Falls das Gut i auf dem Fahrzeug f ladbar ist.} \\ 0 & \text{sonst.} \end{cases}$

Anhand dieser Matrix wird gewährleistet, daß das richtige Gut mit dem richtigen Fahrzeug transportiert wird. Das ist sehr wichtig, insbesondere für die gekühlten Waren.

5. Die großen Fahrzeuge können die in engen Straßen befindlichen Kunden nicht erreichen. Deshalb ist in einem Feld (Array) die Erreichbarkeit der Kunden durch die im Fuhrpark zur Verfügung stehenden Fahrzeuge zu bestimmen. Diese Matrix enthält $F \cdot K$ binäre Variablen. Die Erreichbarkeit läßt sich symbolisch wie folgt darstellen:

$$\text{Erreichbarkeit}_{fk} = [E_{fk}] :: [1..F, 1..K]$$

$\forall f \in \{1, \dots, F\}$; F = Anzahl der Fahrzeuge

$\forall k \in \{1, \dots, K\}$; K = Anzahl der Kunden

E_{fk} ist die Erreichbarkeit des Kunden k durch das Fahrzeug f.

$E_{fk} = \begin{cases} 1 & \text{Falls das Fahrzeug f den Kunden k erreicht.} \\ 0 & \text{sonst.} \end{cases}$

Eine andere Variante zur Gewährleistung der Beladbarkeit und Erreichbarkeit kann wie folgt dargestellt werden. Zuerst werden die Fahrzeuge des Fuhrparks je nach Typ bzw. Ladekapazität in Gruppen (Clusters) eingeteilt. Nach derselben Methode sind auch die Güter hinsichtlich der Beladbarkeit in Gruppen einzuteilen. Danach wird festgelegt,

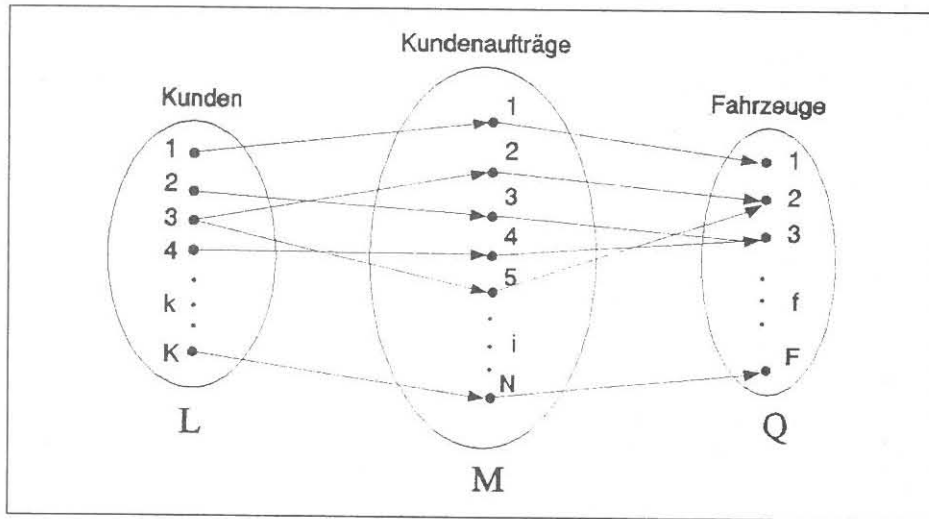
welche Gütergruppe mit welchem Fahrzeugtyp transportiert werden darf. Die Erreichbarkeit der Kunden durch Fahrzeugtypen kann genauso wie die Beladbarkeit ermittelt werden. Es soll zwischen der Beladbarkeit und der Erreichbarkeit keine Widersprüche geben.

6. Die im Fuhrpark zur Verfügung stehenden Fahrzeuge sind mitunter von der Kapazität her inhomogen. Infolgedessen empfiehlt sich eine Gruppierung (Clusterung) der Fahrzeuge nach der Kapazität, damit die Ladekapazität jeder Gruppe nicht überschritten wird. Dies führt zur Reduzierung der Daten.
7. Die Ladeeinheiten sind zu vereinheitlichen. Die Erfahrung zeigt, daß der Laderaum eines Verteilfahrzeuges eher eine Obergrenze für die Beladung des Fahrzeuges bildet als die Tragfähigkeit. Daher sollen die Ladeeinheiten nach Volumen und nicht nach Gewicht betrachtet werden.
8. Als Zielfunktion ist die Anzahl der Fahrzeuge unter Beachtung der vorgegebenen Nebenbedingungen zu minimieren. Dies ist ein reines Zuordnungsproblem: Welches Gut wird auf welches Fahrzeug geladen und welches Fahrzeug wird welchen Kunden beliefern oder welches Fahrzeug kann welche Kundenaufträge bedienen?

