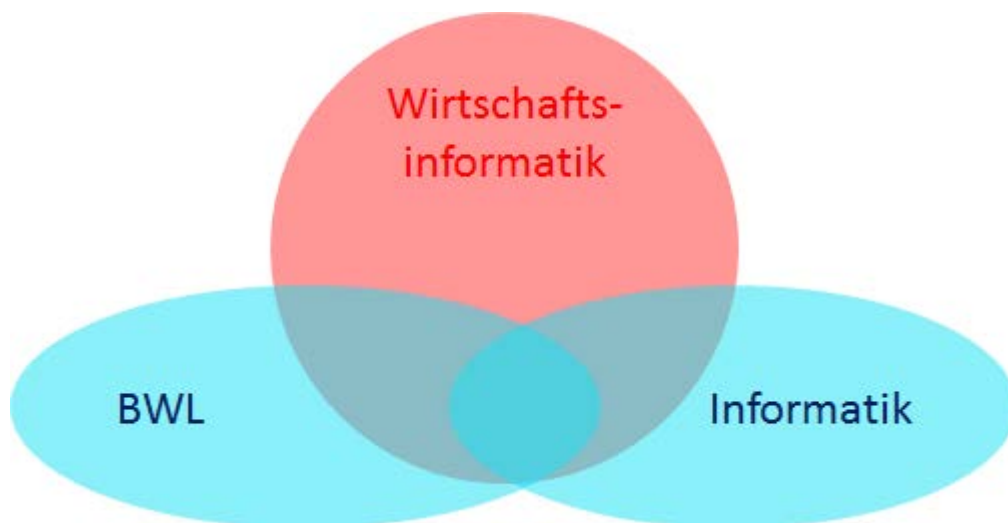


# AK WI

## Anwendungen und Konzepte der Wirtschaftsinformatik

Nummer 2 (2014)



# Editorial

Liebe Leserinnen und Leser,

vor Ihnen liegt die zweite Ausgabe des E-Journals **Anwendungen und Konzepte in der Wirtschaftsinformatik (AKWI)**. Die erste Ausgabe erschien vor exakt einem halben Jahr im September 2013. Damit haben wir unser Ziel zwei Ausgaben pro Jahr herauszugeben bis jetzt erfüllt. Darauf sind wir stolz.

Dieses Heft enthält Beiträge in den Rubriken Grundlagen und Praxis sowie eine Buchbesprechung. Die Themen der Beiträge decken die Bereiche Cloud-Computing, Geschäftsprozesse, Optimierung, Simulation, Logistik und e-Business ab. Sie spiegeln damit die Vielfalt der Arbeitsgebiete in der Wirtschaftsinformatik wieder. Die Autoren kommen aus den Hochschulen der angewandten Wissenschaft und aus Unternehmen. Auch aus den Titeln wird sofort die anwendungsorientierte Arbeit der Autoren deutlich.

Alle Beiträge wurden in den Monaten Oktober bis Dezember 2013 eingereicht, von 2 unabhängigen Gutachtern begutachtet und von den Autoren anschließend überarbeitet. Dass dieser Prozess bei keiner Arbeit länger als 5 Monate gedauert hat, betrachte ich als eine bemerkenswerte Leistung der Autoren, Gutachter und Herausgeber. Dabei ist anzumerken, dass alle Beteiligten ehrenamtlich für die Zeitschrift arbeiten. Dafür möchte ich Ihnen danken.

Die Zeitschrift wird weiterhin als E-Journal an der Hochschule Luzern unter Federführung von Konrad Marfurt gehostet. Auch dafür gebührt Ihm Dank.

Unsere Zeitschrift steht kostenfrei unter <http://akwi.hswlu.ch/index.php/AKWI> im Netz zur Verfügung. Für unsere Autoren entstehen durch die Publikation keine Kosten, allerdings erhalten sie auch keine Honorare. Damit wir unter diesen Bedingungen erfolgreich arbeiten können, reichen die Autoren druckfertige Manuskripte, die unserer [Formatvorlage](#) genügen, in deutscher oder englischer Sprache ein. Außerdem bitten wir unsere Autoren um eine [Einverständniserklärung zur Publikation](#) und eine Selbsteinschätzung welcher der Rubriken, *Grundlagen*, *Trends*, *Praxis*, *Kurz erklärt*, *Buchbesprechung* oder *Abschlussarbeit* ihr Beitrag zugeordnet werden soll. Bei Abschlussarbeiten gehen wir davon aus, dass es sich um Zusammenfassungen hervorragender Thesearbeiten handelt, die zusammen mit dem betreuenden Hochschullehrer eingereicht werden. Nach der Einreichung beginnt sofort unser Begutachtungsprozess.

Nach dieser ausführlichen Beschreibung des Procederes der Beitragseinreichung hoffe ich Sie zu einer solchen motiviert zu haben. In diesem Sinne verbleiben wir, stellvertretend für die Herausgeber

Christian Müller und Konrad Marfurt



Christian Müller



Konrad Marfurt

# EFFIZIENTE UND EFFEKTIVE LAGERHALTUNG - IT-GESTÜTZTE DURCHFÜHRUNG VON BESTANDSMANAGEMENT-PROJEKTEN

Professor Dr.-Ing. Frank Herrmann

Hochschule Regensburg - University of Applied Sciences Regensburg  
Innovation and Competence Centre for Production Logistics and Factory Planning (IPF)  
PO box 120327, 93025 Regensburg, Germany  
E-Mail: Frank.Herrmann@OTH-Regensburg.de

## STICHWORTE

Lagerhaltungsverfahren, u.a. optimal, Leitfaden, durch Softwaresystem umsetzbar, industrielle Anwendung.

## ABSTRAKT

Ein effizientes und effektives Bestandsmanagement ist zentral für geringe Auswirkungen durch fehlende Produkte und geringe Lagerhaltungskosten. Die für viele Unternehmen geeigneten Formeln zum Bestandsmanagement und seine produktspezifische Anwendung werden in einem Leitfaden zusammengefasst und anhand eines realen Anwendungsfalls erläutert. Seine manuelle Durchführung ist sehr aufwendig. Es wird erläutert, welche Schritte des Leitfadens wie durch ein Programm durchgeführt werden können.

## 1. EINLEITUNG

Eine sehr wichtige Zielsetzung in der Steuerung von Unternehmen ist die Realisierung von minimalen Beständen, da vermeidbare Bestände unnötige Lagerkosten bewirken. So wird in Unternehmen oftmals eine bestandslose Fertigung gefordert; auch ausgedrückt durch eine „Just-In-Time“-Produktion. Dies führte zur Entwicklung von ausgefeilten Lagerhaltungsverfahren. Danach kann jedes Lager, welches stochastischen Nachfragen und Lieferzeiten gegenübersteht, optimal betrieben werden. Dies setzt voraus, dass diese Daten, und auch die notwendigen Kostensätze, mit einem hohen Aufwand genau erhoben werden. Viele Produkte verursachen jedoch nur geringe Lagerhaltungskosten, so dass einfachere, aber nicht optimale Lagerhaltungsverfahren in der industriellen Praxis eingesetzt werden sollten. Hierzu werden geeignete Kriterien und Verfahren vorgeschlagen, und ihre konkrete Anwendung in einem Bestandsmanagementprojekt in einem Unternehmen wird in einem Leitfaden beschrieben. Die weitgehende automatische Durchführung des Leitfadens durch ein Softwaresystem wird vorgestellt.

## 2. LAGERHALTUNGSVERFAHREN

Mit Lagerhaltungsverfahren wird eine möglichst kostengünstige Lagerhaltung der einzelnen Produkte angestrebt.

### Das klassische Model

Wäre die Nachfrage nach einem Produkt eines Unternehmens pro Zeiteinheit, z.B. einem Tag, konstant, nämlich  $d$ , so liese sich deterministisch berechnen, wann das Lager leer ist. Ist die Produktionszeit eines Produkts ebenfalls konstant, nämlich  $t_p$  Zeiteinheiten, so müsste  $t_p$  Zeiteinheiten bevor das Lager leer ist mit der Produktion von Produkten begonnen werden. Erst wenn ein Produktionsauftrag (komplett) fertig gestellt worden ist, wird die komplette produzierte Menge ins Lager transportiert und steht danach komplett zur Verfügung; es handelt sich um eine so genannte geschlossene Produktion und der zwischenzeitlich auftretende Bestand wird bei den Produktionseinrichtungen gelagert und als „work-in-process“-Bestand bezeichnet. Die Produktionsmenge eines Produktionsauftrags sei  $q$ ; er wird als Produktionslos bezeichnet. Für seine Durchführung fallen einmalig fixe Rüstkosten von  $K$  Geldeinheiten an. Die Lagerung einer Einheit des Produkts führt zu Lagerhaltungskostensatz von  $h$  Geldeinheiten. Für die optimale Lösung des Problems lässt sich zeigen, dass  $q_{opt}$  eine Konstante ist. Seine Höhe ist bestimmt durch die Rüstkosten, die Lagerhaltungskosten sowie die Produktionsgeschwindigkeit, nämlich durch

$$q_{opt} = \sqrt{\frac{2 \cdot K \cdot d}{h} \cdot \frac{p}{(d+p)}}; \text{ eine detaillierte Herleitung ist in}$$

(Herrmann 2009) beschrieben und das Modell wird ebendort sowie in der Literatur unter der Bezeichnung klassisches Losgrößenmodell mit konstantem Bedarf und endlicher Produktionsgeschwindigkeit diskutiert. Damit

ergibt sich der in Abbildung 1 angegebene „Sägezahn ähnliche“ Verlauf. Er ist charakteristisch für das deterministische Bestandsmanagement. Wie Abbildung 1 zeigt, ist der Zeitpunkt für das Aufsetzen des Produktionsauftrags bestimmt durch das Erreichen eines bestimmten Lagerbestands. Dieser Bestand wird als Bestellpunkt  $s$  bezeichnet. Analytisch ist er bestimmt durch  $s = t_p \cdot d$ .

Wiederbeschaffungszeitraums im Lager vorliegt, die überhaupt in einem Wiederbeschaffungszeitraum benötigt werden kann (allerdings besitzt beispielsweise bei einer Normalverteilung jeder Wert eine, wenn auch eine geringe, Wahrscheinlichkeit). Dies führt zu der Frage, ob es nicht ökonomischer ist, eine geringere Menge zu lagern, um dadurch (gegenüber dem Extremszenario) Lagerkosten einzusparen, aber zwangsläufig Fehlmengensituationen zuzulassen.

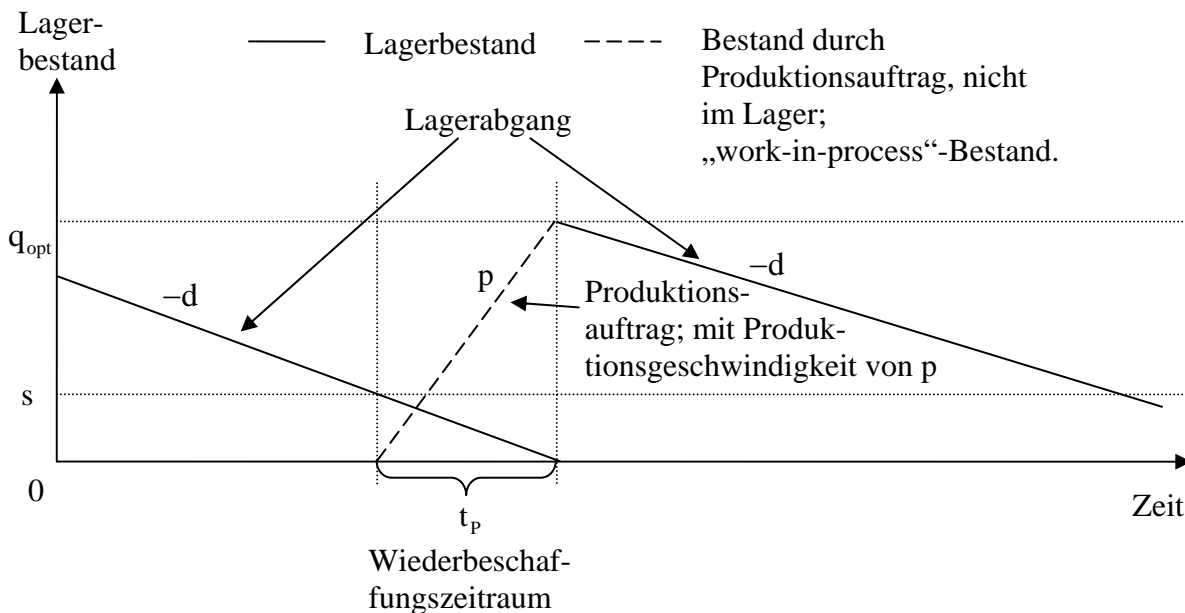


Abbildung 1: Bestandsentwicklung bei deterministischer Nachfrage und Produktionszeit

### Stochastischer Größen und Problemstellung

Tatsächlich treten zufällige Produktionszeiten bzw. Lieferzeiten und zufällige Nachfragen auf. Dies führt zu einem so genannten stochastischen Bestandsmanagement. Damit ist die Nachfrage in dem Produktionszeitraum, bzw. in dem Wiederbeschaffungszeitraum,  $t_p$  nicht mehr konstant, sondern zufällig. Vor allem ist sie zum Zeitpunkt des Aufsetzens eines Produktionsauftrags nicht bekannt. Folglich ist nicht zu erwarten, dass der Lagerbestand am Ende eines Wiederbeschaffungszeitraums leer ist; also genau zu dem Zeitpunkt, wenn die durch den Produktionsauftrag, zu diesem Wiederbeschaffungszeitraum, hergestellte Menge im Lager eintrifft und verfügbar ist. Somit ist zu erwarten, dass Wiederbeschaffungszeiträume mit einer Fehlmenge oder mit einem (noch vorhandenen) Lagerbestand auftreten und entsprechende Kosten verursachen. Mit einem Fehlmengenkostensatz  $p$  pro Mengeneinheit (ME) unbefriedigter Nachfrage werden Fehlmengenkosten berechnet. Verantwortlich für Fehlmengenkosten sind beispielsweise zusätzliche Kosten für die Extralieferung einer benötigten Mengeneinheit oder der Einnahmeverlust, wenn die Nachfrage verloren geht. Es sei angemerkt, dass das Auftreten von Fehlmenen bei einem stochastischen Bedarf nur dann völlig ausgeschlossen werden kann, wenn stets die maximale Menge zu Beginn eines

Entscheidend ist folglich die Minimierung der Summe aus Fehlmenen- und Lagerkosten. Daneben treten noch Bestellkosten auf, so dass diese drei ein Optimierungsproblem bestimmen. Bereits mit den wegweisenden Arbeiten in den 1950er und 1960er Jahren konnte gezeigt werden, dass dieses Optimierungsproblem durch eine so genannte  $(s,q)$ -Lagerhaltungspolitik optimal gelöst werden kann. Sie arbeitet nach der folgenden Entscheidungsregel: Zu jedem Zeitpunkt  $t$ , wenn der für die Disposition verfügbare Lagerbestand  $I_t^D$  einen vorgegebenen Bestellbestand  $s$  erreicht hat, wird eine Lagerbestellung der Größe  $q$  ausgelöst, die nach einer Lieferzeit  $l$  im Lager eintrifft. Da in der industriellen Praxis ein Lager beliebig lange betrieben wird, wird seine im Mittel optimale Steuerung angestrebt. Dafür wurde nachgewiesen, dass es sich sowohl beim Bestellbestand  $s$  als auch bei der Bestellmenge  $q$  um Konstanten handelt. Es sei betont, dass diese Regel auch für einen endlichen Planungshorizont gilt; dann sind  $s$  und  $q$  allerdings nicht konstant.

Die bisherige Betrachtung unterstellt eine kontinuierliche Bestandsüberwachung. Typischerweise erfolgt die Überprüfung des Bestellbestands in der industriellen Praxis zu festen regelmäßigen Abständen, z.B. zu Beginn eines Tages, so dass Bestandsreduzierungen innerhalb eines Überwachungsintervalls lediglich zeitlich verzögert erkannt

werden. Deswegen zerfällt die Zeitachse in aufeinander folgende Perioden gleicher Länge, an deren Beginn beim Erreichen oder Unterschreiten des Bestellbestands eine Bestellung ausgelöst wird. Diese Bestellung ist für die Disposition unmittelbar (als disponibler Bestand) verfügbar, physisch aber erst nach einer Lieferzeit. Damit hat die Bestandsentwicklung den in Abbildung 2

„undershoot“). Für diesen Fall wird ein im Mittel optimaler Lagerbetrieb erreicht, sofern nicht eine feste Bestellmenge  $q$  verwendet wird, sondern mit einer variablen Bestellmenge der disponible Bestand auf das so genannte Bestellniveau  $S$  angehoben wird. Ist  $I_t^D$  wieder der zum Zeitpunkt  $t$  für die Disposition verfügbare Lagerbestand, so

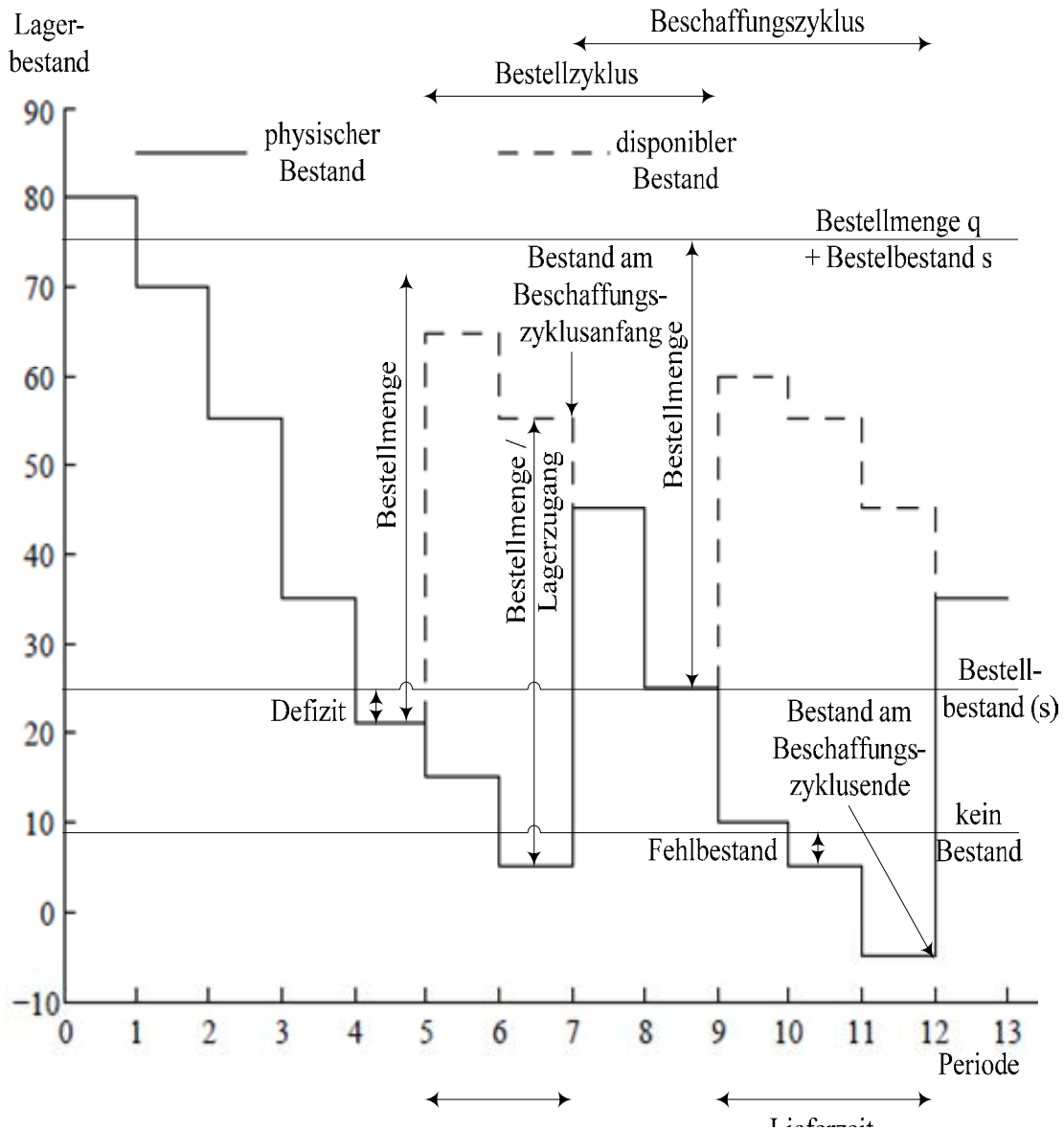


Abbildung 2: Entwicklung des Lagerbestands bei Einsatz einer (s,q)-Lagerhaltungspolitik

angegebenen Bedarfsverlauf; in ihr ist der Bestand am Periodenende eingetragen und es wird eine Lagerentnahme zu Periodenbeginn unterstellt. Die Fehlmenge wird vorgemerkt und beim Eintreffen einer Lieferung bevorzugt ausgeliefert; in Abbildung 2 ist sie als negativer Bestand eingezeichnet.

Das Beispiel zeigt, dass der Bestand zu Beginn einer Periode auch unterhalb des Bestellbestands liegen kann und die dabei auftretende zufällige Differenz zwischen dem Bestellbestand und dem Lagerbestand heißt Defizit (u,

hat die variablen Bestellmenge die Höhe  $(S - I_t^D)$ . Diese Politik wird als (s,S)-Lagerhaltungspolitik bezeichnet.

Diese im Mittel optimale Steuerung des Lagerbetriebs ergibt sich, sofern der stochastische Prozess zu den Periodenbedarfen und bei zufälligen Lieferzeiten auch der zu den Lieferzeiten durch stochastisch unabhängige und identisch verteilte Zufallsvariablen mit einem endlichen Mittelwert und einer endlichen Streuung beschrieben werden kann. Diese Forderung umfasst die Bedingungen an einen stationären stochastischen Prozess und wird

deswegen als quasi stationär bezeichnet. Bei einem stationären stochastischen Prozess sind (unter anderem) der Mittelwert und die Standardabweichung im Zeitablauf konstant. Bezogen auf die Nachfragemengen sei  $D$  die zu diesen Kenngrößen (gemeinsame) Zufallsvariable mit dem Erwartungswert  $E(D)$  und der Standardabweichung  $\sigma(D)$  sowie der Verteilungsfunktion  $\Phi_D$ . Entsprechend ist bei zufälligen Lieferzeiten  $L$  die zu diesen Kenngrößen (gemeinsame) Zufallsvariable mit dem Erwartungswert  $E(L)$  und der Standardabweichung  $\sigma(L)$  sowie der Verteilungsfunktion  $\Phi_L$ . Unter der Annahme, dass eine (s,S)-Lagerhaltungspolitik bzw. eine (s,q)-Lagerhaltungspolitik verwendet wird, kann nun ein Entscheidungsmodell gebildet werden, durch das der Bestellbestand  $s$  und das Bestellniveau  $S$  bzw. die Bestellmenge  $q$  bestimmt wird; ein solches Entscheidungsmodell ist im Detail in (Herrmann 2009) beschrieben worden. Wie oben bereits angedeutet worden ist, ist die (s,S)-Lagerhaltungspolitik kostengünstiger als die eine (s,q)-Lagerhaltungspolitik. Da in der industriellen Praxis gerne mit konstanten Bestellmengen gearbeitet wird, wird dieser Arbeit eine (s,q)-Lagerhaltungspolitik zugrunde gelegt. Für einen Vergleich dieser beiden Politiken sei auf (Tempelmeier 2005) verwiesen.

Für die industrielle Praxis problematisch ist die Festlegung der Fehlmengenkosten. Wenn überhaupt, ist sie nur mit einem hohen Aufwand möglich; demgegenüber liegen in der Regel genaue Informationen über Bestellkosten und Lagerkosten vor. Deswegen wird alternativ mit Servicegraden gearbeitet, mit denen üblicherweise die Wahrscheinlichkeit für

- das Eintreten eines Fehlmengen- oder Fehlbestandsereignisses,
- die Höhe der Fehlmenge oder des Fehlbestands oder
- die Dauer einer Lieferunfähigkeitssituation

beschränkt wird. Im Folgenden wird der lieferantenfokussierte mengenbezogene  $\beta$ -Servicegrad betrachtet. Für weitere Kriterien sei auf [Temp02] bzw. (Tempelmeier 2005) sowie vor allem auf (Silver et al. 1998) verwiesen. Er stellt sicher, dass  $\beta$  den Anteil der erwarteten befriedigten Nachfrage an der insgesamt erwarteten Nachfrage beschreibt; also formal:

$$\beta = \frac{E(\text{befriedigte Nachfrage einer Periode})}{E(\text{Gesamtnachfrage einer Periode})} \quad \text{Mit}$$

$$E(\text{befriedigte Nachfrage einer Periode}) = E(D) - E(\text{Fehlmenge pro Periode}) \quad \text{gilt: } \beta = \frac{E(D) - E(F)}{E(D)}, \text{ wobei}$$

$F$  die Fehlmenge pro Periode ist; weitere Informationen zu dieser Kennzahl finden sich beispielsweise in [Schn81], [Temp02] und (Robrade 1991).

Da der Bestellbestand positiv ist, kann eine Fehlmenge und damit eine Lieferunfähigkeit nur während der Wiederbeschaffungszeit und in der Periode vor dem Auslösen einer Bestellung auftreten. Mit anderen Worten

sind somit für die Analyse der Lieferfähigkeit ausschließlich die Wiederbeschaffungszeiträume relevant, weswegen diese auch als Risikozeiträume bezeichnet werden.

Treten zufällige Lieferzeiten auf, so könnte eine spätere Bestellung eine so kurze Lieferzeit haben, dass ihre Bestellmenge vor derjenigen zu einer früheren Bestellmenge eintrifft; also eine Bestellung eine andere überholt. Dann kann das Lager nicht mehr optimal betrieben werden; dies ist nach Ansicht des Autors in (Hochstädter 1969) schön ausgearbeitet worden. Deswegen wird bei stochastischen Lieferzeiten davon ausgegangen, dass, falls mehrere Lagerbestellungen gleichzeitig ausstehen, sie in der Reihenfolge, in der sie aufgegeben worden sind, im Lager eintreffen. Für diesen Fall kann nachgewiesen werden, dass die Lieferfähigkeit ausschließlich dadurch bestimmt ist, ob die Summe aus dem gesamten Bedarf ( $y$ ) in der Wiederbeschaffungszeit und dem Defizit ( $u$ ) (gegenüber dem Bestellbestand) zu Beginn der Wiederbeschaffungszeit den Bestellbestand  $s$  überschreitet. Damit ist die Wirkung eines Bestellbestands  $s$  auf die Lieferfähigkeit unabhängig von dem Vorliegen von offenen Bestellungen zu einem Bestellzeitpunkt; mit anderen Worten: für die Analyse der Wirkung eines Bestellbestands  $s$  auf die Lieferfähigkeit kann angenommen werden, dass zum Bestellzeitpunkt keine Bestellung aussteht.

### Berechnungsformeln für das Bestellniveau

Da außerhalb eines Wiederbeschaffungszeitraums keine Lieferunfähigkeit auftritt, darf angenommen werden, dass die Lagerbestandsentwicklung außerhalb eines Wiederbeschaffungszeitraums konstant und kontinuierlich verläuft. Um die Abweichungen von den tatsächlichen Werten zu minimieren, wird diese Konstante gleich dem Durchschnitt der Periodenbedarfe gesetzt, also dem Erwartungswert  $E(D)$ , wie dies in Abbildung 3 dargestellt ist (es sei daran erinnert, dass die Zufallsvariable  $D$  die Periodennachfrage beschreibt).

