

Auftragsannahme- und Lagerhaltungsentscheidungen bei auftragsbezogenen Instandhaltungsprozessen

Masterarbeit

zur Erlangung des akademischen Grades
„Master of Science (M.Sc.)“ im Studiengang
Wirtschaftswissenschaft

der Wirtschaftswissenschaftlichen Fakultät
der Leibniz Universität Hannover

vorgelegt von

Robert Matern

geb. am 7. März 1987 in Tschimkent
(Matrikel-Nr. 2798160)

Erstprüfer: Prof. Dr. Stefan Helber

Hannover, den 11. September 2015

Inhaltsverzeichnis

Abkürzungsverzeichnis	iv
Symbolverzeichnis	1
Abbildungsverzeichnis	2
Tabellenverzeichnis	3
Quellcodeverzeichnis	4
Vorwort	5
1 Einleitung	6
1.1 Problemstellung	6
1.2 Zielsetzung	8
1.3 Aufbau der Arbeit	9
2 Grundlagen zu auftragsbezogenen Instandhaltungsprozessen	12
2.1 Einordnung in die Produktionswirtschaft und Begriffsbestimmung .	12
2.2 Charakteristika	15
2.3 Relevanz für betriebliche Entscheidungen	17
3 Das Konzept des Revenue Managements	21
3.1 Herkunft des Revenue Managements	21
3.2 Anwendungsvoraussetzungen und Instrumente des Revenue Managements	23
3.3 Mathematische Modellformulierung des Revenue Managements . .	25
4 Bestehende Ansätze zur Annahme von Aufträgen in der Auftragsfertigung und bei Instandhaltungsprozessen	35
5 Ein exaktes Lösungsverfahren zur Auftragsannahme- und Lagerhaltungsentscheidung bei auftragsbezogenen Instandhaltungsprozessen	44

5.1	Grundlegendes zum Lösen und Implementieren des Auftragsannahme- problems	44
5.2	Mathematische Formulierung eines Modells zur Auftragsannahme- und Lagerhaltungsentscheidung	48
6	Numerische Untersuchung	61
7	Schlussbemerkung	65
	Anhang	66
.1	Programmcode	66
.2	Szenarien	81
.2.1	Rechnung	81
	Literaturverzeichnis	82

Abkürzungsverzeichnis

APS	Advanced Planning and Scheduling Systeme
ATO	Assemble-to-Order (engl. Begriff für kundenindividuelle Fertigung mit standardisierten Komponenten)
DLP	Deterministisch-lineares Programm
DP	Dynamisches Programm zur Annahme von Aufträgen
DP-storage	Dynamisches Programm zur Annahme von Aufträgen unter Beachtung von Lagerbeständen
GE	Geldeinheiten
MRO	Maintenance-Repair-and-Overhaul (engl. Begriff für Instandhaltungsprozess)
MTO	Make-to-Order (engl. Begriff für Auftragsfertigung)
MTS	Make-to-Stock (engl. Begriff für Lagerfertigung)
OK	Opportunitätskosten
RM	Revenue Management
SCM	Supply Chain Management (engl. Begriff für Wertschöpfungslehre)
u. B. d. R.	unter Berücksichtigung der Randbedingungen

Symbolverzeichnis

\mathbf{a}_j	Vektor des Ressourcenverbrauchs für Produkthanfrage j
a_{hj}	Verbrauch der Ressource h für Produkthanfrage j
\mathbf{c}	Vektor der Ressourcenkapazität
c_h	Kapazität der Ressource h
D_{jt}	Aggregierte erwartete Nachfrag nach Produkt j zur Periode t
h	Ressource aus der Menge \mathcal{H}
\mathcal{H}	Menge an Ressourcen
i	... \mathcal{I}
\mathcal{I}	Gesamtmenge an möglichen Kombinationen ...
j	Produkthanfrage aus der Menge \mathcal{J}
\mathcal{J}	Menge an möglichen Produkthanfragen
m	... Ausführungsmodus aus der Menge \mathcal{M}_j
\mathcal{M}_j	Menge an produktspezifisch-möglichen Ausführungsmodi für ein Produkt j
π_h	Bid-Preis der Ressource h
$p_j(t)$	Wahrscheinlichkeit der Nachfrage nach Produkt j in Periode t
r_j	Erlös des Produkts j
t	Periode des Buchungshorizonts T
τ	Endperiode $t = 1$ im Buchungshorizont T
T	Buchungshorizont
$V(\mathbf{c}, t)$	Ertragsfunktion in Abhängigkeit der Kapazitäten \mathbf{c} in der Periode t
x_{jm}	Anzahl der akzeptierten Anfragen nach Produkt j im Ausführungsmodus m