Eine Demonstration dieser Zuordnung ist anhand der Mengenlehre (Abb. 2) ersichtlich.

L ist eine Menge von Kunden $k = 1, \dots, K$. Jedes Element (Kunde) dieser Menge kann einen, keinen oder mehr als einen Auftrag erteilen, in dem er das bestellte Gut, die Menge und das Zeitfenster für die Belieferung angibt. Infolgedessen enthält jeder Kundenauftrag die folgenden Daten:

Abbildung 2: Erfassung und Disposition der Kundenaufträge



1. den Kunden, der diesen Auftrag erteilt,
2. das bestellte Gut,
3. die gewünschte Menge,
4. den Beginn des Zeitfensters,
5. das Ende des Zeitfensters und
6. aufgrund der bestellten Menge kann die benötigte Umschlagzeit ermittelt werden.

Es bildet sich dann die Menge M von Kundenaufträgen $i = 1, \dots, N$. Die Menge Q enthält die zur Verfügung stehenden Fahrzeuge $f = 1, \dots, F$. Die Aufgabe besteht nun darin, alle Kundenaufträge auf die Fahrzeuge so zu disponieren, daß die Anzahl der benötigten Fahrten minimal wird. Dabei sollen die folgenden Nebenbedingungen berücksichtigt werden:

- die begrenzte Kapazität der Fahrzeuge,
- die Verträglichkeit der Güter,
- die Beladbarkeit der Güter auf den Fahrzeugen,
- die Erreichbarkeit der Kunden durch die Fahrzeuge,
- das Zeitfenster bei den Kunden und
- die gesamte Tourendauer.

Für die Zuordnung der Kundenaufträge auf die Fahrzeuge wird ein Feld (Array) wie folgt definiert:

$[1..N]$, wobei N die Anzahl der Kundenaufträge ist. Jedes Element dieses Feldes (Kundenauftrag) wird als Variable betrachtet. Jede dieser Variablen soll einen einzigen Wert von dem Feld $[1..F]$ bekommen, wobei F die Anzahl der Fahrten (Fahrzeuge) ist, die damit initialisiert werden. Mehrere Variablen können jedoch den gleichen Wert bekommen. Das heißt, jeder Kundenauftrag (Variable) kann einem Fahrzeug (einer Fahrt) zugeordnet werden. Bei der Initialisierung der Variablen sollen alle vorgegebenen Restriktionen berücksichtigt werden. Wenn alle Variablen initialisiert sind, dann ist das Problem gelöst. Die Gruppe von Kundenaufträgen, die den gleichen Wert einnimmt, bildet eine Tour und kann mit einem entsprechenden Fahrzeug transportiert werden.

Hinweis

Die Kundenaufträge werden auf die Touren (Fahrten) disponiert. Hierbei wird angenommen, daß die Anzahl der Touren der Anzahl der Fahrzeuge entspricht. Dies bedeutet jedoch nicht, daß der Fuhrpark die gleiche Anzahl von Fahrzeugen wie die Anzahl der Touren besitzen muß. Die Anzahl der notwendigen Fahrzeuge hängt von den Nebenbedingungen ab. Beispielsweise kann ein Fahrzeug an einem Tag mehrere Touren durchführen, wenn die Nebenbedingungen dies erlauben, als ob dieses Fahrzeug mehrfach vorhanden wäre. Auf diesen Hinweis wird später nochmal eingegangen.

Vereinfachend läßt sich das Ergebnis der Zuordnung wie in der Tabelle 4 darstellen.

Tabelle 4: Zuordnung der Kundenaufträge auf die Touren

Kundenaufträge → ↓ Fahrzeuge	1	2	3	4	...	N-1	N
1	X		X				
2						X	X
⋮							
F		X		X			

Diese Tabelle zeigt die Zuordnung der Kundenaufträge auf die Fahrzeuge (Fahrten). Hier haben die Fahrten und Fahrzeuge die gleiche Nummer. Das Zeichen (X) bedeutet, daß der entsprechende Kundenauftrag in der jeweiligen Fahrt erfüllt wird, z.B. werden die Kundenaufträge 1 und 3 in der ersten Fahrt bedient. Zu bemerken ist, daß es in jeder Spalte nur ein einziges Zeichen (X) geben darf, d.h. jeder Kundenauftrag nur einmal bedient wird. Die bestellte Menge in jedem Kundenauftrag soll kleiner/gleich der Kapazität der vorhandenen Fahrzeuge sein. Falls die bestellte Menge größer als die Kapazität ist, dann soll der jeweilige Kundenauftrag wie zwei Aufträge betrachtet werden. Bei der Zuordnung sollen alle angegebenen Nebenbedingungen berücksichtigt werden. Das Ziel ist, die Anzahl der benötigten Touren (Fahrten) zu minimieren.