Zu einer konkreten Bestellung mit einer zufälligen Lieferzeit ( $l$ ) liegt eine zufällige Summe ( $y^*$ ) aus dem Defizit ( $u$ ) (gegenüber dem Bestellbestand) zu Beginn der Wiederbeschaffungszeit und dem gesamten Bedarf ( $y$ ) in der Wiederbeschaffungszeit vor, die als Gesamtnachfrage  $y^* = y + u$  in der Wiederbeschaffungszeit bezeichnet wird, und die wie Abbildung 3 zeigt, kleiner oder genauso groß wie ihr durchschnittlicher Wert  $(E(U) + E(D) \cdot E(L))$

oder sogar größer als der Bestellbestand  $s_{opt}$  sein kann; in Abbildung 3 ist der Fall  $y^* = E(U) + E(D) \cdot E(L)$  eingezeichnet (es sei daran erinnert, dass die Zufallsvariable  $U$  das Defizit und die Zufallsvariable  $L$  die Lieferzeit beschreibt). Das stochastische Verhalten von den auftretenden Summen ( $y^*$ ) wird durch eine Zufallsvariable

$Y^*$  beschrieben. Zunächst wird eine stetige Zufallsvariable mit der Dichte  $\phi_{Y^*}$  angenommen. Bewirkt eine konkrete

zufällige Summe  $y^*$  einen Fehlbestand (es gilt:  $y^* \geq s_{opt}$ ), so hat dieser die Höhe  $(y^* - s_{opt})$ . Damit ist

$$\beta = \frac{E(D) - E(F)}{E(D)} \text{ ergibt: } \beta = 1 - \frac{E(F_{Y^*}(s_{opt}))}{q_{opt}}$$

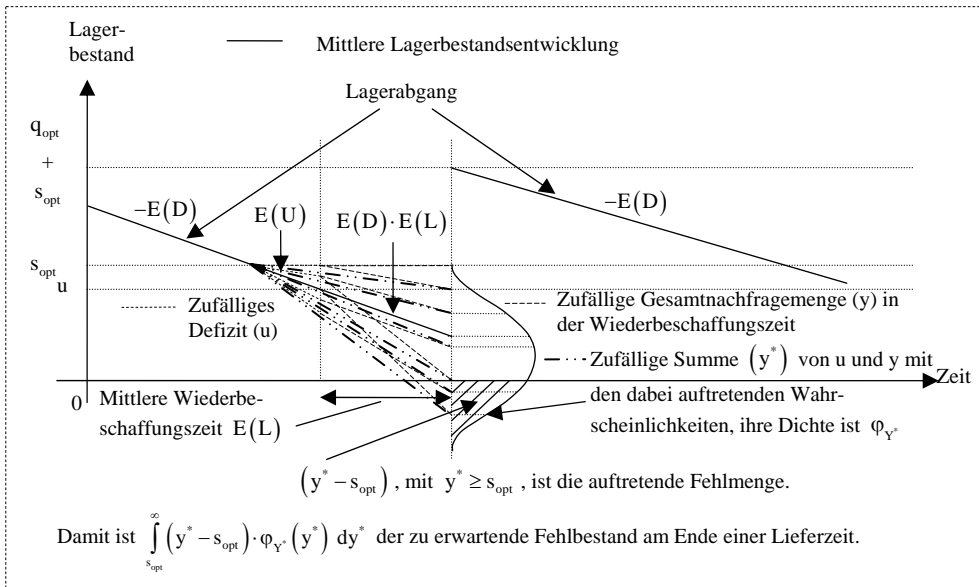


Abbildung 3: Mittlere Bestandsentwicklung bei stochastischer Nachfrage und Wiederbeschaffungszeit sowie beispielhafte zufällige Gesamt-Nachfragemengen

$E(I_{Y^*}^{f,End}(s_{opt})) = \int_{s_{opt}}^{\infty} (y^* - s_{opt}) \cdot \varphi_{Y^*}(y^*) dy^*$  der zu erwartende Fehlbestand am Ende einer Lieferzeit bzw. eines Beschaffungszyklus. Ist sogar  $y^* \geq s_{opt} + q_{opt}$ , so liegt eine Fehlmenge zu Beginn eines Beschaffungszyklus vor. Damit ist der zu erwartende Fehlbestand am Anfang von einem Beschaffungszyklus bestimmt durch

$$E(I_{Y^*}^{f,Anf}(s_{opt})) = \int_{s_{opt}+q_{opt}}^{\infty} (y^* - s_{opt} - q_{opt}) \cdot \varphi_{Y^*}(y^*) dy^*$$

Folglich ist  $E(F_{Y^*}(s_{opt})) = E(I_{Y^*}^{f,End}(s_{opt})) - E(I_{Y^*}^{f,Anf}(s_{opt}))$  die zu erwartende Fehlmenge von einem Beschaffungszyklus. Da damit alle auftretenden (zu erwartenden) Fehlmenen berücksichtigt sind, ergibt ihre Division durch die zu erwartende Anzahl an Perioden in einem Beschaffungszyklus die zu erwartende Fehlmenen pro Periode. Da die Fehlmenge annahmegemäß vorgemerkt wird, ist die mittlere Lagerbestandsentwicklung kontinuierlich und konstant und zwar mit dem Erwartungswert  $E(D)$  als konstanten Bedarf. Sie hat den in Abbildung 4 dargestellten Verlauf. Daraus folgt, dass die Bestellzyklen und die Beschaffungszyklen im Mittel den gleichen Verbrauch haben und dieser gleich der Bestellmenge  $q_{opt}$  ist. Damit ist  $\frac{q_{opt}}{E(D)}$  die zu erwartende

Anzahl an Perioden in einem Beschaffungszyklus und folglich ist die zu erwartende Fehlmenge pro Periode

gerade  $\frac{E(F_{Y^*}(s_{opt}))}{q_{opt}} \cdot E(D)$ . Sein Einsetzen in

Damit lautet die konkrete Berechnungsformel für den Fall einer kontinuierlichen Zufallsvariable  $Y^*$ :

$$\beta(s_{opt}) = 1 - \frac{\int_{s_{opt}}^{\infty} (y^* - s_{opt}) \cdot \varphi_{Y^*}(y^*) dy^* - \int_{s_{opt}+q_{opt}}^{\infty} (y^* - s_{opt} - q_{opt}) \cdot \varphi_{Y^*}(y^*) dy^*}{q_{opt}}$$

Ist  $Y^*$  diskret, so berechnet sich der Erwartungswert des Fehlbestands am Anfang eines Beschaffungszyklus durch

$$E(I_{Y^*}^{f,Anf}(s_{opt})) = \sum_{y^*=s_{opt}+q_{opt}+1}^{y_{max}} (y^* - s_{opt} - q_{opt}) \cdot P(Y^* = y^*)$$

und am Ende eines Beschaffungszyklus durch

$$E(I_{Y^*}^{f,End}(s_{opt})) = \sum_{y^*=s_{opt}+1}^{y_{max}} (y^* - s_{opt}) \cdot P(Y^* = y^*), \text{ wobei bei}$$

nicht nach oben beschränkten zufälligen Summen  $y^* y_{max}$  durch  $\infty$  ersetzt wird. Dann gilt für den  $\beta$ -Servicegrad:

$$\beta(s_{opt}) = 1 - \frac{\sum_{y^*=s_{opt}+1}^{y_{max}} (y^* - s_{opt}) \cdot P(Y^* = y^*) - \sum_{y^*=s_{opt}+q_{opt}+1}^{y_{max}} (y^* - s_{opt} - q_{opt}) \cdot P(Y^* = y^*)}{q_{opt}}$$

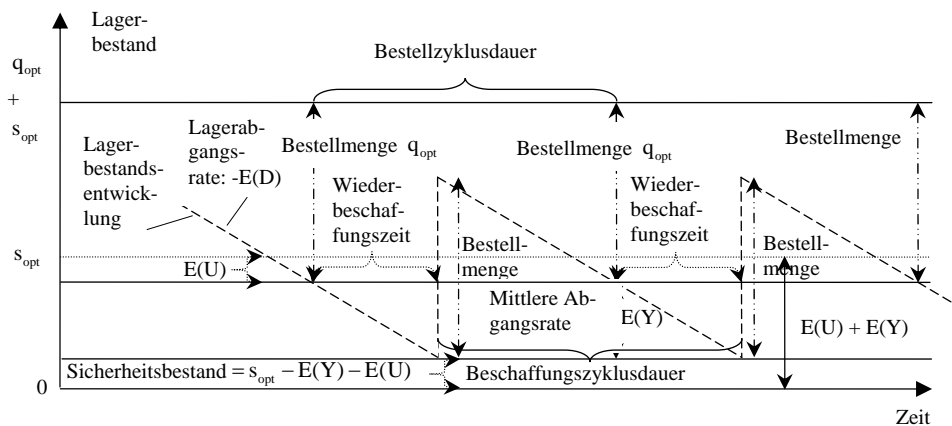


Abbildung 4: Entwicklung des mittleren physischen Lagerbestands bei einer (s,q)-Lagerhaltungspolitik

### Simultane Berechnung von Bestellniveau und Losgröße

Für die Bestimmung von  $s_{opt}$  ist zu berücksichtigen, dass die erwartete Fehlmengene  $E(F_{Y^*}(s_{opt}))$  keine stetige Funktion ist; es also möglich ist, dass kein  $s_{opt}$  mit  $E(F_{Y^*}(s)) = (1 - \beta(s)) \cdot q_{opt}$  existiert. Da  $E(F_{Y^*}(s))$  eine monoton fallende Funktion ist, ist der gesuchte Bestellbestand  $s_{opt}$  das kleinste  $s$  mit  $E(F_{Y^*}(s)) \leq (1 - \beta(s)) \cdot q_{opt}$ .

Entscheidend für die Berechnung des  $\beta$ -Servicegrad sind die beiden Integrale bzw. die beiden Summen. Sie werden in der Stochastik unter der Bezeichnung Verlustfunktion 1. Ordnung diskutiert. Formal lautet für eine Zufallsvariable  $X$  mit einer kontinuierlichen Verteilung (Dichte  $\phi_X(z)$ ) die Verlustfunktion 1. Ordnung:

$$\Phi_X^1(s) = \int_x^\infty (z - s) \cdot \phi_X(z) dz.$$

Ist  $X$  eine diskrete Zufallsvariable, so lautet die Verlustfunktion 1. Ordnung:

$$\Phi_X^1(s) = \sum_{x=x_{min}}^{x_{max}} (x - s) \cdot P(X = x),$$

wobei  $x_{min}$  der minimale und  $x_{max}$  der maximale Wert aus dem Wertevorrat von  $X$  ist; gegebenenfalls ist für  $x_{min} = -\infty$  und für  $x_{max} = \infty$  zu verwenden.

Unberücksichtigt blieben bisher die Bestell- und Lagerkosten. Zunächst zu den Bestellkosten. Bezogen auf eine Bestellmenge von  $q$  dauert im Mittel ein Bestellzyklus

$$\frac{q}{E(D)}$$

Perioden. Mit dem variablen Produktionskostensatz  $c$  pro Mengeneinheit betragen die variablen Produktionskostensatz in einem Bestellzyklus im Mittel  $c \cdot q$  Geldeinheiten. Wegen den fixen Rüstkosten von  $K$  Geldeinheiten lauten die Bestellkosten in einem Bestellzyklus im Mittel  $K + c \cdot q$  Geldeinheiten. Damit

lauten die erwarteten Bestellkosten pro Zeiteinheit bzw.

$$\text{Periode } E(BK) = \frac{K + c \cdot q}{q} = (K + c \cdot q) \cdot \frac{E(D)}{q}.$$

In (Herrmann 2009) wurde nachgewiesen, dass die erwarteten Lagerhaltungskosten bei einem Lagerkostensatz  $h$  pro Mengeneinheit und Periode berechnet werden durch:

$$E(LK) = \left( s - E(Y^*) + \frac{q}{2} \right) \cdot h \cdot E(BK) + E(LK)$$

sind die zu minimierenden Gesamtkosten, wobei der  $\beta$ -Servicegrad

$$\frac{E(F_{Y^*}(s))}{q_{opt}} - (1 - \beta(s)) \leq 0.$$

Über die Bildung einer Lagrange-Funktion kann diese Restriktion in die Gesamtkostenfunktion integriert werden, so dass mit Methoden aus der Analysis das Minimierungsproblem gelöst werden kann. Dazu wird diese Restriktion mit einem Lagrange-Multiplikator ( $\kappa$ ) multipliziert und der resultierende Term wird in die oben genannte Kostenfunktion als Summand eingefügt. Allerdings weicht seine Einheit von denen der restlichen Terme ab. Dies ist durch Multiplikation mit  $E(D)$  zu korrigieren, so dass insgesamt die folgende Lagrange-Funktion (Kostenfunktion) entsteht:

$$L(q, s, \kappa) = (K + c \cdot q) \cdot \frac{E(D)}{q} + \left( s - E(Y^*) + \frac{q}{2} \right) \cdot h + \kappa \cdot E(D) \cdot \left( \frac{E(F_{Y^*}(s))}{q} - (1 - \beta) \right).$$

Interessant ist an dieser Stelle der Vergleich zur Kostenfunktion im Optimierungsmodell für die Minimierung der Fehlmengenkosten; s. (Herrmann 2009).

$$\text{Statt dem Term } \kappa \cdot E(D) \cdot \left( \frac{E(F_{Y^*}(s))}{q} - (1 - \beta) \right)$$

enthält sie die expliziten Fehlmengenkosten  $p \cdot \frac{E(D)}{q} E(F_{Y^*}(s))$ .

$$\text{Wegen dieses Zusammenhangs kann der Lagrangefaktor } K \text{ als impliziter Kostensatz (marginale$$



Opportunitätskosten) interpretiert werden. Beim Vorliegen einer kontinuierlichen Verteilungsfunktion für die Zufallsvariable  $Y^*$  führt die Anwendung der Karush-Kuhn-Tucker-Bedingung zur folgenden Lösung des Minimierungsproblems; im Kern wird mit den ersten partiellen Ableitungen der zu minimierenden Gesamtkostenfunktion und der Restriktion ein Gleichungssystem zur Bestimmung der drei Parameter aufgestellt und gelöst.

Formel 1:

$$q_{opt} = \sqrt{\frac{2 \cdot E(D) \cdot (K + \kappa_{opt} \cdot E(F_{Y^*}(s_{opt})))}{h}}$$

Formel 2:

$$\kappa_{opt} = \frac{h \cdot q_{opt}}{E(D) \cdot (P(Y^* \leq s_{opt} + q_{opt}) - P(Y^* \leq s_{opt}))}$$

und

Formel 3:

$$E(F_{Y^*}(s_{opt})) = \int_{s_{opt}}^{\infty} (y^* - s_{opt}) \cdot \varphi_{Y^*}(y^*) dy^* - \int_{s_{opt} + q_{opt}}^{\infty} (y^* - s_{opt} - q_{opt}) \cdot \varphi_{Y^*}(y^*) dy^* = (1 - \beta) \cdot q_{opt}$$

Eine Lösung für  $(q_{opt}, s_{opt}, \kappa_{opt})$  wird iterativ wie folgt bestimmt: Zunächst wird eine Anfangslösung für die Bestellmenge ( $q_1$ ) errechnet, in dem Fehlmengen ausgeschlossen werden; dazu kann implizit  $E(F_{Y^*}(s_{opt})) = 0$  in Formel 1 gesetzt werden. Diese Anfangsbestellmenge wird in Formel 3 eingesetzt, wodurch eine Anfangsfehlmenge und ein Anfangsbestellbestand berechnet werden. Wird dieser Anfangsbestellbestand in Formel 2 eingesetzt, so ergibt sich der Anfangswert für den Lagrange-Multiplikator  $\kappa$ . Das Einsetzen dieses Anfangswerts für den Lagrangefaktor und der Anfangsfehlmenge in Formel 1 liefert die zweite Bestellmenge. Über Formel 3 wird zunächst die zweite Fehlmenge und dann der zweite Bestellbestand berechnet. Formel 2 liefert den zweiten Lagrange-Multiplikator. Dieses Vorgehen wird solange wiederholt, bis zwei aufeinander folgende Bestellmengen und Bestellbestände nahezu identisch sind; ein vergleichbarer Algorithmus ist in (Herrmann 2009) angegeben worden.

Die in der 1. Iteration berechnete Bestellmenge basiert auf einer im Allgemeinen erheblichen Unterschätzung des optimalen Lagrange-Multiplikators  $\kappa_{opt}$  und damit auf einer Überschätzung des optimalen Bestellbestands  $s_{opt}$ , weswegen die zuerst berechnete Bestellmenge niedriger als die optimale Bestellmenge  $q_{opt}$  ist. Für die lagerbedingte Lieferzeit eines Auftrags hat dies zur Folge, dass im Vergleich zur kostenminimalen Situation ein zu geringer Anteil des „Lieferzeit-Risikos“ durch die Höhe der

Bestellmenge absorbiert wird. Denn je höher die Bestellmenge ist, umso länger ist ein durchschnittlicher Bestellzyklus und umso geringer ist der relative Anteil der Länge der Wiederbeschaffungszeit, in der in der Regel eine lagerbedingte Lieferzeit auftreten kann, an der Gesamtlänge eines Bestellzyklus. Dies führt zu einer Unterschätzung der Fehlmenge pro Periode. Wegen eines positiven Lagrange-Multiplikators und eines vorhandenen Fehlbestands steigt die Schätzung der optimalen Bestellmenge in der zweiten Iteration, wodurch die Bestellmenge nun einen höheren Anteil des „Lieferzeit-Risikos“ absorbiert. Dies erhöht die Fehlmenge und bewirkt eine geringere Überschätzung des optimalen Bestellbestands  $s_{opt}$ . Aus der Fortsetzung dieser Argumentation ergibt sich, dass die Bestellmengen monoton ansteigen, die Bestellbestände monoton fallen und die Lagrange-Multiplikatoren monoton ansteigen. Dieser Anstieg der Bestellmengen bei gleichzeitiger Verringerung der Bestellbestände bedeutet, dass ein Teil der Risikoabsicherung vom Sicherheitsbestand auf den durch die Bestellmenge beeinflussten Grundlagerbestand verschoben wird. Da der Bestellbestand nicht beliebig klein wird und die Bestellmenge nicht beliebig hoch wird, konvergieren die Bestellmengen und die Bestellbestände gegen endliche Schranken  $q$  und  $s$ . Ferner lässt sich nachweisen, dass die Folge der Beträge der Differenzen von aufeinanderfolgenden Bestellmengen und Bestellbeständen monoton fallen. Deswegen terminiert der Algorithmus, wenn die Schleife beendet wird, sofern diese beiden Beträge erstmalig kleiner als ein fest vorgegebenes  $\epsilon$  sind, und die Bestellmenge und der Bestellbestand der letzten Iteration habe eine geringe Abweichung zu den Schranken  $q$  und  $s$ .

Damit stellt das Verfahren eine schrittweise Anpassung der vorläufigen Schätzwerte der Bestellmenge und des Bestellbestands (sowie der Fehlmenge und des Lagrange-Multiplikators) an ihre optimalen Werte dar. Das optimale Verhältnis von Bestellbestand und Bestellmenge ist dann erreicht, wenn eine kostenminimale Verteilung des „Lieferzeit-Risikos“ auf die Bestellmenge  $q$  und den Bestellbestand  $s$  vorliegt.

Bei manchen Datenkonstellationen liegt die optimale Lösung bereits nach der ersten Iteration vor. In diesem Fall wird die optimale Bestellmenge durch

$$q_{opt} = \sqrt{\frac{2 \cdot E(D) \cdot K}{h}}, \text{ also } \kappa_{opt} = 0, \text{ berechnet; hierbei}$$

handelt es sich um die optimale Bestellmenge im von Ford W. Harris bereits 1915 vorgestellten klassischen Losgrößenmodell mit konstantem Bedarf, bei dem von einer unendlichen Produktionsgeschwindigkeit, also einem unmittelbaren (i.e. unendlich schnellen) Lagerzugang, ausgegangen wird. Diese Lagerhaltungspolitik wird in der Literatur als  $(s, q)$ -Lagerhaltungspolitik mit periodischer Bestandsüberwachung bezeichnet. Ist zum Zeitpunkt der Auslösung einer Lagerbestellung der disponible Lagerbestand stets genau gleich dem Bestellbestand  $s$ , so ist das Defizit  $U$  stets gleich Null. Diese Eigenschaft eines Bestellbestands liegt genau dann vor, wenn

- (a) der disponible Lagerbestand nach jedem Abgang vom disponiblen Lagerbestand überwacht wird und
- (b) die nachgefragte Menge zwischen zwei aufeinander folgenden Inspektionen des Lagers entweder 0 oder 1 ist.

In diesem Fall wird von einer (s,q)-Lagerhaltungspolitik mit kontinuierlicher Bestandsüberwachung gesprochen. Es sei erinnert, dass diese Lagerhaltungspolitik in der Literatur auch dann angewandt wird, wenn die beiden Bedingungen nicht erfüllt sind.

Demgegenüber beeinflussen sich Bestellmenge und Bestellbestand bei manchen Datenkonstellationen so stark, dass erst nach vielen Iterationen des obigen Algorithmus (nahezu) optimale Werte gefunden werden. Aufgrund der oben vorgestellten Konvergenzanalyse liegt generell eine Unterschätzung der Bestellmenge vor. Daher ist zu erwarten, dass dieser Effekt bei zunehmendem Lagerhaltungskostensatz (h) (und auch bei geringen fixen Bestellkosten) zunehmend deutlicher auftritt. Wie ebenfalls bei der Konvergenzanalyse begründet worden ist, führt ein zu geringer relativer Anteil der Länge der Wiederbeschaffungszeit, in der allein eine lagerbedingte Lieferzeit auftreten kann, an der Gesamtlänge eines Bestellzyklus, zu einer Überschätzung des Bestellbestands. Diese Situation tritt bei einer zunehmenden Lieferzeit zunehmend deutlicher auf. Die variablen Produktions- und Bestellkosten (c) beeinflussen die Berechnung der Bestellmenge (nach Formel 1) und des Bestellbestands (nach Formel 3) nicht, weswegen dieser Parameter diesen Effekt nicht hervorruft. Der Parameter c beeinflusst die zu minimierenden Gesamtkosten durch einen von der Berechnung der Bestellmenge (q) und dem Bestellbestand (s) unabhängigen Term, nämlich E(BK). Deswegen liegt eine zunehmende (und schließlich signifikante) Kostenreduktion durch die Anwendung des Verfahrens bei zunehmender Verringerung dieses Kostensatzes vor.

Ist die Nachfrage in der Wiederbeschaffungszeit einschließlich des Defizits diskret verteilt, so sind in den obigen Berechnungsformeln für die simultane Berechnung von Bestellbestand und Bestellmenge bzw. in dem Algorithmus die Verlustfunktion 1. Ordnung für eine kontinuierliche Verteilung durch die Verlustfunktion 1. Ordnung für eine diskrete Verteilung zu ersetzen und die Verteilungsfunktion ist diskret. Genauso wie bei einer kontinuierlichen Verteilung ist auch hier zu erwarten, dass die Bestellmengen ansteigen und die Bestellbestände abnehmen. Allerdings könnte in der Nähe des Optimums die Bestellmenge die optimale Bestellmenge überschreiten und der Bestellbestand den optimalen Bestellbestand unterschreiten, wodurch eine Verschlechterung auftreten könnte, die in der nächsten Iteration wieder korrigiert werden dürfte. Dies könnte eine Divergenz des Verfahrens zur Folge haben. In diesem Fall böte sich ein Abbruch des Verfahrens mit einer voraussichtlich suboptimalen Lösung an.

### Berücksichtigung des Defizits

Für die Anwendung der Berechnungsformeln ist die Kenntnis der Verteilung  $Y^*$  zu der Summe aus der Nachfrage in der Wiederbeschaffungszeit (y) und dem Defizit (u) (gegenüber dem Bestellbestand) zu Beginn der Wiederbeschaffungszeit, also der Gesamtnachfrage  $y^* = y + u$  in der Wiederbeschaffungszeit, erforderlich.

Im Folgenden wird sie auf die Verteilung zu den Periodennachfragemengen (D) und bei zufälligen Lieferzeiten auch auf deren Verteilung (L) zurückgeführt. In beiden Fällen handelt es sich bei der Verteilung der Nachfrage in der Wiederbeschaffungszeit (Y) um eine Summe von Zufallszahlen. Bei stochastische

Wiederbeschaffungszeiten lautet sie:  $Y = \sum_{i=1}^L D_i$ .

$E(Y) = E(L) \cdot E(D)$  ist ihr Erwartungswert und  $Var(Y) = E(L) \cdot Var(D) + Var(L) \cdot (E(D))^2$  ist

ihre Varianz (der Nachweis ist in auf [Herr09a] angegeben). Bei einer konstanten Lieferzeit ergeben sich der Erwartungswert und die Varianz dadurch, dass in den Formeln  $Var(L) = 0$  gesetzt wird (dies führt zu den in [Herr09a] genannten Formeln). Für das Defizit (u) können keine Aufzeichnungen existieren, da diese erst nach dem Implementieren einer (s,q)-Lagerhaltungspolitik gemessen werden können. Allerdings werden die stochastischen Eigenschaften der Zufallsvariable zum Defizit (U) maßgeblich durch die Verteilung zu den Periodennachfragemengen beeinflusst. Nach der Erneuerungstheorie lässt sich die Verteilungsfunktion zu U durch die folgende Formel approximieren, sofern die Bestellmenge im Vergleich zur mittleren Periodennachfragemenge ( $E(D)$ ) groß ist. Im

kontinuierlichen Fall gilt  $\Phi_U(u) \approx \frac{1}{E(D)} \cdot \int_0^u [1 - \Phi_D(d)] dd$

für  $u \geq 0$  und im diskreten Fall ist

$$P(U = u) \approx \frac{1 - P(D \leq u)}{E(D)} = \frac{\sum_{d=u+1}^{\infty} P(D = d)}{E(D)} \quad \text{die}$$

Wahrscheinlichkeit (s. [Karl58]). Es darf davon ausgegangen werden, dass das Defizit U und die Periodennachfragemenge D stochastisch unabhängige Zufallsvariablen sind. Ebenfalls mit der Erneuerungstheorie ist der Erwartungswert des Defizits

$$E(U) = \frac{(E(D))^2 + Var(D)}{2 \cdot E(D)}$$

und die Varianz des Defizits berechnet sich durch

$$\text{Var}(U) = \frac{E((D - E(D))^3)}{3 \cdot E(D)} + \frac{1}{2} \text{Var}(D) \cdot \left[ 1 - \frac{\text{Var}(D)}{2 \cdot E(D)^2} \right] + \frac{1}{12} \cdot E(D)^2.$$

Wegen

$$Y^* = Y + U \text{ ist } E(Y^*) = E(Y) + E(U) \text{ und}$$

$$\text{Var}(Y^*) = \text{Var}(Y) + \text{Var}(U).$$

Wie im Leitfaden noch näher begründet werden wird, tritt in der industriellen Praxis häufig eine Normalverteilung für die Periodennachfragemengen  $D$  auf und die Lieferzeit ist entweder konstant oder ebenfalls normalverteilt. Dann ist die Nachfrage in der Wiederbeschaffungszeit normalverteilt und es darf unterstellt werden, dass die Gesamtnachfrage in der Wiederbeschaffungszeit ebenfalls einer Normalverteilung folgt. Mit den oben genannten Formeln werden ihre charakterisierende Größen, nämlich ihr Erwartungswert und ihre Varianz berechnet. Statt einer Normalverteilung tritt auch noch eine Gammaverteilung auf. Hierfür gilt die gleiche Aussage. Aus ihrem Erwartungswert und ihrer Varianz lassen sich ihre charakterisierenden Parameter, nämlich der Skalenparameter  $k$  und der Formparameter  $\alpha$ ,

$$\text{durch } k = \frac{E(Y^*)}{\sigma^2(Y^*)} \text{ und } \alpha = \frac{E^2(Y^*)}{\sigma^2(Y^*)} \text{ berechnen.}$$

### 3. LEITFADEN ZUR BESTIMMUNG DER STEUERUNGSPARAMETER ZUR LAGERHALTUNG

#### Erfüllung der Verfahrensparameter

Für die Anwendung der Lagerhaltungsverfahren in der industriellen Praxis ist ein langer Planungshorizont ( $T$ ), der als unendlich lang angesehen werden kann, zu unterstellen und die in diesem Planungshorizont anfallenden Periodenbedarfe und bei zufälligen Wiederbeschaffungszeiten auch diese bilden einen quasi-stationären stochastischen Prozess, d.h. sie lassen sich durch stochastisch unabhängige und identisch verteilte Zufallsvariablen mit einem endlichen Mittelwert und einer endlichen Streuung beschreiben.

Eine gravierende Einschränkung für viele industrierelevante Probleme ist die Forderung nach stochastischer Unabhängigkeit. Allerdings kann auf so genannte dynamische Sicherheitsäquivalente, aus der stochastischen dynamischen Optimierung, zurückgegriffen werden, mit denen der stochastische Prozess durch gewisse als Prognosen deutbare bedingte Erwartungswerte beschrieben werden kann. Daher wird vorgeschlagen, das Vorliegen von stochastischer Unabhängigkeit anzunehmen.

Die restliche Bedingung impliziert, dass die jüngsten Vergangenheitswerte zu den Periodenbedarfen und den Wiederbeschaffungszeiten über einen langen Zeitraum bekannt sind, und diese müssen einen im Zeitablauf konstanten Mittelwert und eine im Zeitablauf konstante

Streuung besitzen. Für den Leitfaden wird vorgeschlagen, beim Erfüllen dieser Bedingung die eigentliche restliche Bedingung als erfüllt anzusehen. Produkte mit einem ausgeprägten Lebenszyklus haben keinen Bedarfsverlauf mit einem im Zeitablauf konstanten Mittelwert und einer im Zeitablauf konstanten Standardabweichung. Diese Bedingung wird in der Regel auch in der Anlaufphase und am Ende von einem Lebenszyklus eines Produkts verletzt. Auf solche Produkte sind die oben erläuterten stochastischen Lagerhaltungspolitiken nicht anwendbar und werden deswegen von der weiteren Betrachtung ausgeschlossen.

#### Kostensätze

Für die Anwendung der einzelnen Varianten der (s,q)-Lagerhaltungspolitik werden für jedes Produkt seine konkreten Kostensätze, die Verteilung seiner Nachfrage in der Wiederbeschaffungszeit einschließlich des Defizits und sein Servicegrad benötigt.

Bei den Kostensätzen sind die fixen Rüstkosten und der Lagerkostensatz anzugeben. Nach den Überlegungen in (Gudehus 2003) sind die fixen Rüstkosten bestimmt durch Bestellmengenkosten ( $k_{NAuf}$ ), spezifische Transportkosten ( $k_{TrLE}$ ), spezifische Einlagerkosten ( $k_{Lein}$ ) und das Fassungsvermögen der Ladeeinheiten ( $C_{LE}$ ) über die Formel  $(k_{NAuf} + (k_{TrLE} + k_{Lein}) \cdot (C_{LE} - 1) / 2 \cdot C_{LE})$ . Der Lagerkostensatz ist bestimmt durch Beschaffungspreis/Verbrauchseinheit ( $P_{VE}$ ), den Lagerzinssatz ( $z_L$ ), der sich aus einem Kapazitätszinssatz und einem Risikozinssatz zusammensetzt, den Lagerordnungsfaktor ( $f_{LO}$ ), den spezifischen Lagerplatzkosten ( $k_{LPI}$ ) sowie das Fassungsvermögen der Ladeeinheiten ( $C_{LE}$ ) über die Formel

$$\left( P_{VE} \cdot z_L + 2 \cdot f_{LO} \cdot \frac{k_{LPI}}{C_{LE}} \right).$$

#### Vorliegende Verteilungen

Für die Anwendung der einzelnen Varianten der (s,q)-Lagerhaltungspolitik ist eine konkrete Verteilung notwendig. Im Folgenden wird zunächst untersucht, welche Verteilungen auftreten. Generell handelt es sich bei den Periodenbedarfen, den Lieferzeiten, den Nachfragen in den Wiederbeschaffungszeiten und den Defiziten um Zeitreihen. Die zu einer Zeitreihe gehörende diskrete Verteilung ist das Histogramm zu dieser Zeitreihe. In vielen Fällen handelt es sich sogar um eine kontinuierliche Verteilung; dies ist hilfreich, da dann die Verlustfunktion 1. Ordnung oftmals geschlossen gelöst werden kann. Nach der schließenden Statistik kann mit dem Kolmogorov-Smirnov Test überprüft werden, ob eine vorgegebene (kontinuierliche) Verteilung  $\Phi$  vorliegt. Es sei daran erinnert, dass für den Kolmogorov-Smirnov Test auch die charakterisierenden Parameter von  $\Phi$  bekannt sein müssen (beispielsweise sind dies bei einer Normalverteilung der Erwartungswert und die Streuung), die über die vorliegenden Vergangenheitswerte berechnet werden.