Abbildungsverzeichnis

1	Grafische Darstellung der Gliederung der Arbeit	10
2	Grafische Veranschaulichung eines Wertschöpfungsprozesses	14
3	Grafische Darstellung des Buchungshorizonts bei der Auftragsfertigung	25
4	Beispielhafte Darstellung einer rekursiven Folge eines Netzwerk RM .	29
5	Darstellung der Systemzustände des beispielhaften Netzwerk RM .	30
6	Darstellung der Systemzustände des beispielhaften Netzwerk RM (optimaler Auftragseingang)	32
7	Darstellung der Systemzustände des Netzwerk RM mit konkurrierenden Anfragen	33
8	Rekursive Übergänge der Systemzustände ohne Memofunktion . . .	45
9	Rekursive Übergänge mit Memofunktion	46
10	Zusammenhang von Buchungsperioden und den Buchungsabschnitten	50
11	Darstellung der Systemzustände des Netzwerk RM unter Beachtung der Möglichkeit der Auftragsannahme- und Lagerhaltungsentscheidung	55
12	Wahrscheinlichkeitsverteilung Szenario 1	64

Tabellenverzeichnis

1	Ergebnistabelle für das beispielhafte Netzwerk RM	31
2	Ergebnistabelle für das beispielhafte Netzwerk RM mit konkurrierenden Anfragen	34
3	Überblick über Publikationen des traditionellen Konzepts des Revenue Managements in der Fertigungsindustrie	36
4	Optimale Politik für das beispielhafte Netzwerk RM unter Beachtung der Möglichkeit der Auftragsannahme- und Lagerhaltungsentscheidung	57
5	Optimale Politik für das zweite beispielhafte Netzwerk RM unter Beachtung der Möglichkeit der Auftragsannahme- und Lagerhaltungsentscheidung	59
6	Überblick über Publikationen des traditionellen Konzepts des Revenue Managements in der Fertigungsindustrie	61
7	Szenario 1	63
8	Auswertung des Szenarios 1	64

Quellcodeverzeichnis

1	Parameter Definition für die Dynamische Programmierung im Netzwerk Revenue Management	66
2	Python-Algorithmus für die Dynamische Programmierung im Netzwerk Revenue Management	70

Vorwort

1 Einleitung

1.1 Problemstellung

Die Entscheidung über die Annahme von Kundenaufträgen unter Unsicherheiten hat weitreichenden Einfluss auf die Produktionsplanung und den Ertrag eines auftragsgesteuerten Fertigungsunternehmens. Ein solches Unternehmen der Auftragsfertigung steht vor dem Entscheidungsproblem, ob der Auftrag zur Produktion des Gutes wirtschaftlich ist. Im weiteren Verlauf dieser Arbeit werden Unternehmen betrachtet, die als Dienstleistung das Instandhalten von Gütern anbieten. Abhängig des eingehenden Kundenauftrags, indem der Zustand des Gutes beschrieben ist, durch den die notwendigen Prozessschritte zur Instandsetzung des Gutes abgeleitet werden können, generiert das Unternehmen unterschiedliche Erträge. Die Prozessschritte zur Instandsetzung des Gutes geben zusätzlich den notwendigen Ressourcenbedarf für die auszuführende Tätigkeit an, die notwendig sind um das Gut in seinen ursprünglichen bzw. geforderten Zustand zu versetzen. Ressourcen zur Instandsetzung von Gütern können z. B. Material oder Personalstunden sein.¹ Abhängig des möglichen Ertrags und des für den Auftrag notwendigen Ressourcenbedarf muss das produzierende Unternehmen die Entscheidung über Annahme oder Ablehnung des Kundenauftrags treffen. Sofern nur dieser einfache Fall betrachtet wird, bei dem nur der einzelne Kundenauftrag zur Auswahl steht und die Abarbeitung der Aufträge nach der Reihe erfolgt, ist die Entscheidung für das Unternehmen einfach getroffen. Der Kundenauftrag wird angenommen, sofern der Aufwand des Ressourceneinsatzes niedriger als der erzielte Ertrag ist.² Sofern der Unternehmen eine begrenzte Ressourcenkapazität zur Instandsetzung der Güter besitzt, muss zusätzlich der absolute Ressourcenverbrauch des Auftrags für Annahmeentscheidung geprüft werden. Mit Annahme des Auftrags ist ein auftragsbezogener Ertrag erzielt und ein produktbezogener Ressourcenverbrauch eingetreten. Nachdem diese Entscheidung getroffen ist, wird der zeitlich nachfolgende Kundenauftrag betrachtet.