Zuerst wird $F \leq N$ (F = Anzahl der Fahrten, N = Anzahl der Kundenaufträge) vorgeschlagen. Wenn die Anzahl der vorgeschlagenen Fahrten nicht ausreicht, führt dies zu keiner Lösung, weil die Nebenbedingungen verletzt werden.

Zur Generierung der Lösung besitzt Charme verschiedene Strategien. Da das Ziel in diesem Problem ist, die Anzahl der Fahrten zu minimieren, versucht die Strategie von Charme allen Variablen im Feld $[1..N]$ ein und denselben Wert des Feldes (domain) $[1..F]$ zu geben. Das heißt, alle Kundenaufträge werden in einer Fahrt bedient. Aber hierbei werden die Nebenbedingungen verletzt. Durch das sogenannte Backtracking erzeugt Charme die nächst zulässige Lösung und prüft gleichzeitig die Nebenbedingungen. Werden alle Nebenbedingungen respektiert, so repräsentiert die gefundene Lösung die „optimale“ Lösung, ansonsten wird durch Backtracking eine neue zulässige Lösung generiert usw.

3.3 Die mathematische Formulierung

Die mathematische Formulierung dieses logistischen Problems kann wie folgt beschrieben werden:

Zuordnen :: [1..N] of [1..F]

dabei sind: N die Anzahl der Kundenaufträge und
 F die Anzahl der Fahrzeuge

Achtung!

Die Anzahl der Fahrzeuge entspricht hier der Anzahl der Touren (Fahrten). Das bedeutet aber nicht, daß der Fuhrpark die gleiche Anzahl von Fahrzeugen haben muß, da einige Fahrzeuge mehrere Touren durchführen können. Theoretisch entsprechen die Tourennummern den Fahrzeugnummern, aber praktisch kann ein Fahrzeug mehrere Touren nacheinander durchführen, als ob dieses Fahrzeug mehrfach vorhanden wäre.

Zielfunktion: Anzahl der Fahrzeuge $F \rightarrow \min!$

Nebenbedingungen:

1. Kapazität

$$\sum [menge_{if}] \leq c_f$$

$\forall i \in \{1, \dots, N\}$; N = Anzahl der Kundenaufträge

$\forall f \in \{1, \dots, F\}$; F = Anzahl der Fahrzeuge

Dabei sind:

$menge_{if}$ die bestellte Menge im i -ten Kundenauftrag, der dem f -ten Fahrzeug zugeordnet ist,
 c_f die Kapazität des f -ten Fahrzeuges.

2. Erreichbarkeit

$$E[kunde_{if}, f] = 1$$

$\forall i \in \{1, \dots, N\}$; N = Anzahl der Kundenaufträge

$\forall f \in \{1, \dots, F\}$; F = Anzahl der Fahrzeuge

Dabei sind:

$kunde_{if}$ der Kunde im i -ten Kundenauftrag, der dem f -ten Fahrzeug zugeordnet ist,
 f ist das f -te Fahrzeug und
 E Erreichbarkeit des Kunden, dessen Auftrag i ist, durch das f -te Fahrzeug. Deren Wert wird von der Erreichbarkeitsmatrix entnommen.

3. Beladbarkeit

$$B[gut_{if}, f] = 1$$

$\forall i \in \{1, \dots, N\}$; N = Anzahl der Kundenaufträge

$\forall f \in \{1, \dots, F\}$; F = Anzahl der Fahrzeuge

Dabei sind:

gut_{if} das bestellte Gut in dem i -ten Kundenauftrag, der dem f -ten Fahrzeug zugeordnet ist,
 f ist das f -te Fahrzeug und
 B die Beladbarkeit des im i -ten Kundenauftrag bestellten Gutes auf dem f -ten Fahrzeug. Der Wert von B wird von der Beladbarkeitsmatrix entnommen.