Nach den Überlegungen über Prognoseverfahren (s. z.B. Herrmann 2009) lässt sich eine konkret vorliegende Zeitreihe in die dort genannten Typen von Zeitreihenverläufen einordnen. Zu diesen Zeitreihenverläufen existieren sehr gute Prognoseverfahren mit bekannten oder zu erwartenden theoretischen Verteilungen ihrer Prognosewerte. Häufig handelt es sich sogar um kontinuierliche Verteilungen, die für den Kolmogorov-Smirnov Test verwendet werden sollten. Da ein Zeitreihenverlauf quasi stationär ist, sind nach den in der Literatur üblichen Typen von stationären Bedarfsverläufen, wie sie beispielsweise in (Herrmann 2009) angegeben sind, der regelmäßige, der saisonale und der unregelmäßige Zeitreihenverlauf zu betrachten. Aufgrund der Anfangsbuchstaben wird diese Klassifizierung von Zeitreihenverläufen als RSU-Klassifizierung bezeichnet; ihre konkreten Ausprägungen werden in den nun folgenden Überlegungen präzisiert.

Zunächst wird der Fall eines regelmäßigen Zeitreihenverlaufs betrachtet. Ein konstanter Zeitreihenverlauf lässt sich durch den (n-periodischen) gleitenden Durchschnitt prognostizieren. Wie in (Herrmann 2009) ausführlich begründet worden ist, folgen die Prognosewerte einer Normalverteilung, sofern n hinreichend hoch ist; es sei betont, dass wegen der Annahme eines unendlich langen Betrachtungshorizonts ein solches n gebildet werden kann. Deswegen darf eine Normalverteilung für die Prognosewerte unterstellt werden. Für das alternative Verfahren, die exponentielle Glättung erster Ordnung, gilt dies nach (Herrmann 2009) nicht; allerdings existiert ein asymptotischer Zusammenhang zwischen beiden Verfahren, so dass auch in diesem Fall eine Normalverteilung für die Prognosewerte unterstellt werden darf.

Deswegen wird im Leitfaden getestet, ob die konkrete Zeitreihe einer Normalverteilung folgt. Einzige Ausnahme ist das Vorliegen einer relativ zum Erwartungswert hohen Streuung, da dann negative Periodennachfragen oder Wiederbeschaffungszeiten in der Approximation berücksichtigt werden. Da negative Periodennachfragen oder Wiederbeschaffungszeiten tatsächlich nicht auftreten, weicht die Approximation von der tatsächlichen Zeitreihe ab, und deswegen sollte eine solche Approximation nicht verwendet werden. Anderenfalls werden Steuerungsgrößen berechnet, die zu einer Überschätzung des Servicegrads führen. Ausgeschlossen werden nicht vernachlässigbare Wahrscheinlichkeiten für negative Werte bei der Verwendung einer Gammaverteilung, so dass beim Auftreten einer hohen Streuung auf das Vorliegen einer Gammaverteilung getestet werden sollte.

Liegt zusätzlich ein saisonaler Einfluss vor, so bietet sich eine Prognose durch die multiple lineare Regressionsrechnung an. Nach (Herrmann 2009) folgen seine Prognosefehler einer Normalverteilung. Daher ist es plausibel anzunehmen, dass auch die Prognosewerte einer Normalverteilung folgen. Deswegen soll auf das Vorliegen einer Normalverteilung getestet werden. Da

auch hier starke Streuungen auftreten können, wird zusätzlich getestet, ob eine Gammaverteilung vorliegt.

Nun wird das Vorliegen eines unregelmäßigen Zeitreihenverlaufs betrachtet. Ist der Zeitreihenverlauf nicht sporadisch, aber stark schwankend, so darf nach den oben vorgestellten Überlegungen von einer Gammaverteilung ausgegangen werden. Folglich wird im Leitfaden die Zeitreihe auf das Vorliegen einer Gammaverteilung getestet.

Nun werden die beiden anderen Varianten eines unregelmäßigen Zeitreihenverlaufs, nämlich das Vorliegen von einem sporadischen Zeitreihenverlauf, der stark schwankend sein darf, behandelt. Nach (Schneider 1979) wird sie am ehesten durch eine Gammaverteilung approximiert, so dass auf ihr Vorliegen getestet wird.

Somit treten als kontinuierliche Verteilungen nur die Normalverteilung und die Gammaverteilung auf. Es sei betont, dass aufgrund des zentralen Grenzwertsatzes aus der Stochastik viele Zeitreihen einer Normalverteilung genügen; die Analyse zu Prognosewerten in (Herrmann 2009) basiert auf dem zentralen Grenzwertsatz. Für diese und eine diskrete Verteilung ist die Verlustfunktion 1. Ordnung ( $\Phi^1$ ) zu berechnen. Für eine Normalverteilung lässt sie sich auf diejenige zu der Standardnormalverteilung zurückführen und lautet

$$\Phi^1_{\mathcal{N}(\mu, \sigma^2)}(s) = \sigma(Y^*) \cdot \Phi^1_{\mathcal{N}(0,1)}\left(\frac{s-\mu}{\sigma}\right),$$

mit dem Mittelwert  $\mu$  und der Streuung  $\sigma$  (s. z.B. [Herr09a]). Mit dem Mittelwert  $\mu$  und Streuung  $\sigma$  lautet im Fall einer Gammaverteilung die Verlustfunktion 1. Ordnung

$$G^1(s) = \frac{k}{\alpha} - s - \frac{k}{\alpha} \cdot I(k+1, s \cdot \alpha) + s \cdot I(k, s \cdot \alpha)$$

mit dem

Skalenparameter  $k = \frac{\mu}{\sigma^2}$  und dem Formparameter

$$\alpha = \frac{\mu^2}{\sigma^2}$$

(s. z.B. (Tempelmeier 2005)); dabei ist

$I(k, x \cdot \alpha)$  die so genannte „incomplete gamma function“, die die Wahrscheinlichkeit dafür angibt, dass die zugehörige gammaverteilte Zufallsvariable kleiner oder gleich  $x$  ist. Im Fall einer diskreten Verteilung liegt eine ggf. zufällige Summe von Zufallsvariablen vor, die durch die mehrfache Anwendung der Faltungsformel auf eine Summe über Periodenbedarfe zurückgeführt werden kann (s. z.B. (Tempelmeier 2005)).

### Grundstruktur des Vorgehens bei einer konkreten Unternehmenssituation

In einer konkreten Unternehmenssituation wird aus den jüngsten Vergangenheitswerten zu den Periodenbedarfen und den Wiederbeschaffungszeiten über einen langen Zeitraum jeweils ein Histogramm erstellt, das die jeweilige empirisch gefundene Verteilung in Form einer diskreten Verteilung beschreibt. Durch den Kolmogorov-Smirnov Test auf eine Normalverteilung und eine Gammaverteilung kann eventuell statt der diskreten Verteilung eine der beiden kontinuierlichen Verteilungen

verwendet werden. Mit den oben vorgestellten Formeln wird aus diesen die Verteilung der Gesamtnachfrage in der Wiederbeschaffungszeit berechnet. Wird für jedes Produkt ein Servicegrad festgelegt, so liegen alle Informationen zur Anwendung eines der Varianten der (s,q)-Lagerhaltungspolitik vor. Selbst für die aufwendigste Variante der simultanen Berechnung von der Bestellmenge und dem Bestellbestand sowie bei der Verwendung von einer diskreten Verteilung benötigt ein Computer nur eine geringe Rechenzeit. Da diese Steuerungsparameter einmal eingestellt werden, nämlich für einen relativ langen Planungszeitraum T, könnte das optimale Verfahren für jedes Produkt angewendet werden.

Die Lösungsgüte ist jedoch bestimmt, wie gut die Kostensätze und die Verteilungen (letztlich die der Gesamtnachfrage in der Wiederbeschaffungszeit) die industrielle Realität im Planungszeitraum T abbildet. Seitens der Kostensätze beeinflussen der Rüst- und der Lagerhaltungskostensatz die optimale Bestellmenge. Im klassischen Losgrößenmodell mit konstantem Bedarf sind die Kosten relativ unempfindlich gegenüber Abweichungen von der optimalen Bestellmenge (siehe hierzu z.B. (Herrmann 2009)), weswegen Ungenauigkeiten im Rüst- oder Lagerhaltungskostensatz vergleichsweise geringe Auswirkungen (eben auf die Kosten) haben. Sofern eine Abweichung von der optimalen Bestellmenge nur einen geringen Einfluss auf den optimalen Bestellbestand hat, unterscheiden sich die Kostenfunktion in diesem klassischen Modell und  $E(BK) + E(LK)$  nur durch eine Konstante. Dann gilt die obige Sensitivitätsaussage (für das klassische Modell) selbst bei der simultanen Berechnung von Bestellmenge und Bestellbestand; in jedem Fall gilt sie bei der Berechnung der optimalen Bestellmenge wie bei diesem klassischen Modell. Wie bei der Analyse bei der simultanen Berechnung von Bestellmenge und Bestellbestand begründet worden ist, haben variable Produktions- und Bestellkosten einen deutlichen Einfluss auf die Gesamtkosten. Starken Einfluss auf die Lösungsgüte hat die Genauigkeit der Verteilung der Gesamtnachfrage in der Wiederbeschaffungszeit; s. hierzu z.B. (Herrmann 2009) oder (Tempelmeier 2005). Dies betrifft sowohl die Genauigkeit der Vergangenheitswerte als auch die Prozeßgenauigkeit, also die tatsächliche Einhaltung der Verteilung im Planungszeitraum T durch die (möglichst effiziente und effektive) Bestandsüberwachung und die Bestellabwicklung. Ersteres bedeutet einen hohen Aufwand für die Erhebung der richtigen Daten bzw. eventuell deren Korrektur und letzteres bedeutet einen hohen Aufwand für eine hohe Prozessqualität. Insgesamt werden je nach Lösungsgüte unterschiedlich hohe und zum Teil beträchtliche Kosten für das Sammeln von benötigten Daten, das Durchführen von Berechnungen, das Ausstellen von Aktivitätenberichten usw. benötigt, die als Kontrollkosten für das Lagerhaltungssystem bezeichnet werden. Höhere Kosten für genauere Daten sind nur dann wirtschaftlich, wenn diese zu höheren Einsparungen bei den Kosten der Lagerhaltung führen.

Nach den obigen Überlegungen wird eine Verringerung der durch die Datenqualität verursachten Kontrollkosten durch eine einfache Erhebung der zu verwendenden Verteilung und einer einfachen Kontrolle auf ihre Einhaltung bewirkt. Nach den obigen Überlegungen zu den zu erwartenden kontinuierlichen Verteilungen zu Zeitreihen darf für die Periodenbedarfe und bei zufälligen Wiederbeschaffungszeiten auch für diese jeweils eine Normal- bzw. eine Gammaverteilung unterstellt werden. Bei einer Normalverteilung werden ihre charakterisierenden Größen, nämlich der Erwartungswert und die Streuung (s. z.B. [Hüb03]), durch den Mittelwert und die Streuung der Vergangenheitswerte bestimmt. Mit diesen lassen sich auch die charakterisierenden Größen einer Gammaverteilung, nämlich der Skalenparameter und der Formparameter (s. z.B. [Hüb03]), berechnen. Es bietet sich an, mit dem Kolmogorov-Smirnov Test das Vorliegen einer Normal- bzw. Gammaverteilung zu überprüfen. Bei einem negativen Ergebnis sollte überprüft werden, ob diese Verteilung die tatsächliche zukünftige Entwicklung der Periodenbedarfe und ggf. der Wiederbeschaffungszeiten besser als die Vergangenheitswerte beschreibt oder einfach einen besseren Prozess darstellt, der einzuhalten ist. Im dann positiven Fall sollte dennoch mit dieser Verteilung weitergearbeitet werden.

Bei einer einfachen Datenerhebung können, eventuell nur vermutete, Ungenauigkeiten in den Vergangenheitswerten, vor allem im Hinblick auf die tatsächliche zukünftige Entwicklung der Periodenbedarfe und ggf. der Wiederbeschaffungszeiten, durch eine Approximation behoben werden. Dies ermöglicht wiederum die Vorgabe eines besseren Prozesses. Bei diesem Vorgehen wird mit den durch eine Approximation erzeugten Werten weitergearbeitet; also statt der Beobachtungswerte. Die oben genannte Forderung nach quasi Stationarität gilt dann für diese Werte. Für eine Approximation bietet sich eine der oben vorgestellten Prognoseverfahren an, da, wie oben begründet wurde, ihre Prognosewerte häufig einer Normal- oder Gammaverteilung folgen.

Obwohl, wie oben bereits ausgeführt worden ist, die Rechenzeiten für die simultane Berechnung von Bestellmenge und Bestellbestand bei einer (s,q)-Lagerhaltungspolitik moderat sind, sind unwirtschaftliche Rechenzeiten zu vermeiden. Sie lassen sich durch eine grobe Berechnung von Bestellmenge und Bestellbestand, eine Möglichkeit hierzu wird weiter unten vorgestellt, und durch eine Beschaffung nur im Bedarfsfall sehr deutlich reduzieren. Da ein Computer auch für andere Aufgaben eingesetzt werden kann, haben reduzierte Kontrollkosten eventuell sogar deutliche positive Auswirkungen.

Damit lassen sich insgesamt die Kontrollkosten variieren und auch abschätzen. Verantwortlich für die jährlichen Gesamtkosten der Lagerhaltung sind nach dem oben vorgestellten Optimierungsmodell die jährlichen Kosten für die Wiederbeschaffung, die Lagerhaltung und die Fehlmengen. Die Forderung nach einem Kostensatz für die Fehlmengen mag überraschen, da bei der Lösung

stochastischer Lagerhaltungsprobleme ein vorgegebener Servicegrad einzuhalten ist und nicht ein Optimierungsproblem, bei dem Fehlmengen durch eine Kostenfunktion bewertet werden, gelöst wird. In diesem Zusammenhang sei auch an die Schwierigkeit erinnert, einen Fehlmengenkostensatz genau erheben zu können. Die Wichtigkeit eines Produkts, beispielsweise als Komponente für ein Endprodukt (oder auch für das Produktspektrum, z.B. für das Prestige), ist bestimmt durch die Auswirkung einer Fehlmenge. Ohne eine Information über eine Fehlmenge kann daher das Einsparungspotential durch ein aufwendigeres Bestandsmanagement nicht bewertet werden. Da sicher nicht alle Produkte unterschiedliche Lagerhaltungspolitiken haben werden, bietet sich eine ABC-Klassifizierung an.

### **Behandlung von A-Teilen**

Bezogen auf eine solche ABC-Klassifizierung wird vorgeschlagen, für jedes A-Teil anzunehmen, dass selbst kleine prozentuale Einsparungen in den jährlichen Gesamtkosten der Lagerhaltung wirtschaftlich sind. Folglich ist ein hoher Aufwand für die Genauigkeit der Daten und die Berechnung von Bestellmenge und Bestellbestand gerechtfertigt. Deswegen werden bei A-Teilen die vorliegenden Daten systematisch und soweit wie möglich vollständig auf Korrektheit überprüft und ggf. korrigiert. Mit einem Prognoseverfahren können sinnvolle Glättungen der Daten vorgeschlagen werden, weswegen Prognoseverfahren in diesem Sinne an dieser Stelle im Leitfaden eingesetzt werden. Es wird angenommen, dass die so gefundene empirische Verteilung der Periodennachfragemengen und ggf. die der Wiederbeschaffungszeiten die Realität im Unternehmen, vor allem was die tatsächliche zukünftige Entwicklung angeht, exakt beschreiben. Mit diesen Verteilungen wird nach den oben angegebenen Formeln die Verteilung der Gesamtnachfrage in der Wiederbeschaffungszeit ermittelt; sie beschreibt folglich auch die Realität im Unternehmen exakt. Dann wird mit dieser diskreten Verteilung oder einer sehr guten Approximation durch eine kontinuierliche Verteilung eine simultane Berechnung von Bestellmenge und Bestellbestand durchgeführt. Dies entspricht dem Vorgehen in der Literatur, s. beispielsweise (Silver et al. 1998).

### **Behandlung von B-Teilen**

Als B-Teile sollten solche Produkte ausgewählt werden, für die eine periodische Bestandsüberwachung mit einer Normalverteilung (oder bei einer hohen Streuungen mit einer Gammaverteilung) der Gesamtnachfragemenge in der Wiederbeschaffungszeit zu dem besten Kompromiss zwischen moderaten Kontrollkosten für das Lagerhaltungssystem und den jährlichen Gesamtkosten der Lagerhaltung führt. Damit sind die Einsparungen bei den Lagerhaltungskosten durch eine simultane Berechnung von Bestellmengen und Bestellbestand geringer als die dafür notwendigen höheren Aufwände in den Kontrollkosten, und die Einsparungen bei den Kontrollkosten durch eine einfachere Berechnung der

Steuerungsparameter sind geringer als die dadurch verursachten höheren Gesamtlagerhaltungskosten. Eine Normalverteilung dürfte die Prozessrealität oftmals nur ungenau abbilden. Folglich erscheint ihre genaue Erhebung nicht erforderlich zu sein. Deswegen wird vorgeschlagen, die Daten nur mit einem moderaten Aufwand zu überprüfen und ggf. mit einfachen Maßnahmen zu bereinigen. Für die Periodennachfragemengen und ggf. die der Wiederbeschaffungszeiten wird jeweils eine Normalverteilung (oder bei hohen Streuungen eine Gammaverteilung) als hinreichend genau angesehen. Bestätigt die Anwendung des Kolmogorov-Smirnov Tests auf die so erhobenen Vergangenheitsdaten dies nicht, so wird mit einem der oben angesprochenen Prognoseverfahren versucht, die Daten so zu korrigieren, dass sie einer dieser beiden Verteilungen genügt. Je geringer die jährlichen Gesamtkosten der Lagerhaltung für ein B-Teil sind, desto ungenauer darf eine Normalverteilung (oder bei hohen Streuungen eine Gammaverteilung) die Prozessrealität abbilden. Ist dies bei höherwertigen B-Teilen nicht möglich, so wird versucht, dieses Produkt als A-Teil einzustufen. Bei niederwertigen B-Teilen wird durch ihre Einstufung als C-Teil, wie weiter unten noch erläutert werden wird, das Verwenden einer diskreten Verteilung vermieden. Deswegen wird mit einer diskreten Verteilung nur in Ausnahmefällen gerechnet, aber ihr Auftreten kann nicht gänzlich vermieden werden. Liegen starke gegenseitige Abhängigkeiten zwischen dem Bestellbestand und der Bestellmenge mit beträchtlichen Auswirkungen auf die Gesamtkosten vor, so wird eine simultane Berechnung von Bestellbestand und Bestellmenge durchgeführt. Kriterien hierzu ergeben sich aus der Analyse der simultanen Berechnung von Bestellbestand und Bestellmenge im Abschnitt „Lagerhaltungsverfahren“. Demgegenüber schlagen Wissenschaftler wie Silver, Pyke und Peterson (s. (Silver et al. 1998) für B-Teile eine noch weitergehende Verfahrensvereinfachung vor, nach der eine kontinuierliche Bestandsüberwachung mit der Berechnung der Bestellmenge über die klassische Losgrößenformel sowie des Bestellbestands im Kern über die Verlustfunktion 1. Ordnung und zwar möglichst mit einer Normalverteilung zur Beschreibung der Nachfrage in der Wiederbeschaffungszeit implementiert werden sollte.

### **Behandlung von C-Teilen**

Bezogen auf die ABC-Klassifizierung wird vorgeschlagen, für jedes C-Teil anzunehmen, dass nur eine marginale Einsparung in den jährlichen Gesamtkosten der Lagerhaltung durch eine optimale Lagerhaltungspolitik gegenüber einer sehr einfachen Lagerhaltung erzielt werden kann. Deswegen werden die in einem Unternehmen aufgezeichneten Periodenbedarfe und Wiederbeschaffungszeiten weitgehend ungeprüft übernommen. Verfahrensseitig wird eine kontinuierliche Bestandsüberwachung vorgeschlagen. Dabei wird die Nachfrage in der Lieferzeit grundsätzlich durch eine Normalverteilung approximiert. Nach dem zentralen

Grenzwertsatz aus der Stochastik (s. hierzu beispielsweise [Hüb03]) bewirkt die Zusammenfassung hinreichend vieler Periodenbedarfe eine Normalverteilung mit einer so kleinen Streuung, dass keine negativen Bedarfe auftreten. Dies führt zu einer produktspezifischen normalverteilten Zufallsvariable  $D$ . In vielen Unternehmen ist bereits ein Übergang von täglichen zu jährlichen Bedarfen ausreichend. Gegenüber der Anzahl an zusammengefassten Perioden sind die Lieferzeiten bei C-Teilen klein und werden daher durch die höchste auftretende Lieferzeit  $l$  abgeschätzt; es sei angemerkt, dass Ausreißer in der Lieferzeit von der weiteren Betrachtung ausgeschlossen werden. Diese Abschätzung nach oben führt zwar zu einem höheren Bestellbestand und einem höheren mittleren Bestand, der wegen des geringen Werts von C-Teilen jedoch unproblematisch ist. Wegen des geringen Werts von C-Teilen sind hohe Bestände möglich, weswegen vorgeschlagen wird, nur wenige Lieferzeitpunkte in Form von hohen mittleren Bestellzyklen  $\tau$  zu zulassen. Wie oben erläutert worden ist, bestimmt dieser über die Formel  $q = \tau \cdot E(D)$  eine mittlere Bestellmenge, die als (produktspezifische) optimale Bestellmengen verwendet wird. Dadurch reduzieren sich die Kontrollkosten für das Lagerhaltungssystem. Es sei erwähnt, dass C-Teile häufig nur geringe jährliche Bedarfe besitzen, es sich also um Langsamdreher handelt. In vielen Unternehmen sind Bestellzyklen von 6, 12 und 18 Monaten günstig.

### Festlegung der Servicegrade

Wegen der unterschiedlichen Bedeutung der A-, B- und C-Teile sollte A-Teilen ein höherer Servicegrad als B-Teilen und diesen ihrerseits ein höherer Servicegrad als C-Teilen zugeordnet werden. Da für einen vorgegebenen Servicegrad der erforderliche Sicherheitsbestand bei einem regelmäßigen Bedarfsverlauf mit geringer Streuung sicher kleiner als bei einem stark schwankenden sporadischen Bedarfsverlauf ist, bietet es sich an, die RSU-Klassifizierung bei der Festlegung des Servicegrads mit zu berücksichtigen. Deswegen werden Servicegrade im Leitfaden für jede Kombination der beiden Klassifizierungsarten angegeben, also für neun Produktgruppen. Damit berücksichtigt der Leitfaden sowohl die wertmäßige Bedeutung der Produkte als auch die Rechenaufwände für die Verfahren als auch die Kosten für die Lagerhaltung der Produkte.

### Überführung dieser Schritte in einen Leitfaden

Der Leitfaden besteht nun aus den folgenden Schritten, die in **Fehler! Verweisquelle konnte nicht gefunden werden.** zusammengefasst sind. Es sei betont, dass es sich dabei um ein Verfahrensrahmen handelt, der an verschiedenen Stellen unternehmensspezifisch zu präzisieren ist. So ist beispielsweise unternehmensspezifisch festzulegen, was unter Genauigkeit der zukünftigen Entwicklung der Periodenbedarfe verstanden werden soll. Zunächst werden im ersten Schritte die Kostensätze für die Berechnung der jährlichen Gesamtkosten der Lagerhaltung erhoben. Im zweiten Schritt wird überprüft, welche Produkte die für

A-, B- und C-Teile genannten Kriterien für einen wirtschaftlichen Aufwand für eine Lagerhaltung erfüllen. Aufgrund der Arbeitsweise einer optimalen Lagerhaltung wird hier vorgeschlagen, als Maß für einen wirtschaftlichen Aufwand für eine Lagerhaltung die jährlichen Kosten für die Wiederbeschaffung, den Transport und die Fehlmengen zu verwenden. Basierend auf die oben festgelegte Genauigkeit der Daten für A-, B- und C-Teile werden in Schritt 3 die Kostensätze und in Schritt 4 die Periodennachfragemengen und Wiederbeschaffungszeiten erhoben. Mit dem Kolmogorov-Smirnov Test wird im Schritt 5 bei A- und B-Teilen überprüft, ob die Periodennachfragemengen und ggf. die Wiederbeschaffungszeiten einer Normalverteilung folgen. Bei einer hohen Streuung wird auf eine Gammaverteilung getestet. Im negativen Fall wird für B-Teile mit einem Prognoseverfahren versucht, eine normalverteilte, und bei einer hohen Streuung eine gammaverteilte Zeitreihe zu erreichen. In Schritt 6 wird zunächst die produktspezifische Nachfrage in der Wiederbeschaffungszeit einschließlich des Defizits ( $Y^*$ ) für alle A- und B-Teile aus der Verteilung der Periodennachfrage, der Verteilung der Wiederbeschaffungszeit und der durch die Periodennachfrage bestimmten Verteilung für das Defizit nach den oben angegebenen Berechnungsvorschriften ermittelt; der Ausschluss von C-Teilen ist im Abschnitt über die Verfahren zu C-Teilen begründet worden. Da es sich bei der Nachfrage in der Wiederbeschaffungszeit ( $Y$ ) um eine Summenbildung von Verteilungen handelt, ist es aufgrund des zentralen Grenzwertsatzes (der Stochastik) möglich, sogar realistisch, dass  $Y$  einer Normalverteilung bzw. bei großen Schwankungen in den Wiederbeschaffungszeiten einer Gammaverteilung genügt, obwohl der Periodenbedarf weder normal- noch gammaverteilt ist. Liegt nach Schritt 6 für  $Y^*$  nur eine diskrete Verteilung vor, so wird im Schritt 7 das im Schritt 5 angegebene Vorgehen auf  $Y^*$  angewendet. Die oben angegebene Einteilung von Zeitreihenverläufen wird in Schritt 8 auf die in Schritt 6 berechnete Gesamtnachfrage in der Wiederbeschaffungszeit angewendet; wurde bereits eine kontinuierliche Verteilung berechnet, so wird der durch diese Verteilung bestimmte Zeitreihenverlauf verwendet. Nach den obigen Überlegungen werden im Schritt 9 Servicegrade für jede Kombination der beiden Klassifizierungsarten festgelegt. Mit den zu den A-, B- und C-Teilen genannten einzelnen Varianten der (s,q)-Lagerhaltungspolitik werden für jedes Produkt mit seinen Kostensätzen, seiner Verteilung der Nachfrage in der Wiederbeschaffungszeit einschließlich des Defizits und seinem Servicegrad die beiden Steuerungsparameter, Bestellmenge  $q$  und Bestellbestand  $s$ , errechnet.