¹Vgl. ... ???

²In dieser Arbeit wird auf eine differenzierte Betrachtung der Herstellkosten verzichtet

Es handelt sich damit um ein klassisches Verfahren des „First-come, first-served (FCFS)“.

Für die Entscheidung über die Annahme oder Ablehnung eines Kundenauftrags zur Instandhaltung von Gütern bedarf es einer umfassenderen Betrachtung, als nur die kurzsichtige Entscheidung über die Annahme einzelner Aufträge. Angenommen ein Unternehmen der Auftragsfertigung besitzt ein bestimmtes Kontingent an unterschiedlichen Ressourcen über einen bestimmten Zeitraum zur Erfüllung seiner angebotenen Dienstleistung. In diesem betrachteten Zeitraum treffen differenzierte Kundenaufträge mit unterschiedlicher Wertigkeit ein. Zur Maximierung der Erträge über diesen Zeitraum kann es für das Unternehmen sinnvoll sein, Aufträge mit niedrigem Ertrag abzulehnen, sofern im weiteren Verlauf des betrachteten Zeitraums Aufträge mit höherem Ertrag eintreffen. Dies erfolgt in Abhängigkeit der noch vorhanden knappen Ressourcenkapazität, die für die unterschiedlichen Kundenaufträge aufgewendet werden müssen.

Eine weitere Alternative wäre es für das instandhaltende Unternehmen, das zu reparierende Gut mit ein neuwertiges Gut auszutauschen. Es handelt sich damit um eine Entscheidung der Lagerhaltung bei auftragsbezogenen Instandhaltungsprozessen. Die Entscheidung zur Reduzierung des vorhandenen Lagerbestands zur Befriedung des Kundenauftrags im Fokus einer optimalen Kapazitätsauslastung und einer möglichen Ertragsmaximierung macht vor allem Sinn, wenn die zur Produktion notwendigen Ressourcen für spätere Aufträge notwendig sind und mögliche Kapazitäten für die Befriedigung von aktuell geforderten Gutes vorhanden sind. Auch hier kann zwischen dem einfachen Verfahrens des FCFS und eines differenzierten Verfahrens unter Berücksichtigung von zukünftigen Fertigungsaufträgen unterschieden werden. Die kurzfristige Sichtweise untersucht nur den notwendigen Kapazitätsaufwand und den potentiell möglichen Ertrag. Bei der langfristigen Sichtweise unter Berücksichtigung von zukünftigen Anfragen wird der Lagerbestand über einen längeren Zeitraum betrachtet. Sofern in der nahen Zukunft viele Aufträge eintreffen, muss ein solches langfristiges Verfahren eine Entscheidungsunterstützung dem Unternehmen liefern, ob eine Lagerproduktion erfolgen soll. Wie die Bezeichnung der Entscheidung zur Lagerproduktion bereits im Wortlauf deutlich macht, werden die durch den Ressourcenverbrauch geformten Leistungsbündel (Produkte bzw. Dienstleistung) auf Lager produziert. Damit ist es dem Unternehmen möglich zu einem späteren Zeitpunkt die Anfragen mittels des Lagerbestands zu befriedigen.

1.2 Zielsetzung

In dieser Arbeit werden unterschiedliche mathematischen Modell zur Annahmeentscheidung eines Auftrags zur Instandhaltung eines Gutes betrachtet. Dabei werden verschiedene Betrachtungsweisen der Auftragsannahme via Ressourcenkapazität oder Lagerbestand aufgezeigt und untersucht. Dabei entspricht das reparierte Gut den vom jeweiligen Auftrag geforderten Instandhaltungszustand. Anders formuliert bedeutet dies, dass das produzierende Unternehmen nicht nur die Entscheidungsmöglichkeit über die Instandsetzung des Gutes mittels der Verfügbaren Ressourcenkapazität besitzt, sondern auch die Möglichkeit hat in Abhängigkeit des verfügbaren Lagerbestandes die Kundenaufträge mittels eines neuwertigen Gutes zu befriedigen.