4. Verträglichkeit

$$V [gut_{if}, gut_{jf}] = 1$$

$\forall i, j \in \{1, \dots, N\}$; N = Anzahl der Kundenaufträge

$\forall f \in \{1, \dots, F\}$; F = Anzahl der Fahrzeuge

Dabei sind:

gut_{if} das bestellte Gut in dem i-ten Kundenauftrag, der dem f-ten Fahrzeug zugeordnet ist,

gut_{jf} das bestellte Gut in dem nächsten (j-ten) Kundenauftrag, der auch demselben (f-ten) Fahrzeug zugeordnet ist

V die Verträglichkeit der Güter untereinander. Der Wert von V ist aus der Verträglichkeitsmatrix erhaltbar.

5. Zeitfenster bei den Kunden

Hier werden die kompatiblen Kundenaufträge, die nacheinander und mit demselben Fahrzeug bedient werden, definiert. Jeder Kunde bestimmt in seinem Auftrag ein Zeitintervall (Zeitfenster), in dem die Belieferung erfolgen soll. Die Ankunftszeit beim ersten Kunden in der Tour hängt von der Abfahrtszeit des Fahrzeuges vom Terminal ab. Es wird vorausgesetzt, daß jedes Fahrzeug das Terminal frühzeitig verläßt, um den ersten Kunden in der Tour rechtzeitig zu erreichen. Die Rückfahrt des Fahrzeuges zum Terminal wird durch die Gesamtzeit bestimmt (siehe Punkt 6).

$$t_{if} \geq t_{if} + u_{if} + \tau_{(ij)f}$$

$$b_{if} \leq t_{if} \leq e_{if}$$

$$t_{if} \geq 0$$

$\forall i, j \in \{1, \dots, N\}$; N = Anzahl der Kundenaufträge

Dabei sind:

i, j die Kundenaufträge, die demselben (f-ten) Fahrzeug zugeordnet sind,

t_{if} die Ankunftszeit des f-ten Fahrzeuges beim Kunden, dessen Auftrag i ist,

t_{jf} die Ankunftszeit des f-ten Fahrzeuges beim nächsten Kunden, dessen Auftrag j ist,

$\tau_{(ij)f}$ die Fahrzeit des f-ten Fahrzeuges vom Kunden, dessen Auftrag i ist zum nächsten Kunden, dessen Auftrag j ist,

u_{if} die benötigte Umschlagzeit für den i-ten Kundenauftrag, der dem f-ten Fahrzeug zugeordnet ist,

b_{if} Zeitfensterbeginn des i-ten Kundenauftrages, der dem f-ten Fahrzeug zugeordnet ist,

e_{if} Zeitfensterende des i-ten Kundenauftrages, der dem f-ten Fahrzeug zugeordnet ist.

Um weitere mögliche Kombinationen, die keine zulässige Lösungen liefern, zu entfernen, soll eine weitere Nebenbedingung (die Kompatibilität) eingeführt werden. Alle nicht kompatiblen Kundenaufträge können nicht auf derselben Tour (derselben Fahrt) bedient werden.

Die Kompatibilität wird zwischen den Kundenaufträgen paarweise geprüft. Zwei Kundenaufträge (i, j) sind inkompatibel, wenn das Zeitfensterende des ersten Kundenauftrages (i) plus die Fahrzeit vom Kunden in diesem Kundenauftrag (i) zum Kunden im Kundenauftrag (j) größer als das Zeitfensterende des Kundenauftrages (j) ist und umgekehrt. Das heißt, der Kunde, dessen Auftrag (i) ist, und der Kunde, dessen Auftrag (j) ist, sind inkompatibel, wenn:

$$e_i + \tau_{ij} > e_j \text{ und}$$

$$e_j + \tau_{ji} > e_i$$

$\forall i, j \in \{1, \dots, N\}$; N = Anzahl der Kundenaufträge

Infolgedessen sollen die beiden Kundenaufträge i und j bei der Zuordnung verschiedene Werte bekommen.

Dabei sind:

e_i Zeitfensterende des i-ten Kundenauftrages,

e_j Zeitfensterende des j-ten Kundenauftrages,

τ_{ij} die Fahrzeit vom Kunden, dessen Auftrag i ist zum nächsten Kunden, dessen Auftrag j ist,

τ_{ji} die Fahrzeit vom Kunden, dessen Auftrag j ist zum nächsten Kunden, dessen Auftrag i ist.