Die detaillierte Beschreibung der Einzelschritte des Leitfadens enthält bereits Hinweise, wie dieser durch ein Softwaresystem weitgehend automatisch durchgeführt werden kann. So sind zu den Schritten 6 und 10 die genannten Formeln zu programmieren. Aus den Stammdaten eines ERP- bzw. PPS-Systems können die

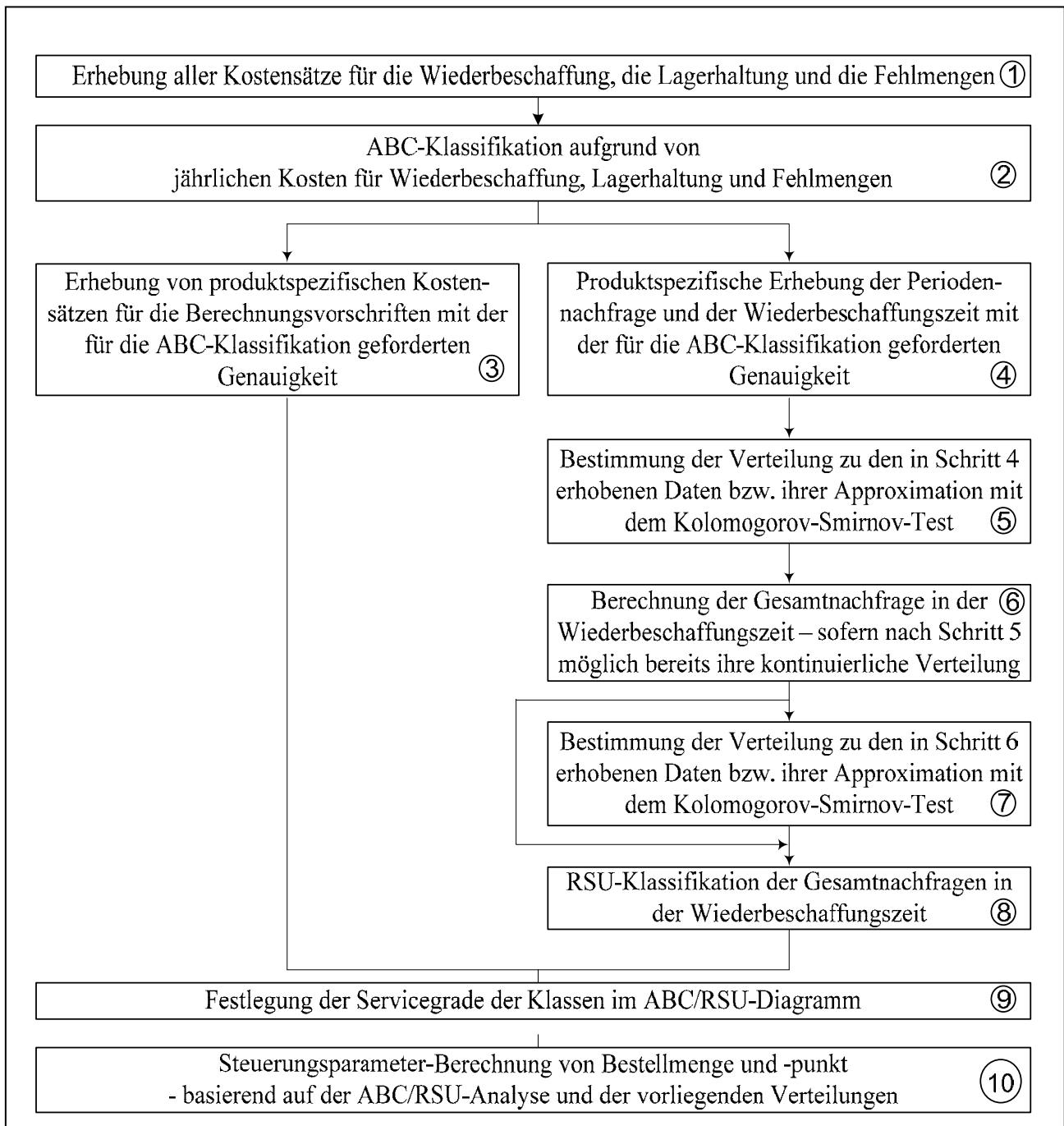


Abbildung 5: Leitfaden zum Bestandsmanagement

Kostensätze für die Schritte 1 und 3 gelesen werden, und die Bewegungsdaten eines ERP- bzw. PPS-Systems enthalten in der Regel die für Schritt 4 benötigten Periodennachfragen und Wiederbeschaffungszeiten. Zur Erreichung der gewünschten Genauigkeit bietet es sich an, dem Benutzer über eine interaktive Eingriffsmöglichkeit zu erlauben, die Daten anzusehen und zu ändern. Um im Schritt 2 die übliche ABC-Klassifikation anwenden zu können, werden die drei Einzelkosten unternehmensspezifisch gewichtet und zu einem Gesamtkostenwert addiert. Über die interaktive Eingriffsmöglichkeit kann ein Benutzer die Gewichtung variieren. Es sei erwähnt, dass durch die Analyse der Abweichung der Lorenzkurve von der Gerade durch den

Nullpunkt mit einer Steigung von eins, die die Ideallinie darstellt, eine Trennung zwischen A-, B- und C-Teilen, also Schritt 2, algorithmisch realisiert werden kann. Schritt 8 kann ebenfalls durch ein Programm erfolgen; in (Tempelmeier 2008) wurde skizziert, wie dabei methodisch vorzugehen ist. Für die Durchführung des Kolmogorov-Smirnov Tests in Schritt 5, wie auch in Schritt 7, existieren kommerziell verfügbare Softwaresysteme, wie beispielsweise das SPSS-System. Mit Programmen für die oben genannten Prognoseverfahren sind die vorgesehenen Approximationen in Schritt 5, wie auch in Schritt 7, durch ein Softwaresystem durchführbar. Unvermeidlich ist die Durchführung von Schritt 9 durch einen Benutzer, die wie bisher über eine interaktive Eingriffsmöglichkeit erfolgen



soll. Ein solches Softwaresystem wurde am IPF implementiert. Die produktspezifischen Daten werden in einer Oracle-Datenbank abgelegt. Über eine Schnittstelle können sie aus dem SAP System automatisch extrahiert werden; das Vorgehen folgt dem in (Herrmann 2007) vorgestellten. Zur Anbindung alternativer ERP- oder PPS-Systeme lässt sich nach dem gleichen Prinzip eine Schnittstelle implementieren. Der Kolmogorov-Smirnov Test erfolgt über eine Schnittstelle zum SPSS-System. Das Endergebnis in Form von konkreten Steuerungsparametern für jedes Produkt kann ebenfalls über eine Schnittstelle zum SAP System übertragen und im Materialstamm eingetragen werden.

#### Anwendung in einem Unternehmen

Abschließend sei erwähnt, dass der Leitfaden in einem Produktionsunternehmen angewandt worden ist, welches seinen Kunden eine Lieferzeit in Deutschland innerhalb von 24 Stunden und in Europa innerhalb von 48 Stunden verspricht. Das betrachtete Sortiment bestand aus 14000 Teilen und die Periodennachfragen sowie die Lieferzeiten der letzten drei Jahre wurden berücksichtigt. Mit dem Leitfaden konnte die Einhaltung der Lieferzeit soweit verbessert werden, dass die Kunden diese Zusage als sehr gut erfüllt ansehen. Zugleich wurde der mittlere Bestand an A-Teilen um 55%, an B-Teilen um 50% sowie an C-Teilen um 25% verringert. Wegen des starken Wachstums des Unternehmens konnte dadurch der Bau einer neuen Lagerfläche hinausgezögert werden. Dank des Werkzeugs wurde dieses Bestandsmanagementprojekt von einem Mitarbeiter innerhalb von einer Woche durchgeführt.

#### 4. FAZIT UND AUSBLICK

Die vorgestellte Arbeit beschreibt die in der Literatur publizierten einzelnen Varianten einer (s,q)-Lagerhaltungspolitik. Sie verbindet diese zu einem Leitfaden für die Durchführung von Bestandsmanagementprojekten. Dabei wurden die Vorschläge aus der Literatur berücksichtigt und erweitert. Es ist zu erwarten, dass seine Anwendung die Lagerbestände in einem Produktionsunternehmen signifikant verringert. Allerdings ist seine manuelle Anwendung sehr aufwendig. Durch das vorgeschlagene Werkzeug lässt sich dieser so deutlich reduzieren, dass ein Bestandsmanagementprojekt regelmäßig in größeren Zeitabständen, wie beispielsweise alle zwei oder drei Jahre, mit vertretbarem Aufwand durchgeführt werden kann. Teilweise werden in Unternehmen alternative Servicegrade eingesetzt, so dass in diesem Sinne das System weiterentwickelt werden wird. Unberücksichtigt blieb bisher auch das Vorliegen von beschränkten Lagerkapazitäten. Allerdings führt dies zu mathematisch deutlich anspruchsvolleren Modellen. Hieran wird am IPF geforscht werden.

#### 5 LITERATUR

Arrow, K.; T. Harris; und J. Marschak. 2005: Optimal Inventory Policy. *Econometrica*, 19, S. 250 – 272.

Arrow, K.; S. Karlin; und P. Suppes. (Herausgeber). 1960: *Mathematical Methods in Social Sciences*. Stanford University Press, Stanford, USA.

Assfalg H. 1976: *Lagerhaltungsmodelle für mehrere Produkte*. Hain Verlag.

Bartmann, D. und M. Beckmann. 1090: *Lagerhaltung, Modelle und Methoden*. Springer Verlag, Berlin.

Bellmann, R.; I. Glicksberg; O. Gross. 1955: On the optimal Inventory Equation. *Management Science* 2, S. 83 – 104.

Brunnberg, J. 1970: *Optimale Lagerhaltung bei ungenauen Daten*.

Chen, F. und Y. Zheng. 1994: Lower Bounds for Multi-Echelon Stochastic Inventory Systems. In: *Management Science* 40, S. 1426 – 1443.

Clark, A. J. und H. E. Scarf. 1960.: Optimal Policies for a Multi-Echelon Inventory Problem. In: *Management Science* 6, S. 475 – 490.

Eppen, G. und L. Schrage. 1981: Centralized Ordering Policies in a Multiwarehouse System with Lead Times and Random Demand. In: Schwarz, L.: *Multi-Level Production/Inventory Control Systems: Theory and Practice*. North-Holland Publishing Company, Amsterdam, 1981.

Erlenkotter, D. 1990: Ford Whitman Harris and the Economic Order Quantity Model. In: *Operations Research* 38, S. 937 – 946.

Federgruen, A. und M. Tzur. 1991: A Simple Forward Algorithm to Solve General Dynamic Lot Sizing Models with n periods in or time. In: *Management Science* 37 (1991), S. 909 – 925.

Federgruen, A. und P. Zipkin. 1984: Approximation of Dynamic, Multi-Location Production and Inventory Problems. In: *Management Science* 30, S. 69 – 84.

Federgruen, A. und P. Zipkin. 1984: Computational Issues in the Infinite Horizon, Multi-Echelon Inventory Model. In: *Operations Research* 32, S. 818 – 836.

Federgruen, A. und P. Zipkin. 1984: Allocation Policies and Cost Approximation for Multi-Location Inventory Systems. In: *Naval Research Logistics Quarterly* 31, S. 97 – 131.

Gudehus, Timm. 2003: *Logistik – Grundlagen, Verfahren und Strategien*. Springer Verlag, Hamburg, (2. Auflage).

Hadley, G. und T. M. Whitin. 1963: *Analysis of Inventory Systems*. Englewood Cliff, Prentice Hall, New Jersey.

Herrmann, Frank. 2007: SIM-R/3: Softwaresystem zur Simulation der Regelung produktionslogistischer Prozesse durch das R/3-System der SAPAG. In: *Wirtschaftsinformatik* 49, 2, S. 127 – 133.

Herrmann, Frank. 2009: *Logik der Produktionslogistik*. Oldenbourg Verlag, Regensburg.

Hochstädter D. 1969: *Stochastische Lagerhaltungsmodelle*. Springer Verlag, Berlin.

Iglehart, D. L. 1963: Optimality of (s, S) Policies in the Infinite Horizon Dynamic Inventory Problem. In: *Management Science* 9, S. 259 – 267.

Iglehart, D. L. 1963: Dynamic Programming and Stationary Analysis in Inventory Problems. In: Scarf H.; Guilford, D. und Shelly, M. (Herausgeber): *Multi-Stage*

- Inventory Models and Techniques, Stanford University Press, Stanford CA, USA, S. 1 – 31.
- Iglehart, D. L. und S. Karlin. 1962: Optimal Policy for Dynamic Inventory Process with Nonstationary Stochastic Demands. In: Arror, K. J.; Karlin, S.; Scarf, H. (Herausgeber): Studies in Applied Probability and Management Science, Stanford University Press, Stanford, USA, Kapitel 8.
- Karlin, S. und H. Scarf. 1958: Inventory Models and Related Stochastic Processes. In: Arror, K. J.; Karlin, S.; Scarf, H. (Herausgeber): Studies in the Mathematical Theory of Inventory and Production, Stanford University Press, Stanford, USA, Kapitel 17.
- Klemm, H. und M. Mikut. 1972: Lagerhaltungsmodelle. Verlag Die Wirtschaft, Jena und Dresden.
- Liu, L.; X. Liu und D. D. Yao. 2004: Analysis and optimization of a multistage inventory-queue system. In: Management Science 50, S. 365 – 380.
- Porteus, E. L. 1990: Stochastic Inventory Theory. In: Heyman, D. P.; Sobel, M. J. (Herausgeber): Handbooks in Operations Research and Management Science, the volume on Stochastic Models. North-Holland, Amsterdam, 1990, S. 605 – 652.
- Robrade A. 1991: Dynamische EinproduktLagerhaltungsmodelle bei periodischer Bestandsüberwachung. Physica-Verlag, Hamburg.
- Rosling, K. 1989: Optimal Inventory Policies for Assembly Systems Under Random Demand. In: Operations Research 37, S. 565 – 579.
- Ropp, W. 1968: Einführung in die Theorie der Lagerhaltung.
- Roundy, R. 1985: 98%-Effective Integer-Ratio Lot-Sizing for One-Warehouse Multi-Retailer Systems. In: Management Science 31, S. 1416 – 1430.
- Scarf, Herbert E. 1959: The Optimality of (S,s)-Policies in the Dynamic Inventory Problem. In: Arror, K. J.; Karlin, S.; Suppes, P. (Herausgeber): Mathematical Methods in Social Sciences. Stanford University Press, Stanford, USA, S. 196 – 202.
- Scarf, Herbert E. 1960: Optimal Policies for the Inventory Problem with Stochastic Lead Time. Planning Research Corporation, PRC R-181, Los Angeles.
- Scarf, Herbert E. 1963: A Survey of Analytic Techniques in Inventory Theory. In Scarf, Herbert E.; Gilford, Dorothy M.; Shelly, Maynard W.: Multistage inventory models and techniques. Stanford University Press, Stanford, USA, S. 185 – 225.
- Scarf, Herbert E.; Dorothy M. Gilford und Maynard W. Shelly. 1963: Multistage inventory models and techniques. Stanford University Press, Stanford, USA.
- Schneider H. 1979: Servicegrade in Lagerhaltungsmodellen. M+M Wissenschaftsverlag, Berlin.
- Silver, Edward A.; David F. Pyke; Rein Peterson. 1998: Inventory Management and Production Planning and Scheduling. Wiley, New York (3. Auflage).
- Suchanek, B. 1996: Sicherheitsbestände zur Einhaltung von Servicegraden.
- Tempelmeier, Horst. 1983: Lieferzeit-orientierte Lagerungs- und Auslieferungsplanung. Physica-Verlag, Trier, 1983.
- Tempelmeier, Horst. 2005: Bestandsmanagement in Supply Chains. Books on Demand GmbH, Norderstedt, Köln.
- Tempelmeier, Horst. 2008: Materiallogistik. Springer Verlag, Köln.
- Veinott, A. und H. Wagner. 1965: Computing Optimal (s, S) Inventory Policies. In: Management Science 11, S. 525 – 552.
- Zheng, Y. S. 1991: A Simple Proof for the Optimality of (s, S) Policies for Infinite Horizon Inventory Problems. In: J. Appl. Prob. 28, S. 802 – 810.
- Zheng, Y. S. und A. Federgruen. 1991: Finding Optimal (s, S) Policies is About as Simple as Evaluating a Single Policy. In: Operations Research 39, S. 654 – 665.
- Zipkin, P. H. 2000: Foundations of Inventory Management. Irwin, Burr Ridge, IL, USA.

#### AUTOR BIOGRAPHIE



**Frank Herrmann**, geboren in Münster (Westfalen), Deutschland, schloss sein Informatik-Studium an der RWTH Aachen in 1989 mit einem Diplom ab; Schwerpunkte waren theoretische Informatik, Compilerbau und Mathematik. Danach war er bis 1996 Mitarbeiter am Fraunhofer-Institut für Informations- und Datenverarbeitung

in Karlsruhe mit den Schwerpunkten: Algorithmen zur Produktionssteuerung für Forschungsprojekte sowie für Industrieunternehmen (Promotion über Heuristiken zur Fertigungssteuerung zum Dr.-Ing. an der TU Dortmund in 1996) wie auch Beratungsprojekte in der Logistik: Durchführung, Leitung und Akquisition. Von 1996 bis 2003 arbeitete er im Bereich Global Support der SAP AG in Walldorf mit der Aufgabenstellung: Betrieb von R/3-Systemen in Anwendungsunternehmen. Sein fachlicher Schwerpunkt war die Technologie des R/3 (bzw. SAP) Systems und die Logistik-Module im R/3 (bzw. SAP) System; vor allem das SCM/APO System. Er bekleidete verschiedene Managementpositionen bei der SAP AG, zuletzt war er Direktor. Seit 2003 ist er Professor für Informationstechnik und Produktionslogistik an der Ostbayerischen Technischen Hochschule Regensburg mit dem Schwerpunkt in quantitative Methoden, einschließlich Optimierung, für die Produktionsplanung und -steuerung wie auch deren Simulation.

# INTERPRETATION DER KOPPELUNGSMÖGLICHKEITEN ZWISCHEN OPTIMIERUNGSAUFGABEN UND SIMULATIONSMODELLEN AM BEISPIEL VON GESCHÄFTSPROZESSEN

Christian Müller  
TH Wildau, FB Wirtschaft, Informatik, Recht  
Hochschulring 1, D-15745 Wildau, Deutschland  
E-mail: [christian.mueller@th-wildau.de](mailto:christian.mueller@th-wildau.de)

**Keywords:**

Simulation, Optimierung, Geschäftsprozesse

[DesmoJ (2013)], [Page (2005)].

**Abstrakt:**

Geschäftsprozesse in modernen Unternehmen zeichnen sich durch eine hohe Komplexität aus, die sich z.B. aus sich schnell verändernden Marktsituationen, verkürzten Produktlebenszyklen bzw. dynamischen Wechselwirkungen zwischen einzelnen Subsystemen ergeben. Das Geschäftsprozessmanagement hat das Ziel möglichst effiziente und kundenorientierte Prozesse zu implementieren. Klassischer Weise werden dazu analytische kennzahlenbasierte Verfahren verwendet. Im Gegensatz dazu sollen hier die Geschäftsprozesse mit Simulationsverfahren untersucht werden. Auf diese Weise können die Prozesse auch schon vor ihrer Implementation beurteilt werden.

Da reale Geschäftsprozesse in der Regel spezifische Entscheidungsprobleme beinhalten, die durch Optimierungsansätze gelöst werden können, erscheint es sinnvoll, eine Simulation von Geschäftsprozessen mit einer Optimierungskomponente zu verbinden [Steglich, Müller (2013)]. Andererseits liegen Geschäftsprozessmodelle oft in parametrisierter Form vor, so das sich die Frage nach einer optimalen Wahl der Parameter stellt. [März et.al. (2010), p 3ff]. Somit stellt sich die Frage nach den möglichen Konstellationen zwischen Optimierungs- und Simulationsaufgabe und deren Interpretation.

Dazu entwickelten März und Krug [März et al. (2010), Kap. 5] ein Klassifikationsschema, welches die grundlegende Kopplungsarten zwischen Simulation und Optimierung beschreibt. Dieser Artikel diskutiert anhand einfacher Beispiele, welche sich aus Geschäftsprozessmodellen ableiten lassen, diese Koppelungsmöglichkeiten. Er baut auf den Überlegungen aus [Müller, Chr (2013)] auf, ergänzt diese durch durch sinngebende Interpretationen und weist auf Schwachstellen hin.

Zur Illustration der Beispiele wurde der EPC Simulator [EPC Simulator (2013)], [Müller, Chr (2012a), (2012b)] verwendet. Dieser ist ein Plugin des Modellierung-Werkzeugs Bflow [Bflow (2013)], [Kern et al. (2010)] und nutzt das DesmoJ Simulations- Framework

**Das Klassifikationsschema von März**

In [März et al. (2010)] wird das Zusammenspiel von Simulation und Optimierung anhand von industriellen Fallbeispielen untersucht. Sie klassifizieren das Zusammenspiel auf vier grundlegende Kopplungsarten.

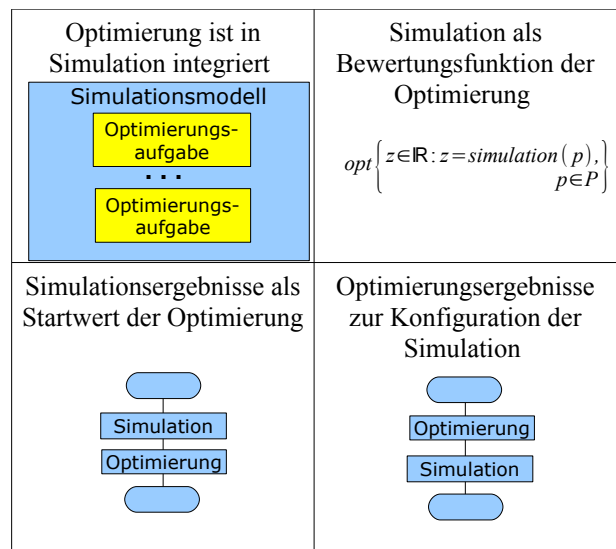


Abbildung 1: Klassifikationsschema von März und Krug

Dabei unterscheiden sie im wesentlichen zwischen sequentieller und hierarchischer Verknüpfung, der zeitlichen Abfolge der Berechnungen und deren gegenseitigen Abhängigkeiten. Ebenfalls sind Kombinationen zwischen diesen Grundtypen denkbar.

**Beispiel: Optimierung eines Bedienprozesses**

Ein Supermarkt ist von 8:00 bis 20:00 geöffnet. Die Verteilung der Zeitabstände, in denen die Kunden den Supermarkt betreten, ist vorgegeben. Jeder Kunde schaut nach dem Betreten des Geschäfts in den Kassenbereich. Wenn dort an jeder Kasse mehr als drei Kunden warten, dann verlässt er das Geschäft wieder. Es wird angenommen, das jeder Kunde einen durchschnittlichen Erlös von € 5,- generiert und jeder Kassierer € 20,- pro Stunde kostet. Es soll die optimale Anzahl von Kassierern bestimmt werden, so das der

Gesamterlös des Geschäfts maximiert wird.

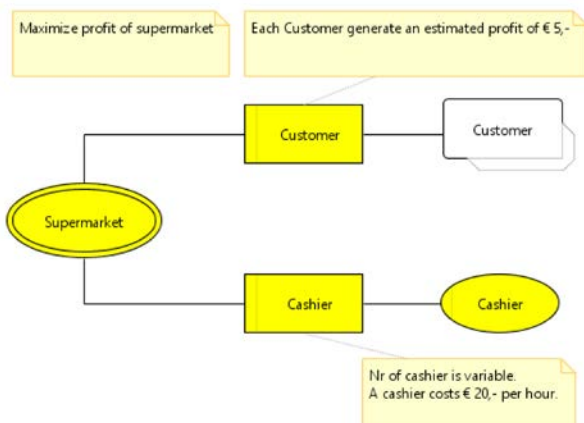


Abbildung 2: Modelldokument Bedienprozess

Das Simulationsmodell wird durch das Modell-dokument (Abb. 2) und das Prozessdokument (Abb. 3) beschrieben.

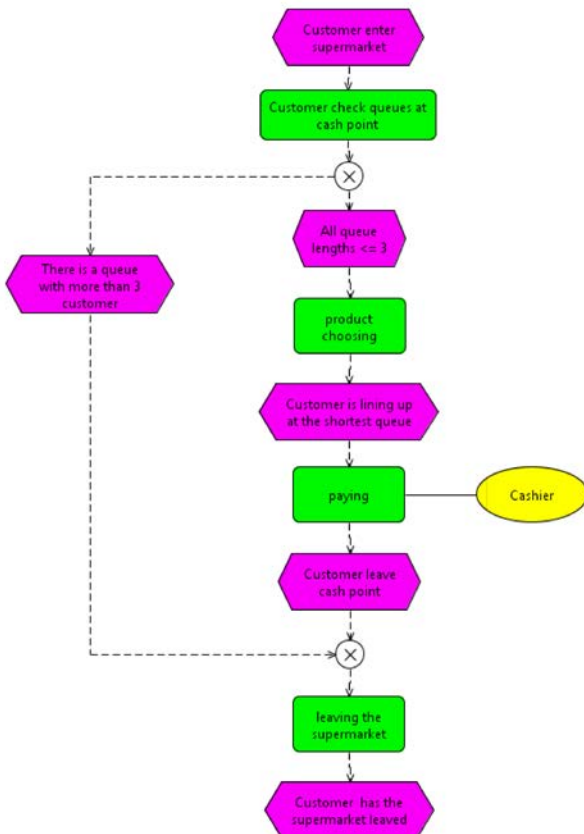


Abbildung 3: Prozessdokument Bedienprozess

Der erwartete Erlös kann mit dem Simulationsmodell, nach mehrmaliger Ausführung zu einer vorgegebenen Anzahl von Kassieren, bestimmt werden.

Mit einem externen Optimierungssystem kann nun die optimale Anzahl von Kassieren bestimmt werden. Damit ist das Simulationsmodell der Optimierungsaufgabe hierarchisch untergeordnet. Das Beispiel fällt damit in die Klasse „Simulation als Bewertungsfunktion

für die Optimierung“.

**Beispiel: Ein einfaches Fahrzeunavigationsmodell:**

In einem ungerichteten Straßennetzwerk  $G = (V, A)$ , wobei  $V$  die Knoten und  $A$  die Kantenmenge ist, fahren Fahrzeuge von zufälligen Start zu zufälligen Zielknoten (Abbildung 4). Die Passierdauer auf einer Kante  $(i,j)$  ist normal verteilt, wobei die mittlere Passierdauer  $dist_{i,j} = ft_{i,j} + vt_{i,j} \cdot q_{i,j}$  abhängig von der aktuellen Anzahl der Fahrzeuge  $q_{i,j}$  auf dieser Kante  $(i,j)$  ist.

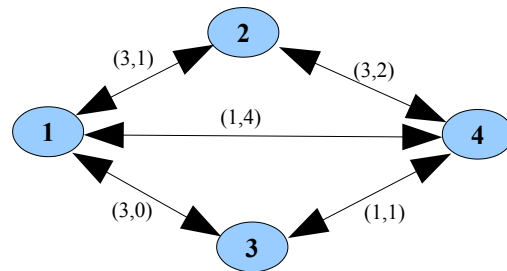


Abbildung 4: Strassennetzwerk  $G = (V, A)$  mit Kantenbewertungen  $(ft_{i,j}, vt_{i,j})$

Bei der Fahrt eines Autos wird zur Wahl der nächsten Kante, basierend auf der mittleren Passierdauer, in jedem Knoten ein kürzestes Wege Problem gelöst. Mit diesem Modell kann der Einfluss der lokalen Navigation in jedem Fahrzeug auf den globalen Verkehrsfluss studiert werden.

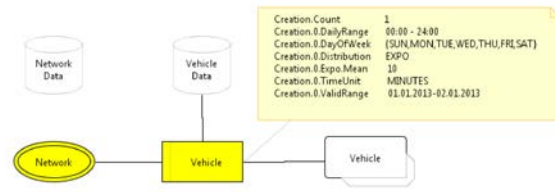


Abbildung 5: Modelldokument Fahrzeugnavigation

Um mit dem EPC Simulator ein Simulationsmodell zu generieren wird ein Modelldokument (Abb. 5) und ein Prozessdokument (Abb. 6) erstellt.

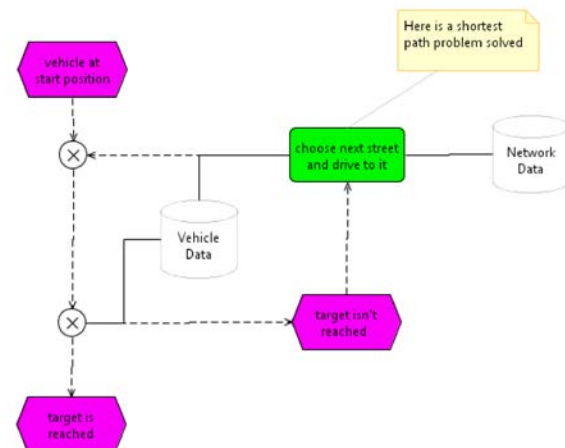


Abbildung 6: Prozessdokument Fahrzeugnavigation

In dem Modelldokument ist eine globale Datenstruktur „NetworkData“ deklariert. Insbesondere sind dort das Netzwerk, die Kantenbewertungen und die Anzahl der Fahrzeuge auf den einzelnen Kanten gespeichert.

Für jedes Fahrzeug werden die individuellen Fahrzeugdaten, dies sind insbesondere die aktuelle Position, das Ziel und eine Methode zur Berechnung des kürzesten Weges zwischen 2 Knoten des Netzwerks, in der Vehikel Datenstruktur gespeichert. Das Verhalten der Fahrzeuge ist dem Prozessdokument (Abbildung 4) beschrieben. Das Fahrzeug fährt von Knoten zu Knoten bis sein Ziel erreicht ist. In jedem Knoten wird ein Kürzestes Wege Problem gelöst, aus dessen Lösung die nächste zu befahrende Kante abgeleitet wird.

Das einzubindende Kürzeste Wege Problem [Hiller et al. (2010)] lässt sich mathematisch als:

$$\sum_{(i,j) \in A} c_{i,j} \cdot x_{i,j} \rightarrow \min!$$

s. t.

$$\sum_{(i,j) \in A} x_{i,j} - \sum_{(j,i) \in A} x_{j,i} = \begin{cases} 1 & \text{if } i=s \\ -1 & \text{if } i=t \\ 0 & \text{otherwise} \end{cases} \quad \text{für alle } i \in V$$

$$x_{i,j} \geq 0; \text{ für alle } (i,j) \in A$$

formulieren.

Bei diesem Modell ist die „Optimierung in das Simulationsmodell integriert“. Jedes Fahrzeug bestimmt in jedem Knoten individuell, basierend auf der globalen Verkehrssituation, seinen optimalen Weg. Damit beeinflusst es selbst die globale Verkehrssituation. Somit sind die Optimierungsaufgaben dem Simulationsmodell hierarchisch untergeordnet und es besteht eine direkte Rückkoppelung der Lösung der Optimierungsaufgabe in das Simulationsmodell.

### Beispiel: Ein einfaches Job Shop Problem

In einer Werkstatt liegen die Aufträge eines Tages vor. Die Werkstatt hat 5 Mitarbeiter, die 8 Std. pro Tag, also zusammen 2400 Minuten pro Tag arbeiten. Es gibt fünf Auftragsstypen wobei jeder Typ eine Planzeit hat. Die Aufträge des Tages überschreiten, gemessen an ihren Planzeiten, die Tagesarbeitszeit von 2400 Minuten nicht. Die Mitarbeiter sind spezialisiert. Ihre Bearbeitungszeiten sind normal verteilt mit individuellen Verteilungsparametern  $(\mu, \sigma)$ .