6. Gesamtzeit (Tourendauer)

Die Gesamtzeit jeder Fahrzeugtour besteht aus der Fahrzeit vom Terminal zum ersten Kunden in der Tour, der Fahrzeit zwischen den einzelnen Kunden, der Umschlagzeit (Übergabezeit) der Güter bei jedem Kunden und der Fahrzeit für die Rückfahrt vom letzten Kunden zum Terminal (die notwendige Zeit für eine Tour). Die Gesamtzeit einer Tour muß den gesetzlichen Regeln der Arbeitszeit entsprechen.

$$G_f = \tau_{(1)f} + \sum u_{if} + \sum \tau_{(ij)f} + \tau_{(j1)f} \leq \Gamma_f$$

$\forall i, j \in \{1, \dots, N\}$; N = Anzahl der Kundenaufträge

$\forall f \in \{1, \dots, F\}$; F = Anzahl der Fahrzeuge

Dabei sind:

G_f die Gesamtzeit des f-ten Fahrzeuges,

$\tau_{(1)f}$ die Fahrzeit des f-ten Fahrzeuges vom Terminal (Lager) zum ersten Kunden, dessen Auftrag i ist,

u_{if} die benötigte Umschlagzeit für den i-ten Kundenauftrag, der dem f-ten Fahrzeug zugeordnet ist,

$\tau_{(ij)f}$ die Fahrzeit des f-ten Fahrzeuges vom Kunden, dessen Auftrag i ist zum nächsten Kunden, dessen Auftrag j ist,

$\tau_{(j1)f}$ die Fahrzeit des f-ten Fahrzeuges vom letzten Kunden, dessen Auftrag j ist zurück zum Terminal,

Γ_f Konstante, die Fahrzeug abhängig sein kann = die gesetzliche bzw. erlaubte Arbeitszeit (z.B. 8 h/d) des f-ten Fahrzeuges.

3.4 Demonstrationsbeispiel

Gegeben seien die folgenden Kundenaufträge, die disponiert werden müssen. Jeder Kundenauftrag enthält die folgenden Daten:

1. Kundennummer,
2. Gutnummer,
3. die bestellte Gutmenge,
4. die ermittelte Umschlagzeit,
5. Beginn des Zeitfensters für die Anlieferung und
6. Ende des Zeitfensters.

In diesem Beispiel werden 20 Kundenaufträge disponiert. Die Anzahl der Kunden ist 9 und die Anzahl der Güter ist 8. Beispielsweise besagt der 1. Kundenauftrag in der Tab. 5, daß der Kunde Nummer 3 vom Gut Nummer 2 drei (3) Ladeeinheiten in dem Zeitraum von 14.00 – 16.00 Uhr haben will. Die ermittelte Umschlagzeit für die drei Ladeeinheiten beträgt 8 Minuten usw.

Tabelle 5: Die Kundenaufträge

Auftragsnummer	Kunde	Gut	Menge	Umschlagzeit	Zeitfensterbeginn	Zeitfensterende
1	3	2	3	8	14.00	16.00
2	5	2	3	8	14.00	16.00
3	4	8	1	5	8.00	10.00
4	4	7	1	5	8.00	10.00
5	5	5	2	9	12.00	14.00
6	5	3	3	12	12.00	14.00
7	6	4	2	12	14.00	16.00
8	6	6	2	12	14.00	16.00
9	7	2	3	16	10.00	12.00
10	7	8	3	16	10.00	12.00
11	2	3	2	10	11.40	14.00
12	2	7	2	10	11.40	14.00
13	9	3	2	10	11.40	14.00
14	3	5	1	8	13.20	16.00
15	3	1	1	8	13.20	16.00
16	4	2	1	7	7.20	10.00
17	8	4	2	15	7.40	12.00
18	8	3	2	15	7.40	12.00
19	8	6	3	15	7.40	12.00
20	9	1	2	10	7.20	10.00

In Tabelle 5 ist die Zeiteinheit für den Umschlag Minute [min] und für das Zeitfenster Stunde [h]. Es kommt noch hinzu, daß die Fahrzeit in Minuten berechnet wird. Diese Zeiten sollen einheitlich sein. Für weitere Berechnungen wird die Minute als Zeiteinheit benutzt. Infolgedessen entspricht z.B. die Zeit 14.00 Uhr die Minute 840 des Tages ($14 \cdot 60$) usw.

Die Tabelle 6 zeigt die Verträglichkeit der Güter untereinander.

Tabelle 6: Die Verträglichkeitsmatrix

↓ Gutnummer →	1	2	3	4	5	6	7	8
1	1	1	0	0	0	1	1	0
2	1	1	0	0	0	1	0	0
3	0	0	1	0	0	0	0	0
4	0	0	0	1	1	0	0	1
5	0	0	0	1	1	0	0	1
6	1	1	0	0	0	1	1	1
7	1	0	0	0	0	1	1	1
8	0	0	0	1	1	1	1	1

In Tabelle 6 repräsentiert 1 die Verträglichkeit und 0 das Zusammenladeverbot der jeweiligen Güter.

Die Tabelle 7 enthält die Fahrzeiten zwischen den Kunden und dem Terminal (der Terminal ist Nummer 1).

Tabelle 7: Fahrzeitmatrix

↓ Kunden →	1	2	3	4	5	6	7	8	9	10
1	0	16	18	18	19	19	20	19	16	14
2	16	0	4	4	3	5	6	5	10	6
3	18	4	0	5	3	3	2	3	8	4
4	18	4	5	0	3	3	2	3	8	4
5	19	3	3	3	0	2	3	4	9	5
6	19	5	3	3	2	0	3	2	7	5
7	17	6	2	2	3	3	0	2	7	6
8	19	5	3	3	4	2	2	0	5	5
9	16	10	8	8	9	7	7	5	0	6
10	14	6	4	4	5	5	6	5	6	0

Angenommen wird, daß die Ladekapazität der Lkw 20 Ladeeinheiten beträgt. Es stehen zwei Fahrzeugtypen zur Verfügung. Das Gut Nummer 4 kann auf die Fahrzeuge des 2. Typs nicht geladen werden. Die Tourendauer beträgt 120 Minuten.

Unter Berücksichtigung der Kapazität, Beladbarkeit, Zeitfenster, Gesamtzeit und Verträglichkeit der Güter ist die benötigte Fahrtenanzahl zu ermitteln.

Das Ergebnis:

Fahrt-Nr.	Typ	Ladung	U-Zeit	F-Zeit	G-Zeit	Haltepunkte
1	2	9	36	42	78	3
2	1	7	32	44	76	3
3	2	6	31	41	72	2
4	2	7	36	39	75	3
5	2	5	29	42	71	3
6	2	2	15	38	53	1
7	1	2	15	38	53	1
8	2	3	17	42	59	2

Das Ergebnis im Detail

Fahrt-Nr.: 1 Ladung: 9 Auslastung: 45%

Haltepunkte: [0,3,5,6] (0 ist Ausgangspunkt „Lager“)

Aufträge: A1 [K3, G2, M3]; A2 [K5, G2, M3]; A8 [K6, G6, M2]; A15 [K3, G1, M1]

(Erläuterung: A1 [K3, G2, M3] heißt, das der 1. Auftrag kommt vom Kunden Nummer 3. Er will vom Gut Nummer 2 die Menge 3 Ladeeinheiten usw.).

Fahrt-Nr.: 2 Ladung: 7 Auslastung: 35%

Haltepunkte: [0,2,5,9]

Aufträge: A6 [K5, G3, M3]; A11 [K2, G3, M2]; A13 [K9, G3, M2]

Fahrt-Nr.: 3 Ladung: 6 Auslastung: 30%

Haltepunkte: [0,7,8]

Aufträge: A9 [K7, G2, M3]; A19 [K8, G6, M3]

Fahrt-Nr.: 4 Ladung: 7 Auslastung: 35%

Haltepunkte: [0,2,4,7]

Aufträge: A3 [K4, G8, M1]; A4 [K4, G7, M1]; A10 [K7, G8, M3]; A12 [K2, G7, M2]

Fahrt-Nr.: 5 Ladung: 5 Auslastung: 25%

Haltepunkte: [0,3,5,6]

Aufträge: A5 [K5, G5, M2]; A7 [K6, G4, M2]; A14 [K3, G5, M1]

Fahrt-Nr.: 6 Ladung: 2 Auslastung: 10%

Haltepunkte: [0,8]

Aufträge: A17 [K8, G4, M2]

Fahrt-Nr.: 7 Ladung: 2 Auslastung: 10%

Haltepunkte: [0,8]

Aufträge: A18 [K8, G3, M2]

Fahrt-Nr.: 8 Ladung: 3 Auslastung: 15%

Haltepunkte: [0,4,9]

Aufträge: A16 [K4, G2, M1]; A20 [K9, G1, M2]

Durch die Modifikation der Gegebenheiten und der Nebenbedingungen können andere Varianten probiert werden.