Bearbeitungszeit [min] $N(\mu, \sigma)$	Mitarbeiter					Planzeit [min]	
	1	2	3	4	5		
Auftrags typ	A	(15,5)	(20,7)	(30,10)	(40,15)	(45,15)	30
	B	(60,10)	(45,10)	(70,10)	(80,15)	(90,20)	60
	C	(80,15)	(80,20)	(60,10)	(90,10)	(100,10)	90
	D	(130,10)	(140,10)	(120,10)	(100,15)	(90,10)	120
	E	(170,40)	(160,30)	(150,10)	(120,30)	(130,20)	150

Tabelle 1: Job Shop Problem

Morgens werden jedem Mitarbeiter von einem Manager die von ihm zu bearbeitenden Aufträge zugewiesen.

Dies ist eine stochastische Optimierungsaufgabe [Kall et al. (1994)], [Birge et al. (1997)]. Diese wird hier mit dem „expected-value“ Ansatz [Madanski (1960)] bearbeitet. Dabei wird die stochastische Optimierungsaufgabe in eine deterministische überführt, indem die Verteilungen durch ihre Erwartungswerte ersetzt werden. Anschließend werden durch eine nachgelagerte Simulation die Auswirkungen dieser Vereinfachung überprüft.

Das Modelldokument für diese Aufgabe ist in Abbildung 7 gegeben.

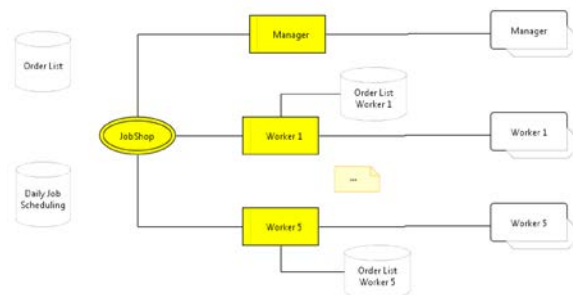


Abbildung 7: Modelldokument Job Shop Problem

In der Datenstruktur Order List stehen die Aufträge, in Daily Job Scheduling die von dem Manager vorgenommene Zuordnung der Aufträge auf die Mitarbeiter. In den Order List Worker <i> befinden sich die individuellen Arbeitspläne der Mitarbeiter. Die Planungstätigkeit des Managers ist in Abbildung 8 beschrieben.

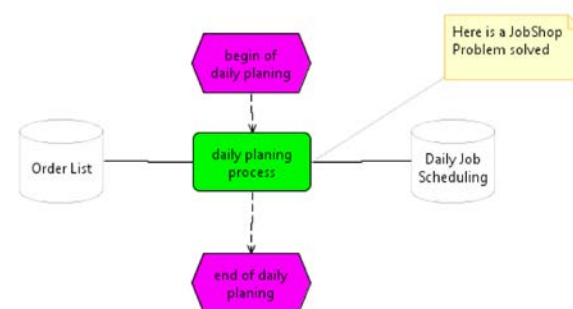


Abbildung 8: Prozessdokument Manager im Job Shop Problem

Dabei muss er ein Job Shop Problem lösen. Das Planungsergebnis, die Zuordnung der Aufträge auf die Mitarbeiter wird in der Daily Job Scheduling Datenstruktur gespeichert.

Das deterministische Job Shop Problem [Domschke et al. (1997)] hat folgende Form:

Indexmengen:

- $a \in A$  Menge der Aufträge eines Tages
- $m \in M$  Menge der Mitarbeitern
- $t \in T$  Menge der Auftragsstypen

Daten:

$$at(a, t) = \begin{cases} 1 & \text{wenn Auftrag } a \in A \text{ vom Typ } t \in T \text{ ist.} \\ 0 & \text{sonst} \end{cases} \quad \text{für alle } a \in A \text{ und } t \in T$$

$$\mu(t, m) = \text{mittlere Bearbeitungszeit eines Auftrages vom Typ } t \in T \text{ bei Mitarbeiter } m \in M$$

Entscheidungsvariablen:

- freie *arbeitszeit*

$$x(a, m) = \begin{cases} 1 & \text{Auftrag } a \in A \text{ wird von Mitarbeiter } m \in M \text{ ausgeführt} \\ 0 & \text{sonst} \end{cases} \quad \text{für alle } a \in A \text{ und } m \in M$$

$$\text{(SOS Bedingung)} \quad \sum_{m \in M} x(a, m) = 1 \quad \text{für alle } a \in A$$

Ziel: *min arbeitszeit*

Restriktionen:

$$\sum_{\substack{a \in A \\ t \in T}} at(a, t) \mu(t, m) x(a, m) \leq \text{arbeitszeit} \quad \text{f.a. } m \in M$$

In dieser Formulierung werden die Aufträge zeitlich gleichmäßig auf alle Mitarbeiter verteilt, so dass die von einem beliebigen Mitarbeiter maximal zu leistende Arbeitszeit minimal wird.

Nach Abschluss der Planung arbeiten die Mitarbeiter ihre Pensen (Abbildung 9) ab.

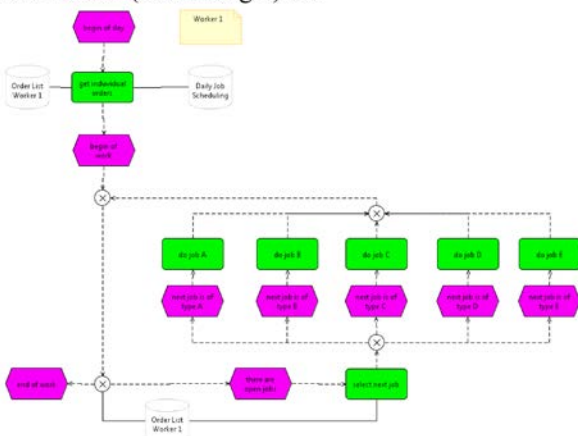
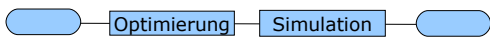


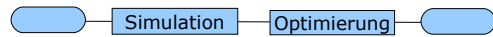
Abbildung 9: Prozessdokument Worker in Job Shop Problem

In diesem Beispiel wird auf der Basis der erwarteten Bearbeitungszeiten ein deterministisches Optimierungsmodell gelöst, da von der Zufälligkeit der Bearbeitungszeiten abstrahiert wird. Anschließend wird die Lösung des Optimierungsmodells mit einer Simulation überprüft. Hier besteht keine Wechselwirkung zwischen Optimierungsaufgabe und Simulationsmodell. Beide Schritte sind sequentiell angeordnet.

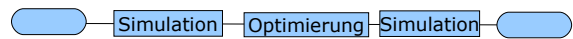


Wird die Aufgabe derart modifiziert, dass die Aufträge mittels Simulation erzeugt werden und anschließend die

Planungsaufgabe gelöst wird, dann erhalten wir bei Vernachlässigung der nachgelagerten Simulation die sequentielle Struktur:



Ohne Vernachlässigung der nachgelagerten Simulation erhalten wir:



Auch diese Struktur ist in dem Klassifikationsschema als eine Kombination der Grundtypen vorgesehen.

### Interpretation der Beispiele in Bezug auf das Klassifikationsschema

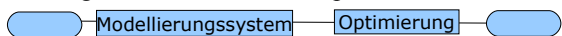
Die oben dargestellten Beispiele decken alle Grundtypen des Klassifikationsschemas ab. Grundsätzlich sind die beiden hierarchischen Fälle

- *Simulation als Bewertungsfunktion der Optimierung* und
- *Optimierung ist in die Simulation integriert* zu unterscheiden. Die sequenziellen Fälle
- *Simulationsergebnis als Startwert der Optimierung* und
- *Optimierungsergebnisse zur Konfiguration der Simulation*

sind einerseits Spezialfälle des hierarchischen Falls, „Optimierung ist in die Simulation integriert“ und andererseits legen die beiden Job-Shop Beispiele auch eine Interpretation als stochastische Validierung eines deterministischen Optimierungsergebnisses



bzw. als Modellierungssystem mit zufälligen Auftragsdaten für das Job Shop Problem nahe.



März und Krug beschreiben die stochastische Validierung als „Überprüfung der Machbarkeit der vorgeschlagenen Lösung“ [März et al. (2010) p44]. Bei dem Begriff Modellierungssystem folgen wir Kallrath [Kallrath (2004)], der eine Übersicht über kommerziell verfügbare Systeme gibt. Für Open Source System sei auf Koch [Koch (2004)] und Steglich, Schleiff [Steglich, Schleiff (2010)] verwiesen. Das Modellierungssystem enthält einen stochastischen Aspekt, wenn einige Daten des Modells durch Zufallsvariablen beschrieben werden und das Modellierungssystem zum Erstellen eines deterministischen Modells konkrete Zufallsvariablen zieht.

In beiden Fällen liegt eine sequentielle Abfolge, eine charakterisierende Eigenschaft dieser Spezialfälle, vor. Somit bieten sich folgende Umbenennungen im Klassifikationsschema an:

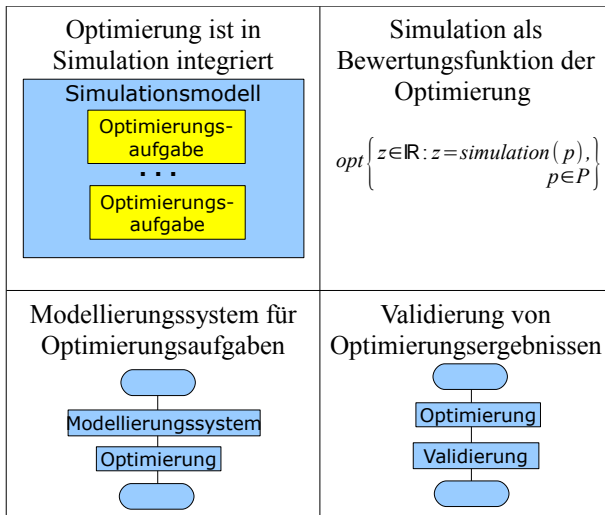
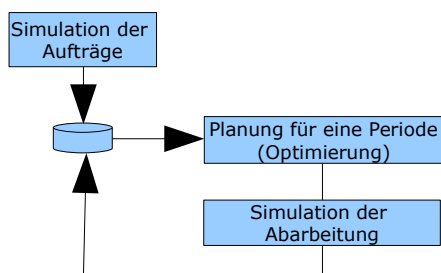


Abbildung 10: Modifiziertes Klassifikationsschema

Um die Grenzen dieser Klassifikation aufzuzeigen soll abschließend eine geringfügige Modifikation des einfachen Job Shop Problems (Abb. 11) diskutiert werden. Dabei wird als Simulationszeitraum nicht nur ein Tag, sondern eine Folge von Tagen betrachtet. Falls in dem Verifikationsschritt eines Tages festgestellt wird, das nicht alle für diesen Tag geplante Aufträge in der zulässigen Arbeitszeit erledigt werden können, dann werden sie auf den nächsten Tag verschoben. Durch diese naheliegende Regel entsteht eine Rückkoppelung von dem letzten in den ersten Simulationsschritt, die auch Konsequenzen auf die noch zu vergebenden Auftragstermine haben kann. Somit liegt nun der Fall „Optimierung ist in Simulation integriert“ vor.



Rückführung der nicht bearbeiteten Aufträge  
Abbildung 11: Modellvariation mit Klassenwechsel

Um solche Klassifikationswechsel durch geringfügige Modellvariationen zu vermeiden, bietet es sich an, nicht nur formalen Kriterien, sondern auch den Anwendungskontext zu betrachten. Bei den Job-Shop Beispielen dominiert der Anwendungskontext und deshalb sollte sie der Gruppe „Optimierung ist in die Simulation integriert“ zugeordnet werden. Falls allerdings der Validierungsaspekt bzw. der Modellierungsaspekt einer Optimierungsaufgabe überwiegen, dann sollte die entsprechende sequentielle Klasse betrachtet werden. In diesem Fall sind dann auch keine Modellvariationen zu erwarten.

März et. al. klassifizierten in [März et al.(2010)] 12 industrielle Praxisbeispiele aus dem Umfeld der Produktionsplanung, wobei neun Beispiele aus der Klasse „Simulation als Bewertungsfunktion“ und zwei in die Gruppe „Optimierung ist in die Simulation integriert“ kamen. Damit zeigt sich die große Bedeutung der optimalen Parameterwahl in der Praxis. Ferner gab es ein sequentielles Beispiel aus der Klasse „Simulationsergebnisse als Startwert der Optimierung“. In diesem Produktionsplanungsbeispiel wurden mittels Simulation Kapazitätsanforderungen ermittelt, um im Nachgang diese Kapazitätsbedarfe mit konkreten Betriebsmitteln zu verknüpfen [März et al.(2010) p 44]. Auch dieses Beispiel lässt sich durch täglich variierende Kapazitätsanforderungen und teilweise nicht ausreichende Betriebsmittel leicht zu einem nicht sequentiellen Modell erweitern. Bei Einordnung entsprechend des Modellkontextes in die Gruppe „Optimierung ist in Simulation integriert“ kann auch hier ein Klassifikationssprung vermieden werden.

#### Zusammenfassung:

Ausgehend von dem Klassifikationsschema von März und Krug wurden einfache Beispiele vorgestellt, die die entsprechenden Koppelungsmöglichkeiten illustrieren. Dabei wurde festgestellt, dass die Klasse „Optimierung ist in Simulation integriert“ eine Oberklasse der beiden sequentiellen Klassen ist. Diese Klassen können allerdings als spezielle Modellierungs- bzw. Validierungs- Aufgabe interpretiert werden und erhalten damit eine eigene methodische Bedeutung. Praxisbeispiele die diese sequentielle Struktur aufweisen, verlieren allerdings bei naheliegenden Modifikationen diese Struktur. Deshalb ist es sinnvoll Praxisaufgaben bei denen die sequentielle Struktur nicht charakteristisch ist, gleich in die größere Klasse einzuordnen.

Aus Sicht der industriellen Praxis gehören die überwiegende Anzahl der auftretenden Kombinationen in die Klasse „Simulation als Bewertungsfunktion der Optimierung“. Diesen Schluss legen die von März und Krug [März et al. (2010)] vorgestellten Praxisbeispielen nahe.

#### Literatur:

- Birge, Louveaux 1997 „Introduction to stochastic programming“ Springer Berlin Heidelberg
- Bflow 2013 „bflow\*-toolbox“ Projekt Homepage <http://www.bflow.org/>
- DesmoJ 2013 “A Framework for Discrete-Event Modelling and Simulation” <http://desmoj.sourceforge.net/home.html>
- EPC Simulator 2013 „EPC Simulator Projekt Homepage“ <http://www.tfh-wildau.de/cmuller/EpcSimulator/>
- Domschke, W.; Scholl, A.; Voß, S 1997 “Produktionsplanung”. Springer Berlin Heidelberg
- Hiller, Liebermann 2010 “Introduction to Operations Research” McGraw-Hill Higher Education
- Kall, Wallace 1994 “Stochastic Programming” John Wiley and Sons
- Kallrath, Josef 2004 “Modeling Languages in Mathematical

- Optimization” Kluwer Academic Publishers
- Kern, Kühne, Laue, Nüttgens, Rump, Storch 2010 “bflow\* Toolbox - an Open-Source Business Process Modelling Tool” Proc. of BPM Demonstration Track, Business Process Management Conference 2010 (BPM'10), Hoboken, USA
- Koch, Thorsten 2013 “Zimpl” <http://zimpl.zib.de/>
- Madanski 1960 „Inequalities for stochastic linear programming problems“ Management Science 6:197-204, 1960
- März et.al. 2010 „Simulation und Optimierung in Produktion und Logistik“ Springer Berlin Heidelberg
- Müller, Chr 2012a „Generation of EPC based simulation models“ Proceedings 26th European Conference on Modelling and Simulation 2012 <http://www.scs-europe.net/dlib/2012/2012-0301.htm>
- Müller, Chr 2012b „Generierung von Simulationsmodellen aus ereignisgesteuerten Prozessketten“ Management und IT, AKWI Jahrestagung 2012 <http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:kobv:526-opus-1709>
- Müller, Chr 2013 “Anmerkungen zur Klassifikation der Koppelungsarten zwischen Optimierungsaufgaben und Simulationsmodellen am Beispiel von Geschäftsprozessen” Integration und Konnektion, AKWI Jahrestagung 2013 <http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:kobv:526-opus-1761>

- Page, B et. al. 2005 “The Java Simulation Handbook” Shaker
- Steglich M, Schleiff T 2010 „CMPL: Coliop Mathematical Programming Language“ Wildauer Schriftenreihe - Entscheidungsunterstützung und Operations Research, Beitrag 1, Technische Hochschule Wildau [FH]
- Steglich M, Müller, Chr 2013 „An approach to combine simulation and optimization of business process“ Proceedings 27th European Conference on Modelling and Simulation <http://www.scs-europe.net/dlib/2013/2013-0808.htm>



**CHRISTIAN MÜLLER** studierte Mathematik an der Freien Universität Berlin und promovierte 1989 über Netzwerkflüsse mit Nebenbedingungen. Von 1990 bis 1992 arbeitete er bei der Schering AG und von 1992 bis 1994 bei den Berliner Verkehrsbetrieben (BVG) im Bereich Fahr- und Dienstplan Optimierung. 1994 erhielt er einen Ruf an die Technische Hochschule Wildau, Deutschland. Seine Forschungsschwerpunkte sind die Konzeption von Informationssystemen, Mathematische Optimierung und die Simulation von Geschäftsprozessen. Seine Email Adresse ist [christian.mueller@th-wildau.de](mailto:christian.mueller@th-wildau.de) und seine Web Seite ist <http://www.th-wildau.de/cmuellet/>.



# Benchmarking eines Transportplanungsoptimierers einer integrierten Supply-Chain-Management Anwendung

Norbert Ketterer  
Wirtschaftsinformatik  
Fachbereich Angewandte Informatik  
Hochschule Fulda  
D-36039, Fulda  
E-mail: norbert.ketterer@informatik.hs-fulda.de

## STICHWORTE

SAP-SCM 7.0, Benchmarking, Transportplanung, Optimierung

## ABSTRACT

Transportplanungsprobleme werden in der Literatur typischerweise nach Art der zu berücksichtigenden Restriktionen (etwa Kapazität, Zeitfenster) sowie der Art der logistischen Aktivität (etwa Anlieferung, Anlieferung plus Abholung) unterschieden. Bildet man das Transportplanungsproblem in ein Modell der Supply-Chain ab, ergeben sich verschiedene Modellrestriktionen und Parameter, die je nach Problem zu berücksichtigen sind.

Moderne integrierte Supply-Chain-Management Systeme, wie etwa das TPVS-Modul in SAP-SCM 7.0 oder gar SAP-TM, sind in der Lage, eine Reihe von Klassen dieser Transportplanungsprobleme direkt abzubilden, falls die Parameter entsprechend gewählt werden.

Für einige Transportplanungsprobleme existieren zudem Benchmarks, die bisher bekannte beste Lösungen mit ihren Kosten pro Lösung spezifizieren und zur Bewertung von Implementierungen von Transportalgorithmen herangezogen werden.

Ziel dieser Arbeit ist es, die wesentlichen bekannten Klassen von Transportplanungsproblemen sowie die zugehörigen Benchmarks darzustellen. Für die wesentlichen Klassen wird die Abdeckung in TPVS dargestellt und die für die Abdeckung notwendigen Parameter spezifiziert. Zudem wird betrachtet, was im Rahmen der integrierten Planung in SCM bei Abbildung der Benchmarks in SNP zu beachten ist. In Ketterer (Ketterer, 2013) wird eine vereinfachte Darstellung dieses Beitrags gegeben, insbesondere ohne Vergleich der Standardheuristiken und Metaheuristiken sowie weniger Details zur konkreten Abbildung in der SCM-Anwendung.

## EINLEITUNG

Das Transportplanungsproblem stellt ein Kernproblem innerhalb des Supply-Chain Management dar; typischerweise wird es in eine Reihe von Teilproblemen klassifiziert, die sich in den verwendeten Restriktionen unterscheiden. Es existieren elementare Lösungsstrategien, die pro Problem in detaillierten Planungsalgorithmen,

i.d.R. via Heuristiken und Metaheuristiken implementiert werden.

Für einige Teilklassen der Probleme existieren Benchmarks, die Szenarien für Kurzstreckentransporte beschreiben, mit denen die Transportplanungsalgorithmen bewertet werden können.

Im SAP-Umfeld existiert bereits seit R/2 die Möglichkeit, Transportplanungen zu implementieren; seit SCM (hier in der Komponente "SAP-SCM-APO-TPVS") existiert auch in der Standardauslieferung ein Optimierer für dieses Problem. Es ist unseres Wissens jedoch bisher nicht dokumentiert, wie die einzelnen Problemklassen sowie deren Benchmarks in die SAP-SCM Welt - hier speziell in Stammdaten und Parameter des TPVS-Optimierers abzubilden sind und die Ergebnisse auf Basis der existierenden Benchmarks bewertet werden können. Das Coding des Optimierers ist nicht bekannt und liegt nicht als einsehbares ABAP-Coding vor.

## TRANSPORTPLANUNGSPROBLEME

Das klassische Transportplanungsproblem (VRP) verallgemeinert das Problem des Handlungsreisenden (TSP) und stellt die Basis für weitere Klassen dar. Oftmals (etwa Cordeau und Laporte, 2007a), (Toth und Vigo, 2002) wird das Problem mit einer Kapazitätsrestriktion und teilweise auch einer zusätzlichen Restriktion der Routenlänge kombiniert. Diese Kombination mit der Kapazität wird oft als „CVRP“ beziehungsweise im zweiten Fall als „DCVRP“ bezeichnet.

Auf eine vollständige formale Formulierung des Problems wird hier verzichtet, sie wird beispielsweise bei Cordeau und Laporte (Cordeau und Laporte, 2007a) oder Toth und Vigo (Toth und Vigo, 2002) beschrieben: die Grundstruktur besteht aus einem kompletten gerichteten Graphen mit  $n+1$  Knoten. Ein Knoten repräsentiert das Depot, die übrigen  $n$  Knoten repräsentieren Kunden bzw. Bedarfslokation mit einem Bedarf  $> 0$ . Jede Kante besitzt Transportkosten; eine Flotte von Fahrzeugen mit Kapazität  $Q$  ist am Depot verfügbar.

Das VRP besteht darin, für die Fahrzeuge einen Weg zu finden, der (Cordeau und Laporte, 2007a), (Soo und Tay, 2009):

- jeden Kunden nur einmal besucht,
- am Depot startet und endet,
- in der Bedarfssumme nicht die Transportkapazität pro Route übersteigt (CVRP),

- in der Routenlänge nicht ein vorgegebenes Limit übersteigt (DCVRP)

Eine Lösung ist eine Menge von  $m$  Touren (Cordeau und Laporte, 2007a), die sich das Depot als gemeinsamen Knoten teilen.

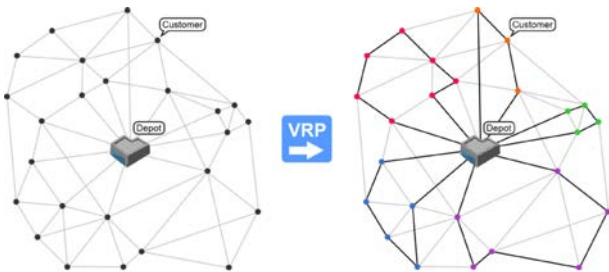


Abbildung 1: Typische Lösung des VRP (Neo, 2013)

Eine wichtige Spezialisierung des CVPR stellt das VRPTW dar, bei dem ein Zeitfenster  $[a_i, b_i]$  pro Knoten vorgegeben wird. Das Fahrzeug kann zwar vor dem Zeitpunkt  $a_i$  den Knoten anfahren, muss dann jedoch unter Erhöhung der Gesamtkosten bis  $a_i$  warten. Nach  $b_i$  wird es nicht mehr bedient. Eine Spezialisierung des VRPTW stellt das VRPPDTW dar (Cordeau und Laporte, 2007b) (Lau und Liang, 2001), bei welchem das Fahrzeug unterwegs auch beladen werden kann und somit Leerkapazitäten zu planen sind. Eine exakte Problemdefinition befindet sich ebenda. VRPB und VRPBTW ähneln dem VRPPD/ VRPPDTW, jedoch mit der Restriktion, dass vor einem Beladen mit Produkten alle Produkte geliefert werden müssen. VRPB und VRPPD werden oftmals auch mit einer Restriktion der Beladereihenfolge kombiniert. Dies entspricht dem Fall, dass ein Fahrzeug oftmals nur LIFO beladen werden kann, da kein freier Zugriff auf die Ladung erfolgen kann – wie etwa wie bei einem Möbelwagen. Eine gängige Klassifikation der verschiedenen Transportplanungsprobleme sowie ihre Beziehung zueinander, findet sich in Toth und Vigo (Toth und Vigo, 2002).

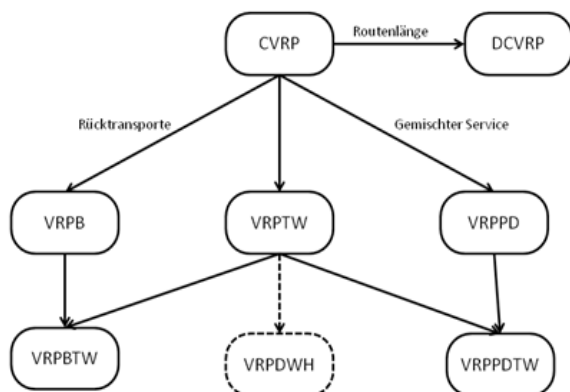


Abbildung 2: Transportplanungsprobleme und deren Beziehung

Eine interessante Erweiterung des VRPTW stellt das VRPDWH dar; hier wird die aktuelle EU-Gesetzgebung bezüglich Fahrt- und Pausenzeiten berücksichtigt (Goel und Gruhn, 2006).

Abbildung 2 basiert auf Toth und Vigo (Toth und Vigo, 2002) wurde jedoch um das VRPDWH erweitert.

### ALGORITHMEN ZUR TRANSPORTPLANUNG

Es existieren exakte Algorithmen für das CVRP, die auf Methoden des "Branch and Bound" und auf "Mengenpartitionierung" basieren, jedoch sind naturgemäß die lösbaren Problemgrößen begrenzt. Dies gilt insbesondere für den Spezialfall des VRPTW. Cordeau und Laporte (Cordeau und Laporte, 2007a) erwähnen ein exaktes Verfahren, welches auf die Problemgröße  $n \leq 15$  beschränkt ist. Durch Relaxation lässt sich selbstverständlich auf Kosten der Lösungsqualität die Problemgröße erhöhen, indem komplexe Restriktionen in Kostenfunktionen umgewandelt werden; pro Problemklasse existieren zur Lösung des Transportproblems jedoch auch leistungsfähige Metaheuristiken, deren Qualität sich dauernd verbessern.

Solomon präsentierte bereits 1987 (Solomon, 1987) Heuristiken für das VRPTW, die auf einer einheitlichen Fahrzeugflotte basieren, deren Größe nicht beschränkt ist.

Elementare Heuristiken zum Aufbau einer initialen Lösung sind hier:

- **Savings Heuristik** -  $n$  individuelle Touren bei  $n$  Kunden, die schrittweise zusammengefasst werden
- **Nearest Neighbor Heuristik** - iteratives Hinzufügen des zum Depot nächsten ungerouteten Kunden zu einer Tour
- **Insertion Heuristik** - eine bestehende Tour wird um einen ungerouteten Kunden an der kostenminimalen Position ergänzt
- **Sweep Heuristik** - ausgehend von einem in etwa mittigen Depot werden die Kunden gemäß einer Winkelbetrachtung in die Tour hinzugenommen

Durch klassische Heuristiken lassen sich bereits gute Lösungen implementieren, die laut Gendreau et al (Gendreau et al, 2002) oftmals im einstelligen Prozentbereich die optimalen Ergebnisse approximieren (Abbildung 3).

Basisheuristiken werden oft in Metaheuristiken zur Generierung von Startlösungen bzw. Lösungsoptimierung integriert - für das VRPTW listen Soo und Tay (Soo und Tay, 2009) Genetische-, Ant-, sowie Simulated Annealing-Algorithmen auf, teilweise aber auch Clustering-Methoden. Soo und Tay erwähnen beispielweise die Verwendung eines k-means Algorithmus zur expliziten Clustering benachbarter Kunden.

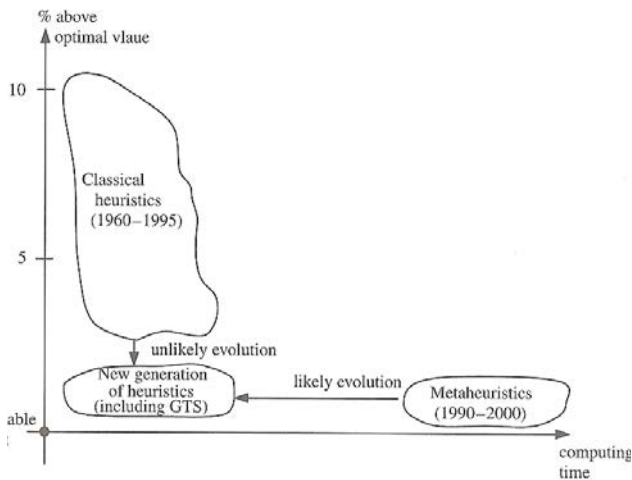


Abbildung 3: Lösungsqualität von Heuristikklassen (Gendreau, 2002)

Algorithmen die für das VRPTW ein gutes Ergebnis liefern, müssen dies nicht für das VRPPDTW tun, da hier die Fahrzeuge nicht am Depot, sondern schrittweise während der Tour beladen werden (Lau und Liang, 2001). Vorschläge für Algorithmen finden sich in Lau und Liang (2001) - es handelt sich um ein 2-Phasenverfahren (Initiale Lösung und Optimierung) mit Hilfe von elementaren Heuristiken wie "Insertion", "Sweep" sowie einem "Tabu-Search". Goel und Gruhn (Goel und Gruhn, 2006) zeigen und messen einen Algorithmus für das VRPDWH.