Variante 2:

Lkw-Kapazität = 20; Tourendauer = 120;
Beladbarkeit; Verträglichkeit;

Das Ergebnis:

Fahrt-Nr.	Typ	Ladung	U-Zeit	F-Zeit	G-Zeit	Haltepunkte
1	2	15	66	52	118	6
2	1	9	47	44	91	4
3	2	11	65	52	117	6
4	2	6	33	49	82	4

Das Ergebnis im Detail

Fahrt-Nr.: 1 Ladung: 15 Auslastung: 75%

Haltepunkte: [0,3,4,5,6,7,8]

Aufträge: A1 [K3, G2, M3]; A2 [K5, G2, M3]; A8 [K6, G6, M2]; A9 [K7, G2, M3];
A16 [K4, G2, M1]; A19 [K8, G6, M3]

Fahrt-Nr.: 2 Ladung: 9 Auslastung: 45%

Haltepunkte: [0,2,5,8,9]

Aufträge: A6 [K5, G3, M3]; A11 [K2, G3, M2]; A13 [K9, G3, M2]; A18 [K8, G3, M2]

Fahrt-Nr.: 3 Ladung: 11 Auslastung: 55%

Haltepunkte: [0,3,4,5,6,7,8]

Aufträge: A3 [K4, G8, M1]; A5 [K5, G5, M2]; A7 [K6, G4, M2]; A10 [K7, G8, M3];
A14 [K3, G5, M1]; A17 [K8, G4, M2]

Fahrt-Nr.: 4 Ladung: 6 Auslastung: 30%

Haltepunkte: [0,2,3,4,9]

Aufträge: A4 [K4, G7, M1]; A12 [K2, G7, M2]; A15 [K3, G1, M1]; A20 [K9, G1, M2]

Variante 3:

Tourendauer = 180; Kapazität;

Das Ergebnis:

Fahrt-Nr.	Typ	Ladung	U-Zeit	F-Zeit	G-Zeit	Haltepunkte
1	X	20	84	45	129	4
2	X	20	120	51	171	6
3	X	1	7	36	43	1

Das Ergebnis im Detail

Fahrt-Nr.: 1 Ladung: 20 Auslastung: 100%

Haltepunkte: [0,3,5,7,8]

Aufträge: A1 [K3, G2, M3]; A2 [K5, G2, M3]; A5 [K5, G5, M2]; A6 [K5, G3, M3];
A9 [K7, G2, M3]; A10 [K7, G8, M3]; A19 [K8, G6, M3]

Fahrt-Nr.: 2 Ladung: 20 Auslastung: 100%

Haltepunkte: [0,2,3,4,6,8,9]

Aufträge: A3 [K4, G8, M1]; A4 [K4, G7, M1]; A7 [K6, G4, M2]; A8 [K6, G6, M2];
A11 [K2, G3, M2]; A12 [K2, G7, M2]; A13 [K9, G3, M2]; A14 [K3, G5, M1];
A15 [K3, G1, M1]; A17 [K8, G4, M2]; A18 [K8, G3, M2]; A20 [K9, G1, M2]

Fahrt-Nr.: 3 Ladung: 1 Auslastung: 5%

Haltepunkte: [0,4]

Aufträge: A16 [K4, G2, M1]

In der Tabelle 9 wird eine Zusammenfassung von (9) möglichen Varianten eingeführt:

Tabelle 9: Zusammenfassung der Ergebnisse

Varianten	Nebenbedingungen					Anzahl der benötigten Fahrten	Anzahl der Haltepunkte
	Kundenzeitfenster	Güterverträglichkeit	Beladbarkeit	Lkw-Kapazität	Gesamtzeit		
1	ja	ja	ja	20	120	8	18
2	ja	ja	ja	10	120	8	18
3	nein	ja	ja	20	120	4	20
4	nein	nein	ja	20	120	4	16
5	nein	ja	ja	10	120	6	19
6	nein	ja	nein	20	180	3	11
7	nein	nein	ja	10	180	5	20
8	nein	nein	ja	20	180	3	16
9	nein	nein	nein	20	180	3	11