### BENCHMARKS FÜR TRANSPORTALGORITHMEN

Langstreckentransporte sind bezüglich der Abwicklung sehr komplex (bsp. Zollabwicklung und Wechsel der Transportmodus), die Route ist jedoch eher einfach - etwa ein Überseetransport, der umgeladen wird. Aus Sicht der Transportplanung ist der Kurzstreckentransport der komplexere Fall, da pro Tour mehrere Punkte anzufahren sind. Einen Vergleich der Transportarten zeigt Abbildung 4.

Bei Solomon (Solomon, 1987) werden drei Grundszenerien für Benchmarks für das VRPTW vorgestellt, die sich in der Art der Verteilung der Kunden sowie in der Breite der verwendeten Zeitfenster unterscheiden. Solomon unterscheidet zufällig verteilte (R\*), geclusterte (C\*) und gemischt zufällig/ geclusterte (RC\*) Szenarien. Jedes dieser drei Grundszenerien wird durch die Länge der zur Verfügung gestellten Zeitfenster unterteilt (die Lokation der Kunden ist pro Gruppe konstant). Er verwendet Szenarien mit eher langen Zeitfenstern (R2, C2, RC2) und mit kurzen Zeitfenstern (R1 C1, RC1), die nur ein Anfahren von wenigen (etwa 5-10) Kunden pro Tour erlauben.

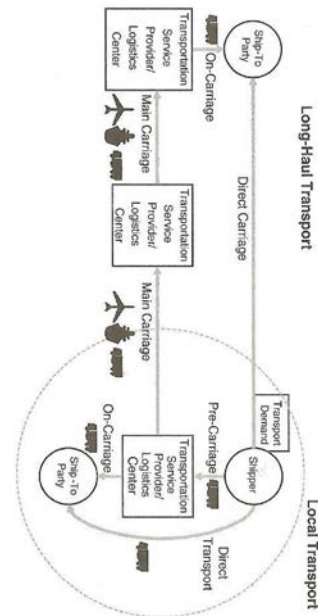


Abbildung 4: Lang- und Kurzstreckentransporte (aus Kappauf, Lauterbach, et. al. 2012)

Die Benchmarks werden des Weiteren bezüglich der Lage der Zeitfenster unterschieden, so dass etwa das R1-Szenario aus den Detailszenarien R101 bis R112 besteht. Kundenbedarfe gleichen sich in den Detailszenarien, das Koordinatensystem ist kartesisch. Die Zielfunktion minimiert hierarchisch die Fahrzeuge und dann die Distanz - ist jedoch detailliert nicht dokumentiert. Ursprünglich wurden 100 Kunden verwendet, es existieren jedoch auch Teilmengen, die nur die ersten 25 bzw. 50 Kunden verwenden. Aktuell beste Lösungen können beispielsweise bei Solomon (Solomon, 2005), Ausschnitte auch auf der Seite der Sintef (Sintef Transportation Optimization Portal, 2013) und bei Gambardella (Gambardella, 2013) nachgeschlagen werden.

Abbildung 5 zeigt das Depot („CUST NO 1“) sowie die ersten 4 Kunden. Abbildung 6 zeigt einen Plot der drei Grundszenerien von Solomon für 100 Kunden, die Transportkapazität beträgt 200 Einheiten.

R101						
CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE TIME
1	35.00	35.00	0.00	0.00	230.00	0.00
2	41.00	49.00	10.00	161.00	171.00	10.00
3	35.00	17.00	7.00	90.00	60.00	10.00
4	55.00	45.00	13.00	116.00	126.00	10.00
5	55.00	20.00	19.00	149.00	159.00	10.00

Abbildung 5: Ausschnitt des R101-Szenarios vom Solomon (2005) (Depot plus die ersten 4 Kunden)

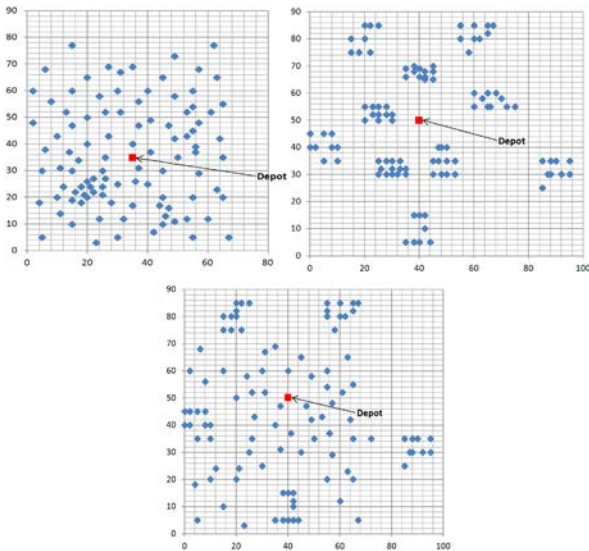


Abbildung 6: Plot des R-, C- und RC-Szenarios

Eine Dokumentation der Resultate zeigt der Ausschnitt der Seite von Solomon (Solomon, 2005) in Abbildung 7. Es wird hier für das Problem die Anzahl der Fahrzeuge, die Entfernung und die Autorenreferenzen angegeben.

Problem	NV	Distance	Authors	Problem	NV	Distance	Authors
R101.25	8	617.1	KDMSS	R201.25	4	463.3	CR+KLM
R101.50	12	1044.0	KDMSS	R201.50	6	791.9	CR+KLM
R101.100	20	1637.7	KDMSS	R201.100	8	1143.2	KLM

Abbildung 7: Beste Ergebnisse für R101 und R201

Lösungsdetails für einige Instanzen sind beispielsweise bei Gambardella (Gambardella, 2013) abrufbar. Im Ausschnitt von Abbildung 8 werden neben der Distanz (1292.68) die Anzahl der Fahrzeuge (8) sowie die Routen angegeben. Es ist gut zu erkennen, dass nur wenige Kunden angefahren werden können, da die Zeitfenster bei R1\* Szenarien sehr knapp bemessen sind.

```

r103
1292.68 13
0 60 45 83 5 99 6
0 71 65 78 34 35 81 77 28
0 2 22 75 56 4 25 54
0 7 19 11 8 46 47 48 82 18 89
0 94 96 95 97 87 13
0 27 69 30 9 66 20 51 1

```

Abbildung 8: Ausschnitt einer Lösung von Gambardella

Man kann erkennen, dass Solomon und Gambardella sich stark auf die Angabe der verwendeten Fahrzeuge und der Distanz konzentrieren. Gambardella gibt in anderen Übersichten zwar auch Laufzeiten des Optimierers an, jedoch scheint uns eine solche Angabe nicht notwendig, da eine Vergleichbarkeit nicht möglich ist; unterschiedliche Laufzeiten aufgrund von Hardware, Betriebssystem, Compiler sind selbst für den gleichen

Quellcode auf verschiedenen „ähnlichen“ Rechnern zu erwarten. Sinnvoll scheint nach unserer Einschätzung lediglich eine Angabe der Anzahl der verwendeten Lösungsschritte des Optimierungsalgorithmus.

Weitere Benchmarks sind: für das VRPTW-Szenario, der Benchmark von Gehring und Homberger, welcher Solomons Benchmark auf bis zu 1000 Kunden erweitert; Daten und Ergebnisse finden sich auf der Seite der Sintef (Sintef, 2013). Li & Lim haben einen Benchmark für den Fall von Pickup- und Delivery erstellt, der ebenfalls bis zu 1000 Kunden beinhaltet (Sintef, 2013). Es existieren zudem Benchmarks für das CVRP auf „Networking and Emerging Optimization“ (Neo, 2013).

Keine Benchmark, jedoch ein interessantes Beispiel zur Veranschaulichung elementarer Szenarien findet sich bei Grunow und Stefánsdóttir (Grunow und Stefánsdóttir, 2012) - anders als üblich werden nicht die Koordinaten, sondern nur die Distanzen vorgegeben. Abbildung 9 zeigt die Distanzmatrix, die Kundenbedarfe sowie Fahrzeugkapazitäten.

$c_{ij}$	Node j						
	Node i	0	1	2	3	4	5
0	0	60	76	74	78	141	117
1	60	0	26	24	75	122	81
2	76	26	0	11	82	120	74
3	74	24	11	0	82	121	75
4	78	75	82	82	0	75	69
5	141	122	120	121	75	0	62
6	117	81	74	75	69	62	0

	Customer i						Vehicle k	
	1	2	3	4	5	6	1	2
$d_i$	8	2	3	10	2	2	9	18
$q^k$								

Abbildung 9: Szenario von Grunow und Stefánsdóttir

In der Literatur wird meist mit Solomons Benchmark gearbeitet, hierzu wird der Benchmark dann oft auch entsprechend modifiziert, etwa in bei Lau und Liang (Lau und Liang, 2001) für das VRPPDTW oder bei Goel und Gruhn (Goel und Gruhn, 2006) für das VRPDWH.

#### ABBILDUNG VON TRANSPORTPLANUNGSPROBLEMEN AUF SAP-SCM-TPVS

TPVS setzt typischerweise auf einer Lieferung (Delivery), Kundenauftrag (Sales-Order) oder einer Umlagerungsbestellung (Stock-Transfer-Order) auf (siehe auch Dickersbach, 2009). Umlagerungsbestellungen können dabei intern in SAP-SCM durch einen Lauf der TLB-Planung erzeugt werden, Lieferungen direkt im SD-Modul des ERP-Systems.

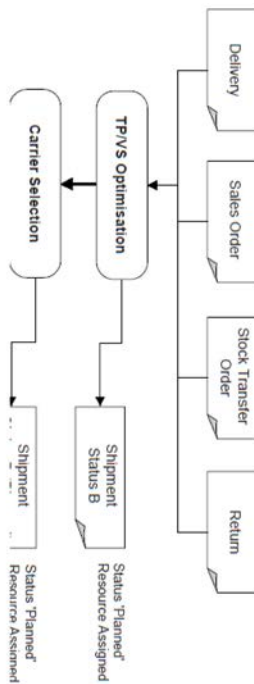


Abbildung 10: Einbettung von TPVS in SAP-SCM Planungsprozess laut Dickersbach, 2009

### Schritte des Transportplanungsprozesses

Die Transportplanung in TPVS kann innerhalb der VS-Optimierung Mengen nicht teilen! Um die Optimierungsergebnisse der Transportplanung nicht a priori fälschlich vorwegzunehmen, sind die Auftragsgrößen so zu wählen, dass der Optimierer wirklich einen Entscheidungsspielraum hat und das Fahrzeug nicht durch eine unteilbare Menge bereits signifikant ausgelastet ist und deshalb eine bestimmte Route nehmen muss. Um die Beladungsentscheidung dem Optimierer zu überlassen und um das Benchmarking effizient handhaben zu können, wird deshalb das folgende Verfahren vorgeschlagen:

1. Belegung der (Kunden)-Lokationen mit Bedarfen gemäß Benchmark, der (Depot)-Lokation mit ausreichend Dispobestand durch Anlage eines rechtzeitigen PPDS-Zugangs
2. Planung des Supply-Netzes mit Hilfe der **SNP-Netzwerkheuristik**, die Umlagerungsbestellanforderungen in Höhe der Kundenbedarfe vom Depot zum Kunden erzeugt. Danach erfolgt ein Deployment der SNP-Planung aus dem Depot heraus. Ein **Pull-Deployment** genügt hier, da ein Push-Deployment die Termine zu früh erzeugen würde. Wichtig ist hier, daß die Bedarfe zeitgerecht in genau der gewünschten Form gedeckt werden und das Deployment die entsprechenden ATD-Mengen berücksichtigt. Die erste Forderung impliziert für beide Schritte die Verwendung einer Heuristik, da ein SNP-Optimierungsprozess die Mengen-Terminbeziehung ändern könnte. Für die korrekte ATD-

Menge im Deploymentlauf genügt ein ausreichender Dispobestand im Depot/ Werk.

3. Erzeugung von minimalen Umlagerungsbestellungen durch TLB - diese können sich an kleinsten Transportmengen (etwa einer Palettenmengen von 1) orientieren. Hierzu ist in TLB für die Transportzusammenstellungsmethode eine Heuristik des Typs "rundungsbasiert" zu verwenden, eine Verwendung des Typs "kapazitätsbasiert" wäre falsch, da dann bereits eine Vorbelegung der Beladung stattfindet.
4. Planung der Transporte in TPVS - aufgrund von Schritt 3 werden minimale Aufträge auf den Fahrzeugen konsolidiert und die Bezugsquelle für die Kunden, die nicht direkt durch das Depot bedient werden, durch den vorgelagerten Kunden der Route ersetzt; die Entscheidung über die Beladung liegt so ausschließlich bei TPVS.

Darstellung der Anfangssituation (Schritt 1 und 2)

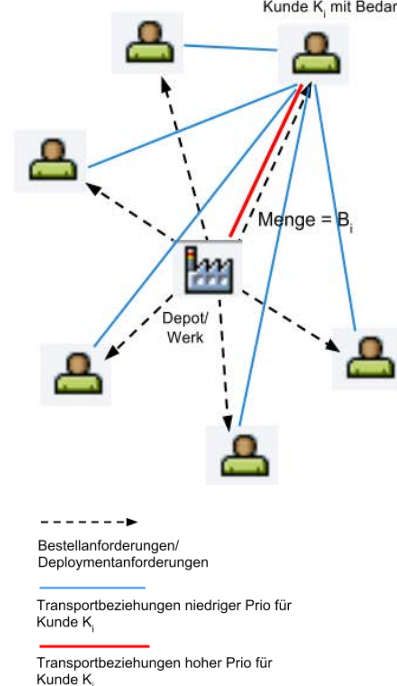


Abbildung 11: Supply-Netz in Anfangszustand

Abbildung 11 zeigt das Supply-Netz nach den ersten beiden Schritten. Wichtig ist hier, dass Transportbeziehungen verschiedener Priorität vorliegen, so dass die SNP-Heuristik die Umlagerungsbestellanforderungen in Höhe des Transportbedarfs direkt vom Depot anlegen kann, welche durch ein Deployment gedeckt werden können.

Darstellung der Situation nach TLB (Schritt 3)

Abbildung 12 zeigt das Supply-Netz nach der Rundung der Mengen auf Palettengrößen im Rahmen des Transport-Load-Buildings. Es wurden einzelne Umlagerungs-

bestellungen erzeugt, die in Summe den Umlagerungsbestellanforderungen entsprechen.

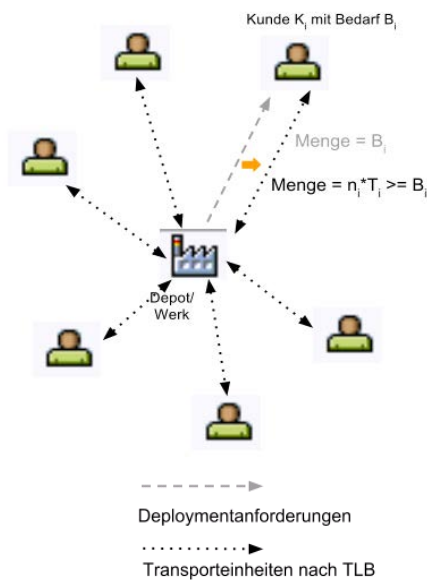


Abbildung 12: Supply-Netz nach dem Transport-Load-Building

Darstellung der Situation nach TPVS-Optimierung (Schritt 4)

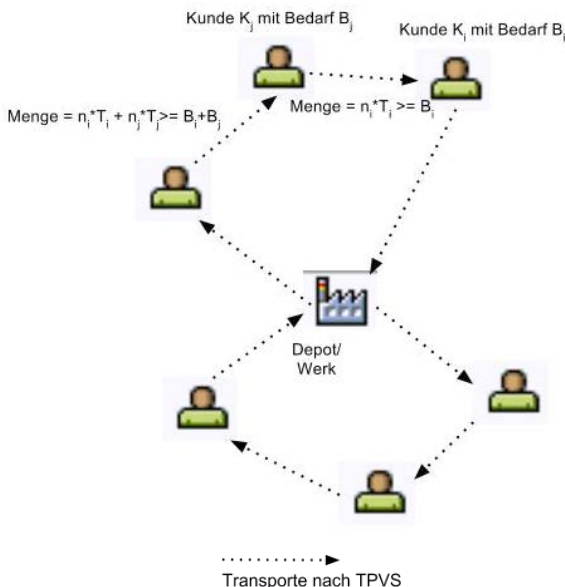


Abbildung 13: Netz nach TPVS

Abbildung 13 zeigt das Supply-Netz nach dem TPVS. Die wesentliche Änderung zu Abbildung 12 besteht darin, dass den Umlagerungsbestellungen neue Bezugsquellen zugeordnet wurden. Es findet i.d.R. kein Bezug von der Lokation, sondern von einem benachbarten Kundenknoten statt.

### Parameter für das VRP (hier ohne Restriktionen)

Die Daten von Solomons Benchmark bilden oft die Basis von Benchmarkdatensätzen für andere Probleme als das VRPTW. Für einige Transportprobleme soll deshalb hier angegeben werden, wie die passenden Stammdaten sowie Optimierungsparameter zu pflegen sind, um eine Struktur ähnlich der Benchmarks von Solomon abzubilden.

### Stamm-/ Bewegungsdaten zur Abbildung des CVRP & DCVRP

Ohne Verwendung eines Geoinformationssystems (GIS) sind die Entfernungen zwischen den Lokationen euklidisch; es kann somit die Vorgabe aus dem Benchmark direkt in die Längen- und Breitengrade der Lokation übernommen werden. Hierzu kann das Szenario im Koordinatensystem auch verschoben werden, etwa um nur die Gradminuten ausfüllen zu müssen. Jede Lokation kann durch eine Massenanlage (TXN: /SAPAPO/SCC\_TL2) der Transportbeziehungen mit jeder der  $n$  übrigen Lokationen automatisiert verbunden werden. Um die Distanzen explizit ausgewiesen zu bekommen und insbesondere um verschiedene Prioritäten den Transportbeziehungen zuweisen zu können, wurde auf die Verwendung einer Transportzone verzichtet. Eine Verbindung einer Lokation mit sich selbst ist nicht notwendig. Es werden somit  $n^2$  Transportbeziehungen benötigt.

Die Distanzen können automatisiert bei Massenanlage berechnet werden, die Servicezeit kann in die Transportbeziehung kodiert werden. Die Priorität zum Depot kann über ein Massenändern (TXN: MASSD) erhöht werden.

Ein Beispiel der Definition der Kundenlokation 2 zeigt Abbildung 14. In diesem Fall wird das Depot durch den Kunden Nr. 1 der Abbildung 5 repräsentiert. Es wurde von der Koordinate (35,35) nach 2/2 (Minuten) verschoben. Die Koordinate 41/49 des Kunden Nr. 2 wird dann konsequenterweise in 8/16 (Minuten) abgebildet.

Identifizier		Externe Lokationsbez.	
GLN	0	Ext.Lokation	SOL
DUNS+4		Bwl.Syst.Verb.	BSG
Geographische Daten			
	Vz.	Grad	Minuten
Längengrad	+	0	8
Breitengrad	+	0	16
Zeitzone	CET	Präz.	300

Abbildung 14: Koordinaten Kunde 2

Die Transportbeziehung weist dann die euklidische Distanz in Seemeilen zwischen beiden Knoten aus; dies gilt auch, wenn der Kunde nicht bei Längen-/ Breitengrad (0,0) liegt.

Produkte	genau 1 Produkt
Lokationen	1 Depot/ n Kunden

<b>Transport-beziehung</b>	genau n <sup>2</sup> / Beziehung zum Depot mit höherer Priorität/ Generierung der Distanz aus Koordinaten der Benchmark/ Verwendung der Verweilzeit für Servicezeit
<b>Zugang</b>	mind 1 (im Depot)/ kann über einen rechtzeitigen PPDS-Zugang angelegt werden/ die Summe der Mengen vor Bedarfstermin muss den Vorgaben entsprechen
<b>Bedarfe</b>	genau n/ einen in jeder Kundenlokation (bei VRPB in Depotlokation)
<b>Fahrzeuge</b>	mind 1 (mit Kaparestriktionen)/ Kennzeichnen des Fahrzeugs als eigenes Transportmittel (im Transportmodus) und Zuweisung zum Depot.

Tabelle 1: Stamm-/ Bewegungsdaten CVRP/ DCVRP

### Optimiererparameter zur Abbildung des CVRP/ DCVRP

Die Kostenfunktionen sind in den Benchmarks nicht exakt spezifiziert; um eine Vergleichbarkeit des Ergebnisses zu erzielen, sollte die Anzahl der Fahrzeuge auf die im Benchmark festgelegte Anzahl fixiert werden; eine verspätete Anlieferung/ Abholung sollte verboten sein. Dies kann außerhalb der Standard-Optimiererparameter über die Definition der Zeitfenster für Abholung/ Anlieferung (/SAPAO/VS11) geschehen. Die Fixkosten pro Fahrzeug können dann so lange modifiziert werden, dass die gewünschte Anzahl von Fahrzeugen verwendet wird – Vergleichskosten sind die Kosten für Verführung und die Kosten für die Transportdistanz. Durch Definition der Verspätung als harte Restriktion in der Definition der Zeitfenster für die Transportoptimierung vereinfacht sich die Wahl der Parameter, da nicht mit den Verspätungskosten experimentiert werden muss; sondern einfach die fixen Kosten variiert werden können, bis die gewünschte Anzahl von Fahrzeugen gewählt wurde.

<b>Verfrühte/ Versp-Anlieferung oder Abholung</b>	Ca.10/ harte Restriktion
<b>Nichtlieferung</b>	>=100000
<b>Fixkosten pro Transport</b>	Variieren (ca. 10 – 1000), bis gewünschte #FZ verwendet ( <b>dies ist die einzige Variable</b> )
<b>Transportkosten pro KM</b>	1

Tabelle 2: Optimiererparameter CVRP/ DCVRP

Zur Abbildung des DCVRP kann die maximale Transportdistanz im Optimiererprofil vorgegeben werden. Typischerweise beschreibt das DCVRP eine maximale Transportdistanz, es kann aber auch eine maximale Zeitdauer vorgegeben werden.

<b>Max Zeitdauer/ Distanz, Stopoffs</b>	Angabe des Transportmaximums pro Kostendimension
---	--

Tabelle 3: Zusätzlicher Parameter für DCVRP

### Stammdaten zur Abbildung des VRPTW

Zur Beschreibung des VRPTW ist das CVRP um Zeitfenster an den Lokationen zu erweitern. Dies kann durch Lokationsressourcen geschehen. Die Optimiererparameter des CVRP sind ausreichend.

<b>Lokations-Ressourcen für Öffnungszeiten</b>	Genau. n Inbound (für Kunden); 1 Outbound für Depot
--	---

Tabelle 4: Zusätzlicher Parameter für VRPTW

### Stammdaten zur Abbildung des VRPB

Wird eine einfache Form des VRPB gewählt, in der lediglich eine Reihe von Kunden angefahren werden muss, bei denen ein Produkt abgeholt wird, um es in das Depot zu transportieren, sind Zugänge bei den Kunden und ein Bedarf im Depot anzulegen. Soll bei einigen Kunden vorher eine Auslieferung stattfinden und erst danach bei anderen Kunden abgeholt werden, entspricht dies der Abbildung einer „Beladereihenfolge“, welches durch TPVS nicht unterstützt wird.

Ist eine Mischung zwischen Lieferung und Abholung erlaubt, und sollen die abzuholenden Produkte in das Depot zurück geliefert werden (und nicht zu einem anderen Kunden), sind die Daten des CVP wie folgt zu ändern.

<b>Produkte</b>	mind 2 Produkte (genaugenommen Pegging-IDs) (A und B)
<b>Zugang</b>	je 1 Zugang für Produkt A pro Kunde, wenn das Produkt abgeholt werden muss/ genau 1 Zugang für Produkt B in Depotlokation
<b>Bedarfe</b>	je 1 Bedarf pro Kunde pro Produkt B, wenn geliefert werden muss/ genau 1 Zugang für Produkt A in Depotlokation

Tabelle 5: Parameteränderung für VRPB

Zur Abbildung des VRPBTW sind wieder die Zeitfensterrestriktionen zu den Lokationen hinzuzunehmen.

### Abbildung des VRPPD und VRPDWH

Zur Abbildung des VRPPD werden entsprechend mehr Produkte benötigt, um pro Kundenlokation Bedarfe und Zugänge unterscheiden zu können. Zudem besteht hier wieder das Problem der Berücksichtigung der Beladereihenfolge.

Es konnte bisher keine Abbildung der Regeln zu Pausenzeiten gefunden werden – es existieren lediglich Ressourcenmodelle der Fahrzeuge.

### BEISPIELE VON PLANUNGSRESULTATEN IN SAP-SCM-TPVS

Es soll mit einer a-priori definierten Anzahl an Fahrzeugen gearbeitet werden, andernfalls sind die Ergebnisse nicht mit Solomons Werten vergleichbar - die Wahl der Anzahl der Fahrzeugen hängt mit dem oben vorgeschlagenen Verfahren nur von den Fixkosten pro Fahrzeug ab.

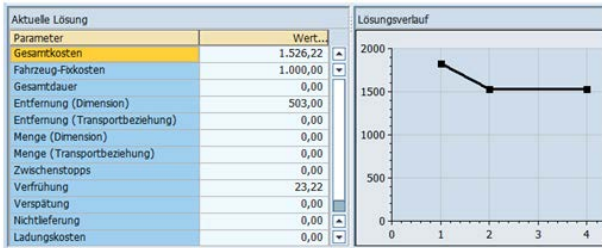
Beispielhaft sollen hier lediglich die Ergebnisse gemäß den Vorgaben von Grunow, Stefánsdóttir (2012) dokumentiert werden, um die prinzipielle Korrektheit des Vorgehens zu dokumentieren:

TranspAbs	Quellokation	Ziellokation	AktBeginn	Uhrzeit	AktEnde	Uhrzeit	Entfernung	Nettodauer
1	FD_CVRP_PP_0	FD_CVRP_DC_6	31.05.2013	08:05:00	31.05.2013	10:02:00	117	1:57
2	FD_CVRP_DC_6	FD_CVRP_DC_5		10:02:00	31.05.2013	11:04:00	62	1:02
3	FD_CVRP_DC_5	FD_CVRP_DC_4		11:04:00	31.05.2013	12:19:00	75	1:15
4	FD_CVRP_DC_4	FD_CVRP_PP_0		12:19:00	31.05.2013	13:37:00	78	1:18
5	FD_CVRP_PP_0	FD_CVRP_DC_1		13:37:00	31.05.2013	14:37:00	60	1:00
6	FD_CVRP_DC_1	FD_CVRP_DC_2		14:37:00	31.05.2013	15:03:00	26	:26
7	FD_CVRP_DC_2	FD_CVRP_DC_3		15:03:00	31.05.2013	15:14:00	11	:11
8	FD_CVRP_DC_3	FD_CVRP_PP_0		15:14:00	31.05.2013	16:28:00	74	1:14

Abbildung 15: Tour im CVRP

Abbildung 15 zeigt eine Tour im CVRP auf Basis der Beispieldaten von Abbildung 9. Man erkennt gut die Rückfahrt zum Depot (FD\_CVRP\_PP\_0) während der Tour.

Abbildung 16: Ausschnitt des Kostenverlaufs



Im Kostenverlauf lässt sich gut erkennen, dass die Metaheuristik nur sehr wenige Iterationsschritte benötigt. Offensichtlich werden leistungsfähige Heuristiken verwendet, die die Anzahl der Iterationen gering halten.

Die Anzahl der Fahrzeug wird experimentell durch Variation der Fixkosten bestimmt– es interessiert schließlich nur die Qualität der Route für eine bestimmte Anzahl von Fahrzeugen.

Transportnummer	TranspAbs	Quellokation	Ziellokation	AktBeginn	Uhrzeit	AktEnde	Uhrzeit	Entfernung
\$0000028174	1	FD_VRPTW_PP_0	FD_VRPTW_DC_4	31.05.2013	12:42:00	31.05.2013	14:00:00	78
Leerfahrt	2	FD_VRPTW_DC_4	FD_VRPTW_PP_0		14:00:00	31.05.2013	15:18:00	78

Abbildung 17: Touren im VRPTW

Ein Beispiel für die Umsetzung des VRPTW findet sich in Abbildung 17. Aufgrund der engen Zeitfenster und der harten Verspätungsrestriktion hat der Optimierer beide Fahrzeuge gewählt und ist zu früh gestartet – hier werden eigentlich drei Fahrzeuge benötigt. Abbildung 18 zeigt beispielhaft die Ressourcenbelegung in der Feinplanungssicht.

Ressource	RessAr	Überis	09.05.2013	30.05.2013	31.05.2013
FD_VRPTW_VHCL_1_A	T	000 %			
FD_VRPTW_VHCL_2_A	T	000 %			
INBOUND_VRPTW_DC_2	H	000 %			
INBOUND_VRPTW_DC_6	H	000 %			
INBOUND_VRPTW_DC_3	H	000 %			
INBOUND_VRPTW_DC_5	H	000 %			
INBOUND_VRPTW_DC_4	H	000 %			
OUTBOUND_VRPTW_PP_0	H	000 %			
INBOUND_VRPTW_DC_1	H	000 %			

Abbildung 18: Detaillierte Ressourcenbelegung bei VRPTW

Es lässt sich gut in Abbildung 18 erkennen, wie die Aktivitäten sich an den Zeitfenstern der Fahrzeuge ausrichten.

Ein einfaches VRPB-Szenario zeigt Bild 19, es lässt sich erkennen, dass das Fahrzeug leer am Depot losfährt, Schrittweise bei den Kunden Ladung aufnimmt und gegebenenfalls zum Depot zurückkehrt, um zu entladen.

Transportabschnitt	TranspNr	Quelllok	Ziellok	Ressource
FD_VRPB_PP_0 > FD_VI Leerfahrt		FD_VRPB_PP_0	FD_VRPB_DC_3	FD_VRPB_VHCL_1
FD_VRPB_DC_3 > FD_V \$0000031044		FD_VRPB_DC_3	FD_VRPB_DC_2	FD_VRPB_VHCL_1
FD_VRPB_VHCL_1				FD_VRPB_VHCL_1
000000180153		FD_VRPB_DC_3	FD_VRPB_PP_0	
000000180155		FD_VRPB_DC_3	FD_VRPB_PP_0	
FD_VRPB_DC_2 > FD_V \$0000031044		FD_VRPB_DC_2	FD_VRPB_DC_6	FD_VRPB_VHCL_1
FD_VRPB_VHCL_1				FD_VRPB_VHCL_1
000000180151		FD_VRPB_DC_2	FD_VRPB_PP_0	
000000180152		FD_VRPB_DC_2	FD_VRPB_PP_0	
000000180153		FD_VRPB_DC_3	FD_VRPB_PP_0	
000000180155		FD_VRPB_DC_3	FD_VRPB_PP_0	
FD_VRPB_DC_6 > FD_V \$0000031044		FD_VRPB_DC_6	FD_VRPB_DC_5	FD_VRPB_VHCL_1
FD_VRPB_VHCL_1				FD_VRPB_VHCL_1
000000180151		FD_VRPB_DC_5	FD_VRPB_PP_0	
000000180152		FD_VRPB_DC_2	FD_VRPB_PP_0	
000000180168		FD_VRPB_DC_2	FD_VRPB_PP_0	
000000180169		FD_VRPB_DC_6	FD_VRPB_PP_0	
000000180153		FD_VRPB_DC_3	FD_VRPB_PP_0	
000000180155		FD_VRPB_DC_3	FD_VRPB_PP_0	
FD_VRPB_DC_5 > FD_V \$0000031044		FD_VRPB_DC_5	FD_VRPB_DC_4	FD_VRPB_VHCL_1
FD_VRPB_DC_4 > FD_V \$0000031044		FD_VRPB_DC_4	FD_VRPB_PP_0	FD_VRPB_VHCL_1
FD_VRPB_PP_0 > FD_VI Leerfahrt		FD_VRPB_PP_0	FD_VRPB_DC_4	FD_VRPB_VHCL_1
FD_VRPB_DC_4 > FD_VI \$0000031044		FD_VRPB_DC_4	FD_VRPB_PP_0	FD_VRPB_VHCL_1
FD_VRPB_PP_0 > FD_VI Leerfahrt		FD_VRPB_PP_0	FD_VRPB_DC_3	FD_VRPB_VHCL_1
FD_VRPB_DC_3 > FD_V \$0000031044		FD_VRPB_DC_3	FD_VRPB_DC_1	FD_VRPB_VHCL_1
FD_VRPB_DC_1 > FD_V \$0000031044		FD_VRPB_DC_1	FD_VRPB_PP_0	FD_VRPB_VHCL_1

Abbildung 19: Beladungen bei Kunden im VRPB

## ZUSAMMENFASSUNG UND AUSBLICK

Es wurden wesentliche Transportprobleme sowie deren Benchmarks betrachtet und vorgestellt, wie diese Probleme in TPVS abgebildet werden können.

In einem weiteren Schritt sollen die wesentlichen Benchmarks in TPVS mit Massendaten durchgeführt werden, sowie Experimente mit eigenen (Meta-)Heuristiken in ERP durchgeführt werden – insbesondere im Hinblick auf die Lösung der VRPB und VRPPD-Problemlklassen und des VRPDWH.

## LITERATUR

- Cordeau, Laporte, et. al. "Vehicle Routing", aus Barnhart, Laporte (Eds.), Handb. in Operations Research & Management Science, Vol. 14, Chapter 6, Elsevier 2007
- Cordeau, Laporte, et. al. "Transportation on Demand", aus Barnhart, Laporte (Eds.), Handb. in Operations Research & Management Science, Vol. 14, Chapter 7, Elsevier 2007
- Dickersbach, "Supply Chain Management with SAP APO: structures, modelling approaches and implementation of SAP SCM 2008", Springer 2009
- Gambardella auf Seite <http://www.idsia.ch/~luca/macsvrptw/solutions/welcome.htm>, abgerufen am 8. Mai 2013
- Gendreau, Laporte, Potvin, „Metaheuristics for the Capacited VRP“, aus Toth, Vigo (Ed.) The Vehicle Routing Problem, Siam, 2002
- Goel, Gruhn, "Drivers' working hours in vehicle routing and scheduling", Pro-ceedings of the IEEE ITSC, 2006
- Grunow, Stefánsdóttir, "Transportation Planning/ Vehicle Scheduling", aus H. Stadtler, B. Fleischmann, et.al, Advanced Planning in Supply Chains, Springer 2012
- Kappauf, Lauterbach, Koch, "Logistics Core Operations with SAP - Inventory Management, Warehousing, Transportation and Compliance", Springer 2012
- Ketterer, „Transportplanungsprobleme und integrierte Anwendungen des Supply Chain Managements“, Tagungsband der AKWI-Tagung 2013,



Herausforderungen an die Wirtschaftsinformatik: Integration und Konnexion, Verlag News& Media, 2013

Lau, Liang, "Pickup and Delivery with Time Windows: Algorithms and Test Case Generation", Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence, 2001

Networking and Emerging Optimization (Neo), <http://neo.lcc.uma.es/vrp/vehicle-routing-problem/>, letztes Update 7. Januar 2013, abgerufen am 8. Mai 2013

Sintef Transportation Optimization Portal, <http://www.sintef.no/Projectweb/TOP/>, letzte Pflege Januar 2013, Zugriff am 24. April 2013

Solomon, "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints", Operations Research, 1987

Solomon auf Seite <http://web.cba.neu.edu/~msolomon/problems.htm>, zuletzt gepflegt 2005, Zugriff am 24. April 2013

Soo, Tay, "A Survey on the Progress of Research on Vehicle Routing Problem With Time Windows Constraints", Symposium on Progress in Information & Communication Technology, 2009

Toth, Vigo, "An Overview of Vehicle Routing Problems" aus Toth, Vigo (Ed.) The Vehicle Routing Problem, Siam, 2002

#### **KONTAKT**

Norbert Ketterer lehrt seit Ende 2008 Wirtschaftsinformatik an der Hochschule Fulda. Sein Hauptinteresse liegt in dem Bereich „Betriebliche Anwendungssysteme“, insbesondere der Unterstützung betrieblicher Geschäftsprozesse durch diese.

# EIN ANSATZ ZUR VERTIKALEN DATENPARTITIONIERUNG IN DER CLOUD

Patrick Müller, Jens Kohler, Thomas Specht  
Hochschule Mannheim

Paul-Wittsack-Straße 10, Mannheim, 68163, Deutschland  
E-Mail: patrick@4-haen.de, j.kohler@hs-mannheim.de, t.specht@hs-mannheim.de

## STICHWORTE

Cloud, vertikale Datenpartitionierung, verteilter JOIN

## KURZFASSUNG

Cloud Computing als eines der aktuellen Hype-Themen erfährt durch aktuelle Datenschutzskandale, Spionageversuche und Sicherheitslücken enormen Gegenwind. Nicht nur unter Privatanwendern, sondern auch im Unternehmensbereich erhöht sich die Skepsis Dienste und Anwendungen aus der Cloud zu nutzen. Auf der anderen Seite verspricht die dynamische Skalierbarkeit von Rechnerressourcen, wie z. B. Speicherplatz, Arbeitsspeicher und Prozessorleistung und deren minuten- bzw. stundengenaue Abrechnung enorme Kostenvorteile. Diese Arbeit zeigt daher ein Konzept zur verteilten Datenhaltung, bei dem Daten logisch auf verschiedene Clouds aufgeteilt werden, sodass bei jedem Provider nur ein Teil liegt, der ohne die anderen Teile wertlos ist. Die Aufteilung bewegt sich auf Datenbankebene. Dabei wird der Ansatz der vertikalen Partitionierung verwendet, um die Daten logisch aufzuteilen. Die Arbeit ist im Rahmen des SeDiCo-Projekts an der Hochschule Mannheim entstanden. Es wurde dabei nicht nur das Konzept erarbeitet, sondern auch die technische Machbarkeit mit einer prototypischen Implementierung nachgewiesen.

## EINLEITUNG

Cloud Computing ist aktuell eines der am meisten beachteten Themen in der Weiterentwicklung der IT-Landschaft. Endverbraucher wie auch Firmen sehen eine schier unglaubliche Explosion an Cloud Computing-Anwendungen. Als Beispiele seien hier exemplarisch für „Software as a Service“ (SaaS) Salesforce, für „Platform as a Service“ (PaaS) Google AppEngine und für „Infrastructure as a Service“ IaaS Amazon EC2 genannt. Demgegenüber stehen wachsende Bedenken im Bereich Datenschutz und Datensicherheit aufgrund regelmäßig auftretender Sicherheitslücken und Spionageversuche staatlicher sowie privater Organisationen.

Um diesen Bedenken entgegenzutreten, entwickelt das von der MFG Stiftung geförderte Forschungsprojekt SeDiCo ein Framework zur sicheren und verteilten Datenspeicherung. SeDiCo steht dabei für „A Secure and Distributed Cloud Datastore“. Zur Erhöhung der Sicherheit wird im Rahmen dieses Projektes auf Datenbankebene ein Ansatz für die vertikale Partitionierung

von Daten bei verschiedenen Cloud-Anbietern entwickelt. Sollte es einem Angreifer gelingen, eine der Partitionen zu kompromittieren, kann er damit nichts anfangen, da er nur einen Bruchteil der Daten hat.

Es ist wichtig die Verteilung der Daten nach Datenschutzaspekten vorzunehmen, damit beispielsweise Kreditkartennummer und Name eines Kunden in jedem Falle in zwei unterschiedlichen Partitionen liegen. Ideen zur sicheren Aufteilung der Daten werden ebenfalls von anderen Projekten verfolgt, vgl. (Ganapathy et al. 2011; Plattner 2013).

Durch eine Abstraktion verschiedener Cloud-Plattformen und Datenbankanbieter soll die Abhängigkeit von bestimmten Providern, der sogenannte „Vendor lock-in“, vermieden werden. Je leichter ein Wechsel zwischen verschiedenen Cloud-Diensten realisiert werden kann, desto häufiger wird diese Möglichkeit von den Benutzern der Dienste voraussichtlich auch genutzt.

Das zentrale Element des SeDiCo-Projektes ist ein Verteilungsalgorithmus zur vertikalen Partitionierung. Dieser wird benötigt, um Datensätze auf unterschiedliche Cloud-Anbieter zu verteilen. Ein solcher Verteilungsalgorithmus wird im Rahmen dieses Artikels von der Konzeption bis zur Implementierung aufgezeigt und schließlich bewertet.

## MOTIVATION

Bisherige Ansätze implementieren vor allem die horizontale Verteilung (s. a. NoSQL) wegen der erhöhten Skalierbarkeit und der besseren Performance bei der dynamischen Erweiterung der zugrundeliegenden Cloud-Ressourcen (scale-up und scale-out). Plattner zeigt hier verschiedene Möglichkeiten der horizontalen Partitionierung (Plattner 2013). U. a. werden dabei komplette Datensätze anhand ihrer Eigenschaften auf verschiedene Ressourcen verteilt (range partitioning). So wäre es denkbar, dass eine Tabelle „Kunden“ nach dem Alter der Kunden partitioniert wird. Kunden zwischen 18 und 30 Jahren würden in eine Partition ausgelagert. Eine zweite Partition würde Kunden zwischen 31 und 50 Jahren und die letzte Partition würde alle Kunden, die älter als 51 Jahre sind, beinhalten. Ein anderer Ansatz, losgelöst von den Eigenschaften der Daten, wäre die Daten nach einem bestimmten Algorithmus, z. B. round robin oder gar zufällig in verschiedene Partitionen zu verteilen. Für weitere horizontale Verteilungs-

strategien sei an dieser Stelle auf die Literatur verwiesen (Plattner 2013).

Unabhängig von der Partitionierungsstrategie haben die horizontale und die vertikale Partitionierung als gemeinsames Ziel die Aufteilung der Daten in disjunkte Teilmengen. Weiterhin müssen beim Lesen der Daten geeignete Join-Operationen implementiert sein, um die verteilten Daten wieder zusammenzuführen.

Im Gegensatz zur horizontalen trennt die vertikale Partitionierung dazu einzelne Datensätze (Tupel) logisch auf und dupliziert deren Primärschlüssel in alle Partitionen. Anhand dieser Schlüssel lassen sich die Daten anschließend auch wieder zusammenfügen. Mit Blick auf die Themen „Cloud Computing“ und „Big Data“ sind die zwei Faktoren Performance und Sicherheit die wichtigsten Entscheidungskriterien für oder gegen einen Ansatz. Gerade in der logischen Trennung der Tupel liegt die Motivation des SeDiCo-Projekts mit der Konzeption und Implementierung eines sicheren Datenspeichers in der Cloud, begründet.

Im Projekt wird die technische Machbarkeit des Ansatzes nachgewiesen. Weitere Analysen bezüglich der Performance, insbesondere beim Join der Daten, würden allerdings den Rahmen des Projekts sprengen. Es ist aber geplant diese Fragestellung in einem Folgeprojekt anzugehen.

## PROBLEMSTELLUNG

Die Betrachtung der Problemstellung verläuft zweistufig. In der ersten Stufe werden verschiedene Ansätze zur Verteilung der Daten betrachtet. In der zweiten Stufe (Lösungsansatz) wird analysiert, wie sich die partitionierten Daten wieder zusammenfügen lassen. Dies ist insbesondere für bestehende Anwendungen (nachfolgend „Legacy Systeme“) ein zentraler Punkt, da SeDiCo den Anspruch erhebt, anwendbar zu sein, ohne Änderungen an bestehenden Systemen durchführen zu müssen. Die prototypische Implementierung fokussiert sich hierbei auf komponentenbasierte Java-Anwendungen.

## Vertikale Partitionierung

Für die Implementierung des Verteilungsalgorithmus kommen die fünf in Abbildung 1 gezeigten Schichten infrage.

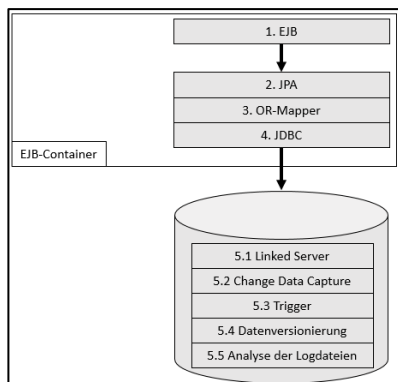


Abbildung 1: Schichten zur Verteilung

## Schicht 1: Enterprise JavaBeans (EJBs)

In der ersten Schicht besteht die Möglichkeit EJBs so zu annotieren, dass die jeweiligen Properties in verschiedene Datenbanken geschrieben werden, um damit die Partitionierung zu realisieren. Dabei wäre jede Entitätsklasse in je eine Klasse pro Partition aufzuteilen. Der Vorteil dieses Ansatzes besteht vor allem in seiner Einfachheit. Damit einher geht jedoch der Nachteil, dass eine bestehende Anwendung stark angepasst werden müsste. Zusätzlich ist dieser Ansatz für einen Anwendungsprogrammierer nicht transparent ist, da die gesplitteten Klassen bei der Entwicklung immer sichtbar blieben.

## Schicht 2: Java Persistence API (JPA)

Als zweite Variante wäre es möglich, die JPA so zu erweitern, dass die Daten vor dem Schreibvorgang in die Datenbank abgefangen und partitioniert werden. JPA-Implementierungen besitzen bereits einen SQL-Parser, daher ist davon auszugehen, dass eine Implementierung mit diesem Ansatz einen deutlich geringeren Aufwand bedeuten würde. Nachteilig ist jedoch, dass alle Zugriffe auf die Daten durch JPA erfolgen müssten. Zudem wäre die Nutzung von erweiterten Datenbankfunktionalitäten wie z. B. Stored Procedures oder Trigger mit dieser Lösung nicht möglich.

## Schicht 3: Object-Relational Mapper (OR-Mapper)

Anstatt einer JPA-Erweiterung wird in Schicht 3 eine OR-Mapper-Erweiterung in Betracht gezogen. Der Ansatz bliebe praktisch derselbe, es könnten jedoch die spezifischen Programmierschnittstellen des gewählten OR-Mappers genutzt werden. Da bei dieser Lösung der Ansatz mit dem Ansatz aus Schicht 2 vergleichbar ist, sind die Vor- und Nachteile ebenso dieselben.

## Schicht 4: JDBC-Treiber

Eine weitere Möglichkeit besteht darin, vorhandene JDBC-Treiber zu kapseln. Aufrufe an JDBC müssten dazu partitioniert und dann an die eigentlichen Treiber durchgereicht werden. Da die JDBC-Treiber-Implementierungen und die genutzten SQL-Dialekte verschiedener Hersteller unterschiedlich sind, müsste eine solche Implementierung jedoch für jeden SQL-Hersteller speziell angepasst werden, was auf einen sehr hohen Implementierungsaufwand hinauslaufen würde. Zudem müsste diese zusätzliche Schicht laufend Änderungen an den JDBC-Treibern berücksichtigen, was auf lange Sicht nicht praktikabel zu gewährleisten wäre.

## Schicht 5.1: Linked Server

Ein Datenbankfeature ist das sog. „Linked Server“. Damit ist die Zusammenschaltung mehrerer Datenbanken gemeint, die dann in einem einzigen Query angesprochen werden können. Dieser Ansatz funktioniert allerdings nur innerhalb der jeweiligen Datenbanken und nicht datenbankübergreifend. Aus diesem Grund wird auch dieser Ansatz nicht weiter verfolgt.

Des Weiteren gibt es bei der Betrachtung der Datenbankebene vier weitere Ansätze, die sich, im Gegensatz

zum „Linked Server“-Feature, auf die Nachverfolgung von Änderungszugriffen beziehen. Diese werden nun nachfolgend kurz angerissen.

### Schicht 5.2: Change Data Capture (CDC)

Einige Datenbanken besitzen eine eingebaute Funktionalität namens „CDC“. Dies ist eine Methode, um Änderungen an Tabellen festzustellen. Allerdings unterstützen nicht alle Datenbanken, insbesondere nicht die freien Datenbanken, diese Funktionalität. Daher wird diese Lösung im Folgenden nicht weiter betrachtet.

### Schicht 5.3: Trigger

Eine weitere Möglichkeit ist es, die zu überwachende Tabelle mit je einem Trigger für Inserts, Updates und Deletes zu versehen. Im Trigger müsste dann der Primärschlüssel der geänderten Zeile in eine spezielle Trackingtabelle übernommen werden. Dazu kommt ein Timestamp, um die korrekte Reihenfolge der Änderungen sicherzustellen. Ein externer Prozess könnte in einem beliebigen Intervall die Tracking-Tabelle abfragen, die entsprechenden Zeilen aus der Quelltable laden und auf die Zielsever verteilen. Nachteilig ist bei diesem Ansatz, dass nicht jede (freie) Datenbank Trigger unterstützt und dass die Änderungen nur zeitverzögert übernommen werden, was zu einer „eventuell konsistenten“ Datenbank führt, die nur noch die BASE-Bedingungen erfüllt (Pritchett 2008).

### Schicht 5.4: Datenversionierung

Das Hinzufügen einer Spalte an die Quelltable wäre eine weitere mögliche Herangehensweise. Diese Spalte müsste entweder eine Ganzzahl enthalten, die mit jeder Änderung hochgezählt wird, oder einen Zeitstempel (Plattner 2013). Dieser Wert aus der Quelltable könnte dann ebenfalls regelmäßig mit der Zieltabelle abgeglichen werden. Der Nachteil ist in diesem Fall erneut der Verlust der ACID-Kriterien.

### Schicht 5.5: Analyse der Logdateien

Als letzte Möglichkeit könnten die binären Logdateien eines SQL-Servers ausgelesen werden. Alle Änderungen müssten dazu aus der Logdatei extrahiert und auf den Zielsevern erneut ausgeführt werden. Diesen Mechanismus verwenden einige Datenbanken zur Master-Slave-Replikation. Selbst Informationen wie z.B. Auto-Increment IDs finden sich in diesem Log wieder und könnten auf einem zweiten Server ausgeführt werden. Eine solche Implementierung müsste auf jeden Fall für jede Datenbank neu angepasst werden, da die exakte Form des Logging nicht standardisiert ist und potenziell mit jeder neuen Datenbankversion wechseln kann.

### Entscheidung für ein Konzept

Tabelle 1 präsentiert eine Übersicht der Vor- und Nachteile der unterschiedlichen Konzepte. Die Schichten 5.1 sowie 5.2, die aufgrund ihrer fehlenden Praktikabilität bereits vorher ausgeschlossen wurden, werden nicht weiter betrachtet.

Die Konzepte sind in vier Kategorien bewertet worden. Die Kategorien sind:

1. Implementierungsaufwand: Wie groß ist der Implementierungsaufwand für diese Arbeit?
2. Eingriff in bestehende Anwendungen: Wie groß sind die Änderungen, die an einem Legacy System vorgenommen werden müssten?
3. Erweiterbarkeit: Kann auf eine Schemaänderung der Tabelle automatisch reagiert werden?
4. Verzögerung der Partitionen: Werden die Partition sofort (synchron) oder mit einer Verzögerung (asynchron) aktualisiert?

Tabelle 1: Vergleich der Partitionierungskonzepte

	EJB	ORM	JDBC	Trigger	Version	Logs
1.	2	1	0	1	1	0
2.	0	2	2	2	1	2
3.	1	1	1	0	1	1
4.	1	1	1	1	0	0
<b>Σ</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>4</b>	<b>3</b>	<b>3</b>

Der Bewertungsmaßstab ist grundsätzlich so gewählt worden, dass eine größere Zahl eine bessere Bewertung bedeutet. Ein kleiner Implementierungsaufwand ist demnach eine größere Zahl. Die Bewertungsskala für die ersten beiden Kategorien bewegt sich zwischen 0 und 2, um Abstufungen zu ermöglichen. Zusätzlich sind diese beiden Kategorien wichtiger einzustufen wie die beiden letzten Kategorien, da beispielsweise die Erweiterbarkeit eine nicht so wichtige Rolle spielt wie der Implementierungsaufwand. Um diese geringere Gewichtung auszudrücken, wurden die letzten beiden Kategorien entweder mit 0 oder 1 bewertet.

Aufgrund der Ergebnisse, wird die Variante ORM-Implementierung ausgewählt. Der Hauptgrund hierfür ist die zu erwartende geringste Komplexität ohne größere Einschränkungen in der Benutzbarkeit der Partitionierungsstrategie. Für die Implementierung wird eine Lösung auf Basis von JPA (Schicht 2) angestrebt. In Fällen, wo JPA Funktionalität fehlt wird der OR-Mapper direkt angesprochen. Als OR-Mapper und Implementierung von JPA wird Hibernate eingesetzt.

### LÖSUNGSANSATZ

Der hier präsentierte Lösungsansatz gliedert sich nun in zwei Schritte. Im ersten Schritt gilt es die vertikale Partitionierung zu implementieren. Danach wird mit dem vertikalen JOIN gezeigt, wie sich die Daten transparent auch wieder zusammenfügen lassen.

### Vertikale Partitionierung

Die folgende Abbildung 2 zeigt den Ablauf eines Partitionierungsvorganges.

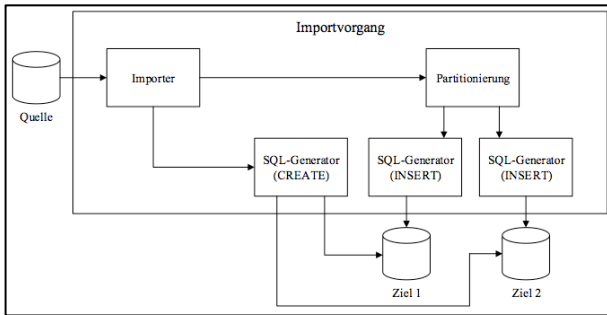


Abbildung 2: Partitionierung von Bestandsdaten

Zu Beginn werden alle bestehenden Datensätze aus der Quelldatenbank importiert. Danach werden die ausgelesenen Datensätze sowie deren Metadaten weitergereicht. Aus diesen wird dann ein Skript zur Erstellung der SQL-Tabellen auf den Zielsystemen generiert und anschließend ausgeführt. Der Partitionierungsalgorithmus generiert hieraus pro Partition eine neue SQL-Tabelle, die jeweils nur die Daten für die entsprechende Partition enthält. Im letzten Schritt werden aus den partitionierten Tabellen SQL-Skripte zum Einfügen (Insert) der Datensätze auf den Zielsystemen generiert. Mit der Ausführung der Skripte ist die initiale Übernahme der Daten in die Partitionen abgeschlossen.

Die Überwachung von neuen Datensätzen, die nach erfolgreicher initialer Datenpartitionierung in die entsprechenden Partitionen geschrieben (Insert) werden, ist durch sog. Hibernate Life-Cycle-Events realisiert. Diese Daten werden im Folgenden „Echtzeitdaten“ genannt. Der gleiche Mechanismus wird ebenso für das Ändern (Update) und das Löschen (Delete) der partitionierten Daten verwendet. Eine Änderung an einer Entität löst ein Ereignis aus, woraufhin die zu ändernden Daten eingelesen werden. Die darauffolgenden Schritte sind analog zur initialen Partitionierung, wobei je nach Ereignis neben INSERT-Statements auch DELETE- oder UPDATE-Statements (s. o.) generiert werden müssen. Abbildung 3 zeigt den Ablauf einer solchen Partitionierung.

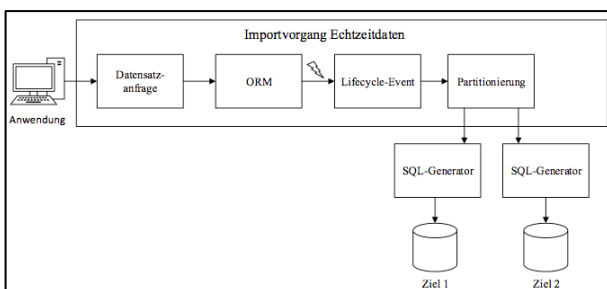


Abbildung 3: Partitionierung von Echtzeitdaten

### Vertikaler JOIN

Das Zusammenfügen der Daten ist nun der zweite Schritt bei der Betrachtung der Problemstellung. Die aufgeteilten Daten lassen sich anhand ihres Primärschlüssels wieder zusammenfügen (JOIN). Dies ist dem MapReduce-Ansatz (Dean and Ghemawat 2004) sehr

ähnlich, bei dem die disjunkten Teilmengen jeder Partition berechnet werden (Map) und anschließend auf die Lösungsmenge vereinigt (reduziert) werden (Reduce).

Die Zusammenführung der partitionierten Daten wird grundsätzlich durch den Primärschlüssel eines Datensatzes ermöglicht. Dieser wird in beiden Partitionen abgelegt und kann damit verwendet werden, um Daten aus unterschiedlichen Partitionen zu integrieren. Ähnlich wie bei der Partitionierung von Echtzeitdaten, wird das PreLoad-Event genutzt, um Ladeanforderung abzufangen und an die entsprechende Partition weiterzureichen. Das PreLoad-Event ist in der aktuellen Fassung von JPA nicht vorhanden, weshalb hier auf ein reines Hibernate Ereignis zurückgegriffen werden musste. Abbildung 4 stellt diesen Ansatz bildlich dar.

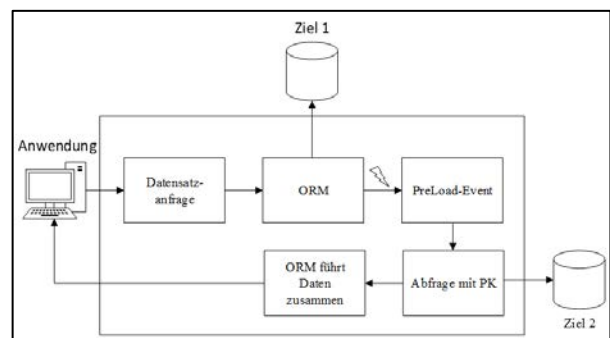


Abbildung 4: Abruf von Daten

### WEITERFÜHRENDE FRAGESTELLUNGEN

Der Einsatz des PreLoad-Events erfordert momentan ein bekanntes SELECT n+1 Problem (Bauer und King 2007). Durch die Erzeugung einer einzigen Query, die n Entitäten zurückgibt, werden erneut n Queries erzeugt und an die zweite Partition gesendet. Bei einer hinreichend großen Anzahl an Datensätzen leidet die Performance stark unter dieser Herangehensweise. Für einen praktikablen Einsatz mit einer hohen Anzahl an Datensätzen wäre es daher erforderlich, für diesen Bereich eine verbesserte Strategie, analog zu z. B. Batch-Updates (Cordts et al. 2011; Bauer und King 2007), zu entwickeln.

Eine weitere Fragestellung ist der Zugriff auf die zweite Partition. Die momentane Implementierung erlaubt nur die Erstellung von Abfragen für die Primärpartition. Die Daten einer anderen Partition können nur geladen werden, wenn die erste Partition Tupel zurückliefert. Für den praktischen Einsatz wäre es erforderlich eine Zugriffsmöglichkeit für alle Partitionen zu entwickeln. Als Stolperstein präsentiert sich dabei vor allem die Kombination von Abfragebedingungen, bei denen die Bedingung nur durch Anfragen an mehrere Server aufzulösen ist. Hierzu wird an dieser Stelle als Konzept eine Anlehnung an In-Memory Datenbanken vorgeschlagen. So sollte vor jeder Abfrage die komplette Datenbank in den Hauptspeicher des Servers geladen werden, der die Abfrage an die verschiedenen Partitionen stellt. Da sich dieser sinnvollerweise innerhalb des Unternehmens-

netzwerks befindet und die Kommunikation SSL-verschlüsselt abläuft, wird diese Vorgehensweise als sicher angesehen. Die Partitionen werden also auf diese Weise wieder zusammengefügt und somit ist die Ausgangstabelle für sämtliche Abfragen wiederhergestellt. Zuletzt stellt sich die Frage nach der Anzahl der jeweiligen Partitionen. Exemplarisch und um die Komplexität im Rahmen des Projekts beherrschbar zu machen, wurde die technische Machbarkeit hier für zwei Partitionen gezeigt. Die Erweiterung um 3 oder n Partitionen erfolgt bei einfachem Primärschlüssel allerdings analog, indem die Primärschlüssel dann eben in jede Partition repliziert werden. Performanceanalysen müssen allerdings erst noch zeigen, ob der Performanceverlust dieses Ansatzes für große Datenmengen zu verkraften ist.