4. Schlußfolgerungen

1. Die Versorgung der Städte mit Gütern über Transit-Terminal trägt zur Reduzierung des Lieferverkehrs bei. Es ist sichtbar, daß sowohl die Anzahl der benötigten Fahrten als auch die Anzahl der Haltepunkte immer kleiner als die Anzahl der Kundenaufträge ist, d.h. die Tourenanzahl und die Sendungsanzahl werden verdichtet.
2. Die Härte der Nebenbedingungen – insbesondere das Zeitfenster bei den Kunden (siehe 1. Variante) – beeinflusst maßgeblich den Reduzierungsgrad des Lieferverkehrs.
3. Die Entzerrung der Zeitfenster bei den Kunden (Anlieferzeit) trägt maßgeblich dazu bei, den städtischen Güterverkehr zu bündeln.
Je größer die Kapazität der Fahrzeuge ist, desto geringer wird die Anzahl der benötigten Fahrten, wenn die Zeitfensterrestriktion dies erlaubt.
4. Durch dieses Programm kann ein objektiver Kompromiß für die Lösungsfindung der Transportprobleme im allgemeinen gefunden werden. Insbesondere hilft das Programm bei der kurz- und langfristigen Planung der städtischen Ver- und Entsorgungsprobleme.
5. Das Programm ist ein Werkzeug für strategische und operative Entscheidungsfindung zum Betreiben eines Transit-Terminals und hilft z.B. bei der Planung
 - des Lagerstandortes,
 - des Fuhrparkes,
 - des Belieferungszeitfensters und
 - bei der täglichen Disposition der Kundenaufträge unter Beachtung der Nebenbedingungen.

Literaturverzeichnis

- /1/ *Lieber, H., Woda, A.*
Technologie des Straßentransports, Verlag Heinrich Vogel, München, 1992.
- /2/ *Martello, S., Toth, P.*
Knapsack Problems Algorithms and Computer Implementations,
Verlag John Wiley & Sons Chichester, New York, Toronto,... 1990.
- /3/ *Domschke, W.*
Logistik: Rundreisen und Touren, 3. Auflage, Oldenbourg Verlag GmbH München, 1990.
- /4/ *Ziegler, H.-J., Niemeier, H.-V. et al*
Computer gestützte Transport- und Tourenplanung, Expert Verlag, 1988.
- /5/ *Engele, G.*
Simultane Standorte- und Tourenplanung (Diss. A), Carl Heymanns Verlag KG, Köln,
Bonn, München, 1980.
- /6/ *AL-Daas, M.*
Auswahl geeigneter Strategien für die Güterversorgung von Städten, Kommunen und
Ballungsgebieten. In: Hebezeuge und Fördermittel, Berlin 33 (1993) 7/8, S. 305-310.
- /7/ *Powell, W. et al.*
An Optimization-Based Heuristic for Vehicle Routing and Scheduling with Soft Time
Window Constraints. In: Transportation Science, Vol. 26, No. 2, May 1992, PP. 69-85.
- /8/ *Bull, A. G.*
Artificial Intelligence, Charme – Eine Einführung, S. 7-1.
- /9/ *Clarck, G. / Wright, J. W.*
Scheduling of Vehicles from a central Depot to a Number of Delivery Points.
In: Operations Research 12 (1964) P. 568-581.
- /10/ *Gillett, B. E. / Miller, L. R.*
A Heuristic Algorithm for the Vehicle-Despatch Problem, In: Operations Research 22
(1974) P. 340-349.

Abstract

The phrase "Logistical Transport Problem" is created to describe the task of City-Logistics. This problem encloses the following subproblems: the classical Transport Problem, the Location Problem, Knapsack Problem, Routing Problem and Travelling Salesman Problem. City-Logistics has generally the following feature: the goods flows from outside of the city would be driven into a city-terminal "transit terminal". They would be there picked up and then they will be distributed in the city by environment friendly vehicles. The effect of this convincing is reducing the number of moving vehicles in the city.

The collected goods need to be transported from the terminal to a set of customers in the city. The distribution of goods must take place subject to many restrictions. Among these restrictions are: each customer places a fixed size shipment and specifies a time window within the supplying can be taken place. All shipments are assumed to be less than the vehicle capacity, and therefore, more than one customer can be supplied with the same vehicle. The goods, which are placed on the same vehicle must be sociable (compatible), and therefore, can one customer be serviced with more than one vehicle. The total freight to be carried by a vehicle must be within its capacity (full truckload). Some goods must be transported with special vehicles (e.g. cool goods) etc. These and other restrictions show the complexity of the task logistical Transport Problem. The huge number of restrictions forces us to solve this problem with a constraints-based-programming, which is an instrument of artificial intelligence. This paper includes a discussion of the Logistical Transport Problem and its solution with a tool of artificial intelligence.