Gerade bei der Betrachtung der Performance wäre ein naheliegender Schritt die Daten zu komprimieren, um den Kommunikationsoverhead so gering wie möglich zu halten. Auf der anderen Seite stehen dabei wieder die Punkte Datensicherheit und -schutz, die eine zusätzliche Verschlüsselung der einzelnen Partitionen nahelegen. Hier kommt wieder einmal die zentrale Herausforderung dieser Thematik zum Vorschein, bei der es zwischen Performance und Sicherheit abzuwägen gilt.

Weitere Fragestellungen, die weitere erhebliche Forschungsarbeit benötigen werden zudem im Ausblick dieser Arbeit aufgegriffen.

## VERWANDTE ARBEITEN

Das hier vorgestellte Konzept und seine Implementierung basieren auf der Idee, die Nutzung von „Cloud Computing“ sicherer zu machen als bisher. Die erhöhte Sicherheit entsteht dabei durch die Verteilung der Daten auf verschiedene Cloud Provider. Obwohl der Begriff des „Cloud Computings“ sehr weit gefasst werden kann, haben die wissenschaftliche Community und auch die Industrie mittlerweile ein recht konkretes Bild davon, was das „Cloud Computing“ charakterisiert. Die Definition des National Institute of Standards and Technology (NIST) hat sich durchgesetzt, da dies die umfassendste und zugleich prägnanteste ist. U. a. werden in dieser Definition die drei Servicemodelle „SaaS“, „PaaS“ und „IaaS“ genannt. Das SeDiCo-Projekt und damit auch diese Arbeit bewegen sich dabei im „IaaS“-Bereich. Dies wurde aus Gründen der Anbieterunabhängigkeit so gewählt, da auf dieser untersten Ebene die Kapselung der verschiedenen Anbieterschnittstellen in ein Abstraktionsframework am einfachsten zu realisieren ist. Dies liegt in der Heterogenität der Dienste in den oberen Schichten des Servicemodells („SaaS“ und „PaaS“) begründet. Durch Virtualisierung lassen sich im „IaaS“-Modell Ressourcen dynamisch zu- und wieder abschalten (Meir-Huber 2010). Dies ermöglicht ein einfaches scale-out aber auch ein einfaches scale-up der Ressourcen (Maget et al. 2007). Im Gegensatz dazu bieten die anderen beiden Servicemodelle diese Möglichkeit nicht, da sie keinen direkten Zugriff auf die zugrundeliegenden Ressourcen ermöglichen.

Mit dem Konzept der Datenverteilung beschäftigt sich auch das MimoSecco Projekt, das allerdings die Ent-

wicklung einer Middleware zum Ziel hat. Damit bewegt sich dieses Projekt eine Stufe höher als SeDiCo, da es sich mit der hardwarebasierten Absicherung (mittels sog. „Tokens“) von beliebigen Clients befasst. Daten lassen sich damit verteilt, über wechselnde Standorte sicher bearbeiten und abrufen. Im Fokus steht dabei nicht die Datenbankebene, sondern der Zugriff auf Dienste und Dateien im Allgemeinen (Schiefer 2013). Andere Arbeiten in diesem Kontext beschäftigen sich mit der Skalierbarkeit von Datenbanken. So wird im Leads-Projekt an der Hochschule in Karlsruhe untersucht, bis zu welchem Datenvolumen SQL-Datenbanken gut skalieren und ab welchem Volumen NoSQL Datenbanken die bessere Wahl sind. Mit Blick auf neue Herausforderungen im Kontext von „Big Data“ (s. a. Ausblick) gibt dieses Projekt sehr nützliche Hinweise und Impulse für die weitere Arbeit im SeDiCo-Projekt.

## ZUSAMMENFASSUNG UND AUSBLICK

Mit dieser Arbeit wurde die technische Machbarkeit der vertikalen Datenverteilung nachgewiesen. Damit die Komplexität der Verteilung im Rahmen des SeDiCo-Projekts beherrschbar bleibt, wurde dieses prototypisch anhand einer Tabelle mit fiktiven Kundendaten und eines Primärschlüssels gezeigt. Verschiedene Ansätze für die Verteilung wurden in die Überlegungen miteinbezogen. Des Weiteren wurden die Vor- und Nachteile der entsprechenden Ansätze aufgezeigt. Diese Analyse brachte zum Vorschein, dass die Verteilung mit Hibernate und den zugehörigen Listern am besten zu realisieren ist. Dies bezieht sich zum einen auf die einfache Integration in bestehende Infrastrukturen und zum anderen auf das nachfolgende Zusammenführen (vertikaler JOIN) beim Abruf der Daten. Im Folgenden werden nun weitere Herausforderungen genannt, die den Rahmen des SeDiCo-Projekts sprengen würden. Sie geben jedoch wichtige Anhaltspunkte und damit die Richtung für künftige nachfolgende Forschungsprojekte vor.

Diese Arbeit konzentrierte sich auf das Kernthema des SeDiCo-Projekts: die vertikale Datenverteilung. Neben diesem zentralen Aspekt steht die Abstraktion verschiedener Cloud-Schnittstellen im Vordergrund. Um einen einheitlichen Zugriff zu unterschiedlichen Cloud-APIs zu gewährleisten, wurde das jclouds Framework benutzt. Nach voriger Evaluation anderer Abstraktionsframeworks wie Libcloud und Deltacloud stellt sich jclouds als das am besten für das Projekt geeignete Framework dar. Dabei ist wichtig, die Daten nicht nur zu verteilen, sondern sie auch verteilt, bei unterschiedlichen Anbietern zu speichern.

Über diese Arbeit hinausgehende Fragestellungen beschäftigen sich nun mit der vertikalen Verteilung von mehreren Tabellen, die über Fremdschlüsselbeziehungen verknüpft sind. Weiterhin werfen Tabellen mit zusammengesetzten Schlüsseln (sog. Compound Keys) weitere Fragestellungen auf. Zuletzt müssen auf Datenbankebene die ACID-Kriterien (Haerder and Reuter

1983) betrachtet werden. Gerade in verteilten Datenbanken ist die Transaktionssicherheit nicht ohne weiteres zu gewährleisten, da Techniken wie z. B. das 2-Phasen-Freigabeprotokoll mit dem zugehörigen Locking-Mechanismus immer ein Abwägen zwischen Performance und Konsistenz bedeutet. Überdies ist die künftige Entwicklung von NoSQL-Datenbanken nicht zu vernachlässigen. Gerade NoSQL-Datenbanken sind prädestiniert für die Verteilung der Daten über mehrere Knoten. Allerdings wird dabei aus Performancegründen die horizontale Verteilung (s. a. MapReduce) wegen dem besseren scale-out genutzt. Es wird also weiterhin zu untersuchen sein, ob sich die hier vorgeschlagene vertikale mit der horizontalen Partitionierung in einer Art und Weise verknüpfen lässt, um die Vorteile beider Partitionierungsarten, also eine hohe Performance und eine erhöhte Sicherheit, zu erreichen. Mit Blick auf den momentanen Trend zu „Big Data“ und dem Versuch diesen mit In-Memory Datenbanken beherrschbar zu machen, eröffnen sich hier weitere Forschungsthemen, die den Rahmen eines einzelnen Projekts bei Weitem sprengen würden.

## LITERATUR

- Bauer, C., King, G. 2007. „Java Persistence mit Hibernate“. Carl Hanser Verlag. München, Wien. S. 473 ff.
- Cordts, S. et al. 2011. „Datenbanken für Wirtschaftsinformatiker“. Vieweg+Teubner Verlag. 1. Auflage. S. 242.
- Dean, J., Ghemawat, S. 2004. „MapReduce: Simplified Data Processing on Large Clusters“. OSDI'04: Sixth Symposium on Operating System Design and Implementation. San Francisco, CA. December.
- Ganapathy, V. et al. 2011. „Distributing Data for Secure Database Services“. In: Proceedings of the 4th International Workshop on Privacy and Anonymity in the Information Society. New York, USA.
- Haerder, T. and Reuter, A. 1983. „Principles of Transaction-Oriented Database Recovery“. In: ACM Computing Surveys. Volume 15, Issue 4. S. 287-317.
- Maged M. et al. 2007. „Scale-up x Scale-out: A Case Study using Nutch/Lucene“. In: Parallel and Distributed Processing Symposium. Long Beach, CA.
- Meir-Huber, M. 2010. „Cloud Computing: Praxisratgeber und Einstiegsstrategien“. 1. Auflage. entwickler.press. S. 10 ff.
- Plattner, H. 2013. „A Course in In-Memory Data Management: The Inner Mechanics of In-Memory Databases“. Springer. S. 63 ff.
- Pritchett, D. 2008. „BASE: An Acid Alternative“. In: ACM Magazine Queue - Object-Relational Mapping. Volume 6, Issue 3, May/June. S. 48-55.
- Schiefer, G. 2012. „Wer liest alle meine Daten in der Wolke?“ OBJEKTSpektrum Online-Themenspecial Cloud Computing. <http://www.aifb.kit.edu/web/Article3082> [Zugriff: 28.10.2013]

## AUTHOR BIOGRAPHIES

**PATRICK MÜLLER** studierte Informatik an der Hochschule Mannheim. Er hat seine Bachelorarbeit zum Thema vertikal-verteilte Datenbanken am Institut für Unternehmensinformatik an der Hochschule in Mannheim sehr erfolgreich abgeschlossen. Diese Arbeit ist maßgeblich in das SeDiCo-Projekt von Prof. Dr. Thomas Specht eingeflossen. Derzeit arbeitet er bei einem mittelständischen Unternehmen als Softwareentwickler und Berater. Seine E-Mail Adresse lautet: [patrick@4-haen.de](mailto:patrick@4-haen.de).

**JENS KOHLER** arbeitet derzeit am Institut für Unternehmensinformatik als wissenschaftlicher Mitarbeiter. Neben seiner Tätigkeit als Mitarbeiter im SeDiCo-Projekt unterstützt er die Lehre an der Fakultät für Informatik mit einem Seminar zum wissenschaftlichen Arbeiten und diversen Veranstaltungen in den Grundlagenfächern „Grundlagen der Informatik“, „Algorithmen und Datenstrukturen“ sowie „Techniken der Programmentwicklung“. Seine E-Mail Adresse lautet: [j.kohler@hs-mannheim.de](mailto:j.kohler@hs-mannheim.de). Seine Webseite finden Sie unter: <http://www.informatik.hs-mannheim.de/~j.kohler>.

**THOMAS SPECHT** leitet das Institut für Unternehmensinformatik an der Fakultät für Informatik der Hochschule Mannheim. Er promovierte mit einer Arbeit zum „frameworkbasierten Engineering multimedialer Online-Dienste“. Stationen seiner wissenschaftlichen Karriere waren das KIT in Karlsruhe (damals noch Forschungszentrum Karlsruhe), das Fraunhofer Institut für Arbeitswissenschaft und Organisation (IAO), die Duale Hochschule in Stuttgart und letztlich die Hochschule Mannheim. Seine derzeitigen Lehrgebiete umfassen die objektorientierte Modellierung und Programmierung, Webarchitekturen, -technologien und -frameworks, verteilte Softwaresysteme und Komponentensoftware. Seine E-mail Adresse lautet: [t.specht@hs-mannheim.de](mailto:t.specht@hs-mannheim.de).

# NETZE FÜR DIE ZUKUNFT – HEUTE GEKNÜPFT

## LOGISTIKSYSTEME FÜR IMMER KOMPLEXERE LOGISTIKNETZWERKE

Wilfried Pfuhl  
Mitglied des Vorstands der inconso AG  
inconso AG  
Dieselstraße 1-7, 61231 Bad Nauheim, Germany  
E-Mail: wpuhl@inconso.de

### ABSTRACT

Internationale Logistiknetzwerke sind aufgrund erhöhter Lieferfrequenzen, immer mehr beteiligter Partner und Absatzkanäle, höheren Transportkosten, der Berücksichtigung von Umweltgesichtspunkten und des Bedürfnisses nach schnelleren Reaktionszeiten immer schwieriger zu steuern.

Logistiksysteme der Zukunft müssen sich deshalb durch eine Reihe wesentlicher Vorzüge auszeichnen: Dabei stehen Transparenz, eine hohe Agilität sowie Reaktionsfähigkeit ganz oben – und das alles jeweils unter Aufrechterhaltung eines steten Informationsflusses zu den Netzwerkpartnern.

Dies wird möglich mit IT-Systemen, die den zukünftigen Anforderungen an Logistiksysteme schon heute gewachsen sind. Um Agilität, Anpassungsfähigkeit und Transparenz zu gewährleisten, decken sie neben der operativen Abwicklung auch die Modellierung, Planung und Simulation integrativ ab. So eröffnen sie den Betreibern gangbare und effiziente Wege von der Kette zum Netzwerk, von der lokalen zur globalen Steuerung.

### NETZE FÜR DIE ZUKUNFT – HEUTE GEKNÜPFT

Nicht erst seit gestern zeichnet sich eine weitreichende Veränderung der gesamten Logistiklandschaft ab: Aus Lieferketten werden Logistiknetzwerke, die Logistiknetzwerke werden größer und komplexer – insbesondere rücken die Intralogistik und die Transportlogistik immer enger zusammen. Um diese Netzwerke zu knüpfen und zu steuern, bedarf es logistischer IT-Lösungen, die sich hinsichtlich Leistungsfähigkeit, Flexibilität und übergreifender Transparenz deutlich über das erheben, was die klassischen Transportmanagement- und Lagerverwaltungssysteme zu bieten hatten.

Mit dem Begriff "Supply Chain Management" hat die Logistik in jüngerer Zeit weltweit eine weitere Aufwertung in der öffentlichen Wahrnehmung erfahren. Wer allerdings aus dem Bild der logistischen Kette auf

eine lineare Struktur schließt, der greift längst zu kurz. Die Wertschöpfung mit allen verbundenen Abläufen vollzieht sich nicht entlang isolierter Ketten, sondern im Geflecht einander überlappender, nicht konvergierender, polyzentrischer Netze. Diese Logistiknetzwerke haben die Aufgabe, diverse Quellen wie zum Beispiel Lieferanten oder Produktionsstätten bedarfsgerecht und wirtschaftlich mit Senken, also Abnehmern und Kunden zu verbinden. Bei den Ressourcenknoten handelt es sich im Wesentlichen um Lager oder um bestandslos betriebene Umschlagpunkte. Die Kanten des Netzwerks bilden Strecken, die mit Hilfe von Transporten zu überwinden sind. (vgl. Bretzke 2008, 33)

Kein Zweifel, dass die Betreiber der „Knoten“ wie der „Kanten“, sprich der Lager und Transporte, ihre jeweils eigenen Bestände, Ressourcen und Abläufe sicher im Griff haben. Allerdings genügt es nicht mehr, nur die eigenen Abläufe zu überschauen. Tatsächlich ist vielerorts schon heute ein übergreifendes Bestandsmanagement entlang einzelner Ketten gewährleistet; ein durchgängiger und übergreifender Informationsfluss (etwa bei der Auftragsabwicklung) entwickelt sich; jetzt kommt es darauf an, die Logistikketten ganzheitlich zu gestalten und zu steuern (vgl. Wallenburg und Wieland 2011, 84). Denn alle Teilnehmer sehen sich schon kurzfristig mit neuen Herausforderungen konfrontiert:

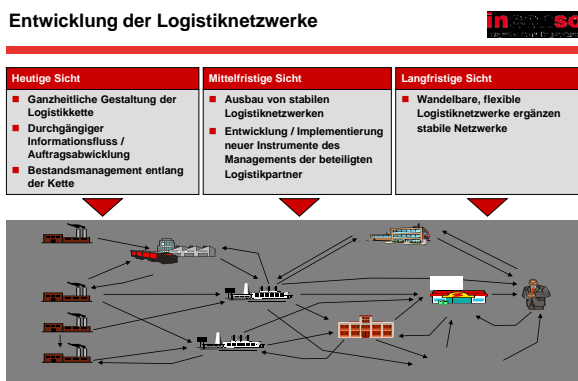
- In allen Branchen werden Wertschöpfungsketten länger und komplexer. Die Zahl der Teilnehmer innerhalb der Wertschöpfungsketten steigt, Warenflüsse müssen über Ländergrenzen hinweg im Blick bleiben.
- Die einzelnen Objekte des Logistiknetzwerks sind immer seltener in einer Hand, damit einher geht ein häufiger Wechsel der Partner und Absatzkanäle.
- Die Interaktion zwischen den Logistikpartnern wird immer intensiver, dies betrifft nicht nur die einmal abgestimmten Standardprozesse und IT-Schnittstellen, sondern auch die Servicelevel-



Vereinbarungen und deren Controlling durch Austausch von Leistungskennzahlen.

- Der Bedarf der Rückverfolgbarkeit einzelner Waren nimmt kontinuierlich zu, die zu beachtenden rechtlichen Vorschriften werden zahlreicher und komplexer.

Mittelfristig müssen daher die heute gestalteten und geplanten Logistikketten zu stabilen Logistiknetzwerken ausgebaut werden, in denen Planung, Steuerung und Überwachung durchgängig gewährleistet sind – unabhängig davon, ob der Betreiber von Haus aus Produzent, Händler, Dienstleister oder Transporteur ist. Hierfür wird die Entwicklung und Implementierung neuer Instrumente des Managements der beteiligten Logistikpartner neben dem Überwachen der Service-Level-Vereinbarungen und den entsprechenden Leistungskennzahlen weiter voranschreiten. Auf lange Sicht werden dann wandelbare, flexible Logistiknetzwerke die stabilen Netzwerke ergänzen. Sie zu planen, steuern und zu überwachen wird eine der großen kommenden Herausforderungen für das Netzwerkmanagement und für die Informationstechnologie.



Grafik 1: Entwicklung der Logistiknetzwerke

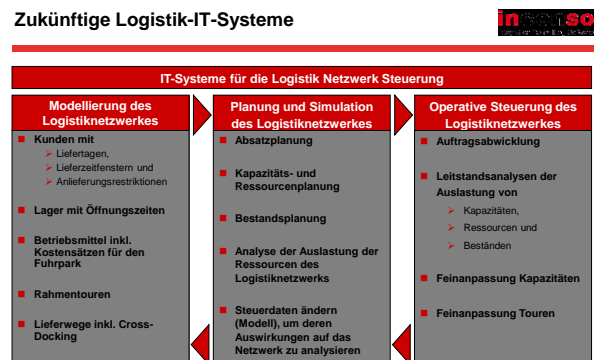
Logistiksysteme der Zukunft müssen sich daher durch eine Reihe wesentlicher Vorzüge auszeichnen: Hinter der Anforderung an hohe Agilität in der Steuerung stehen Merkmale wie eine sofortige Reaktion auf Veränderungen (zum Beispiel der Nachfrage) oder der zügige Ausgleich externer Störungen bis hin zu Krisenszenarien, die bei eventuellen Störungen unmittelbar umgesetzt werden können – und das alles jeweils unter Aufrechterhaltung eines steten Informationsflusses zu den Netzwerkpartnern (vgl. Wallenburg und Wieland 2011, 81).

Neben der kurzfristigen Flexibilität müssen sie auch eine hohe Anpassungsfähigkeit an mittel- und langfristige Veränderungen an den Tag legen – die Systeme müssen auf neue Marktentwicklungen ebenso reagieren können wie auf die Einbeziehung neuer Lieferquellen oder neuer Absatzmärkte im globalen Maßstab. Und sie sollen offen sein: nicht nur für die laufende Prozessoptimierung sondern auch für die

Entwicklung und Einbindung völlig neuer, innovativer Prozesse.

Eine transparente Prozessintegration der beteiligten Partner ist in diesen neuen Netzwerken nicht bloß durch Anbindung und Austausch im Geflecht der Lieferanten und Kunden zu gewährleisten. Zur Abbildung der jeweils konkreten Art der Zusammenarbeit müssen die verteilten Aufgaben im System präzise beschrieben und zugeordnet werden können. Denn nur so sind die fachlichen und kommerziellen Zuständigkeiten zwischen den diversen Vorlieferanten, Herstellern, Transporteuren, Logistikdienstleistern, Distributoren, Versendern und Expressdiensten bis hin zum Endverbraucher und seinen Zahlungen und Retouren sauber abzugrenzen.

Hier wird eines sofort deutlich: Die Leistungsfähigkeit und Flexibilität der IT-Anwendungslandschaft darf hinter den Anforderungen an Transparenz, Effizienz und Anpassungsfähigkeit des Netzwerks nicht zurückstehen. Bleiben die übergeordneten Ziele erhalten – Lösungen schnell und zu wettbewerbsfähigen Kosten zu entwickeln, dann hat das grundlegende Konsequenzen für die Gestaltung des Systems.



Grafik 2: Zukünftige Logistik-IT-Systeme

Mit Blick auf diese künftigen Charakteristika gehen schon jetzt Lösungen in Betrieb, die Netzwerke übergreifend modellieren, planen und steuern können. Dies geschieht sinnvollerweise im Einklang mit der vorgegebenen und übergeordneten Geschäfts- und Netzwerk-Strategie des jeweiligen Betreibers. Diese Netzwerkstrategie lässt sich schon heute im IT-System adäquat modellieren: Hier können Klienten und Mandanten mit fixen oder variablen Liefer-Zeitfenstern und -Restriktionen angelegt werden, Rahmentouren oder Lieferwege lassen sich auch über Cross-Docking-Stationen hinweg einrichten, Ressourcen in Lagern und Fuhrparks nach Verfügbarkeit und Kostensätzen beschreiben.

Auf dieser Grundlage kann das Logistiknetzwerk im System geplant und die Konsequenzen von Änderungen simuliert werden. In die Berechnungen fließen die Rahmendaten der Absatzplanung ein, die im Netzwerk verteilten ruhenden und bewegten Bestände,

Kapazitäten und Ressourcen werden transparent dargestellt und lassen sich kurz- und mittelfristig planen. Eingebunden sind dann differenzierte Auslastungsanalysen für die diversen Ressourcen, wobei die Auswirkungen bestimmter Änderungen und Eingriffe vorab simuliert werden können.

Damit erhält die operative Steuerung des Logistiknetzwerks eine erhöhte Sicherheit. Im Leitstand werden die konkreten Daten abgebildet – hier bietet eine hohe und weitreichende Transparenz die Möglichkeit, auf eventuelle Vorkommnisse jederzeit rasch und flexibel reagieren zu können. Die Erfahrungen fließen in Feinanpassungen etwa der Touren oder Kapazitäten ein – können aber auch zur Revision und Optimierung des gesamten Modells genutzt werden.

### **FALLBEISPIEL: NETZWERKSTEUERUNG IN DER DISTRIBUTION EINES LEBENSMITTELHERSTELLERS**

Im Fallbeispiel ist die Produktion auf fünf Standorte in ganz Deutschland verteilt. Eine einzige Anwendung koordiniert dabei alle Bestände und Prozesse sämtlicher Standorte und steuert die Abläufe zentral: Die einzelnen Lagerstandorte arbeiten wie Bereiche eines einzigen Lagers. Auch Shuttle-LKW zwischen den Standorten und zu dem Lager, das den letztendlichen Warenausgang verantwortet, sind angelegt. Kundenbestellungen lösen systemgestützt Kommissionierung und Auslagerung aus, so dass die Paletten bereits bei Ankunft des Ausliefer-LKW bereitstehen. Mit der letzten Palette erzeugt das System automatisch die gesamte Tourendokumentation.

Für die IT-Systeme heißt das Folgendes: Hier wurde ein integrierter Ansatz durch die enge Verzahnung und die standardisierte Interaktion der IT-Systeme verfolgt. Alle Lieferaufträge werden im SAP Retail System erzeugt. Die logistikspezifischen Daten werden dann für das gesamte Netzwerk im zentralen inconsoSCE verarbeitet. Über die im Warenwirtschaftssystem abgebildete Distributionsstruktur erfolgt im SCE die Zusammenstellung der Touren und damit bereits auch die Festlegung der Kommissionierreihenfolge in den Verteilzentren.

inconsoSCE übergibt die Aufträge direkt den inconsoWMS der regionalen Verteilzentren und erhält von dort laufende Statusmeldungen. Sämtliche Lager-, Kommissionier- und Transitbestände sind über eine konsolidierte Bestandssicht im Zugriff. Die Sammlung und Verarbeitung der Daten ermöglicht eine systematische Überwachung von Prozessen und Produktivitätskennzahlen sowie das Benchmarking der einzelnen Standorte und die Durchführung detaillierter Auswertungen.

Hier wird es sofort deutlich: Die Leistungsfähigkeit und Flexibilität der IT-Anwendungslandschaft darf hinter den Anforderungen an Transparenz, Effizienz und Anpassungsfähigkeit des Netzwerks nicht zurückstehen. Bleiben die übergeordneten Ziele erhalten

– Lösungen schnell und zu wettbewerbsfähigen Kosten zu entwickeln, dann hat das grundlegende Konsequenzen für die Gestaltung des Systems.

### **ZUSAMMENFASSUNG**

Wie auch in diesem Beispiel entstehen IT-Lösungen, die den zukünftigen Anforderungen an Logistiksysteme schon heute gewachsen sind. Um Agilität, Anpassungsfähigkeit und Transparenz zu gewährleisten, decken sie neben der operativen Abwicklung auch die Modellierung, Planung und Simulation integrativ ab. Unabhängig davon, ob diese Systeme beispielsweise auf SAP Applikationen oder auf der inconso Logistics Suite aufsetzen, interagieren diese IT-Lösungen für Logistiknetzwerke mit den verteilten operativen Systemen zur Lagersteuerung und Transportsteuerung. Ausgehend von der Verknüpfung von Transport und Warehouse Management und in Erfüllung der tatsächlichen, gegenwärtigen Anforderungen eröffnen den Betreibern gangbare und effiziente Wege von der Logistikkette zum Logistiknetzwerk, von der lokalen zur globalen Modellierung, Planung und Steuerung.

### **LITERATURVERZEICHNIS**

- Bretzke, W. 2008. *Logistische Netzwerke*. Springer-Verlag Berlin Heidelberg.
- Wallenburg, C.M. und A. Wieland. 2011. „Wie sich Supply Chains gegen Dynamik und Komplexität wappnen können.“ In: Wolf-Kluthausen, H. (Hg.) *Jahrbuch Logistik 2011*. free Beratung GmbH. Korschbroich. 80-84.

**Rezension**  
**Thomas Barton – E-Business mit**  
**Cloud Computing,**  
**Grundlagen und betriebliche**  
**Anwendungen,**

Springer Vieweg Verlag, Wiesbaden 2014.

Die aktuelle Entwicklung des Internets mit den neuen Anwendungen Smart Home und Smart Watch zeigen beispielhaft den Trend zu Big Data schon auf. Wobei hier speziell die Intelligenz der Dinge ein Hinweis auf das künftige Internet der Dinge, auch der intelligenten Dinge, als Trend zu erkennen ist.

Dieser Trend ist ebenso wie die heutigen technischen Möglichkeiten der Datenkommunikation eine Grundlage für das Cloud Computing. Mit den sich dabei ergebenden Kostenreduktionen für die Erbringung dieser Datendienste wird sich das Cloud Computing noch ganz erheblich im Anwendungsbereich weiter ausweiten.

Für das E-Business mit der Abwicklung einer Vielzahl von elektronischen Geschäftsprozessen wird hier ein wesentlicher Weiterentwicklungsschritt im Marketing erwartet.

Die sich daraus ergebenden Praxisanforderungen führen zu der Forderung nach Kompetenzen für viele Berufsfelder in diesem Gebiet für Praktiker und Studierende.

Die betriebswirtschaftlichen Aspekte werden im Buch zum E-Business, dem Cloud Computing, den vielfach angesprochenen betrieblichen Anwendungen sowie den aufgezeigten Beispielen angesprochen.

Die informatikbezogene Grundlagen zum XML, zu Web Services, zu REST und zu CMS werden, soweit dies zum Verständnis notwendig ist, aufgezeigt.

Die enthaltenen Praxisbeispiele zeigen den Zusammenhang des Erkennens der heutigen Möglichkeiten mit der Entwicklung der Anforderungen bis zur Lösungsentwicklung auf.

Mit didaktischem Geschick werden hier die betrieblichen Aufgaben in den Übungs-

aufgaben mit Lösungen und einigen Ausführungen zu Lösungsvarianten angeboten.

Als zusätzliche Online Angebote vom Autor werden ergänzende Übungsaufgaben und einige weitere Erklärungen zu Lösungsvarianten vom Vieweg Verlag unter [www.springer-vieweg.de](http://www.springer-vieweg.de) angeboten.

Als Gesamtbeurteilung ist hier auf den hohen Nutzwert der didaktisch gut entwickelten Argumentationstechnik und die gut auf die heutige Praxis ausgerichteten Übungsaufgaben hinzuweisen.

Im November 2013,  
Prof. Gerhard Fuchs,  
Fachbereich Angewandte Informatik,  
Hochschule Fulda,  
36039 Fulda,

[Gerhard.Fuchs@informatik.hs-fulda.de](mailto:Gerhard.Fuchs@informatik.hs-fulda.de).