Bei der Problemformulierung der Auftragsannahmeentscheidung bei auftragsbezogenen Instandhaltungsprozessen handelt es sich um ein stochastisch-dynamisches Optimierungsmodell des Netzwerk Revenue Managements. Das Optimierungsmodell trifft die Entscheidung, welche Option zum Lösen des Auftragsannahmeproblems gewählt wird. Das Modell betrachtet dabei den maximal möglichen Erwartungswerts nach Annahme der Option. Als Optionen stehen dem Modell die Auftragsannahme zur Instandsetzung des Gutes, die Auftragsablehnung mit ggf. einer Lagerproduktion von neuwertigen Gütern und die Lagerentnahme eines solchen neuwertigen Gutes zur Verfügung. Diese Entscheidung erfolgt in Abhängigkeit der Veränderung der verfügbaren Ressourcenkapazitäten nach einer möglichen Instandsetzung des Gutes, des aktuell-vorhandenen Lagerbestandes der bereits reparierten Güter und der noch potentiell eintreffenden Anfragen des Netzwerks.

Das Revenue Managements (RM) hat seinen Ursprung in der betrieblichen Problemstellung der Annahme von Aufträgen von freien Sitzplätzen in Passagierflugzeugen.³ Das Grundmodell wird zur Annahme von Kundenaufträgen bzw. -anfragen von Dienstleistungsunternehmen mit beschränkten Ressourcenkapazitäten verwendet.⁴ Beim RM wird das Auftragsannahmeproblem der zeitlich nacheinander eintreffenden Produkthanfragen als *dynamisch-stochastisches Optimierungsmodell* formulieren und gelöst. Die Aufgabe des Optimierungsmodells ist laut Talluri und van Ryzin (2004b) die Entscheidungsfindung zu unterstützen, damit der Gesamtertrag des Dienstleisters maximiert wird. Der Gesamtertrag wird bewertet in Geldeinheiten (GE). Mit dem Modell lässt sich eine optimale Politik der Auftragsannahme

³Vgl. ... ???

⁴Vgl. ...

für das betrachtete Netzwerk ableiten.

Die Zielsetzung der Arbeit ist demnach das Grundmodell des Revenue Managements zur Annahme von Kundenaufträgen mit einem Verfahren der Auftragsbefriedigung mittels einer Lagerhaltung von Leistungsbündeln (bzw. Ressourcen) zu erweitern. Zusätzlich zur konzeptionellen Darstellung des hier betrachteten Auftragsannahmeproblems wird für die Modellerweiterung ein Pseudo-Algorithmus entwickelt, welches vorformulierte Beispielszenarien exakt löst. Damit lässt sich eine numerisch Untersuchung durchführen, welche Möglichkeiten sich durch die Betrachtung einer Lagerhaltung in der Auftragsannahme von auftragsbezogenen Instandhaltungsprozessen für Unternehmen ergeben.

1.3 Aufbau der Arbeit

Der Aufbau der Arbeit ist in der Grafik 1 dargestellt. In Kapitel 2 wird vorerst der Begriff der auftragsbezogenen Instandhaltungsprozessen definiert und deren theoretische Grundlagen beschrieben. In diesem Kapitel wird der Begriff in die Produktionswirtschaft eingeordnet, sowie die Beschreibung der Charakteristika und der Relevanz für betriebliche Entscheidungen der auftragsbezogenen Instandhaltungsprozessen aufgeführt. Weiter werden im Kapitel 3 die theoretischen Grundlagen der Arbeit vervollständigt, indem das Konzept des Revenue Management bei der Annahme von Aufträgen vorgestellt wird. In diesem Kapitel wird auf die Herkunft des Konzepts eingegangen. Für das Konzept bestehen zusätzlich Anwendungsvoraussetzungen und Instrumente, die in dem Kapitel beschrieben sind. Des Weiteren wird für das Grundmodell des Revenue Managements die mathematische Modellformulierung dargestellt.

Im Anschluss wird in Kapitel 4 ein Literaturüberblick über bestehende Ansätze zur Annahme von Auftragsproduktion und Instandhaltungsprozessen aufgeführt. Es werden hier vier Ansätze näher betrachtet, die den Fokus auf eine heuristische Lösung des Auftragsannahmeproblems legen.

Das Kapitel 5 zeigt die quantitative Untersuchung der Erweiterung des Grundmodells mit der Entscheidung über eine Lagerhaltung auf. Zum einen wird die mathematische Modellformulierung und zum anderen wird ein Pseudo-Algorithmus zum exakten Lösen der Problemstellung beschrieben. Weiter werden die Grundlagen zur Implementierung des Pseudo-Algorithmus genannt. Im letzten Teil des Kapitels wird

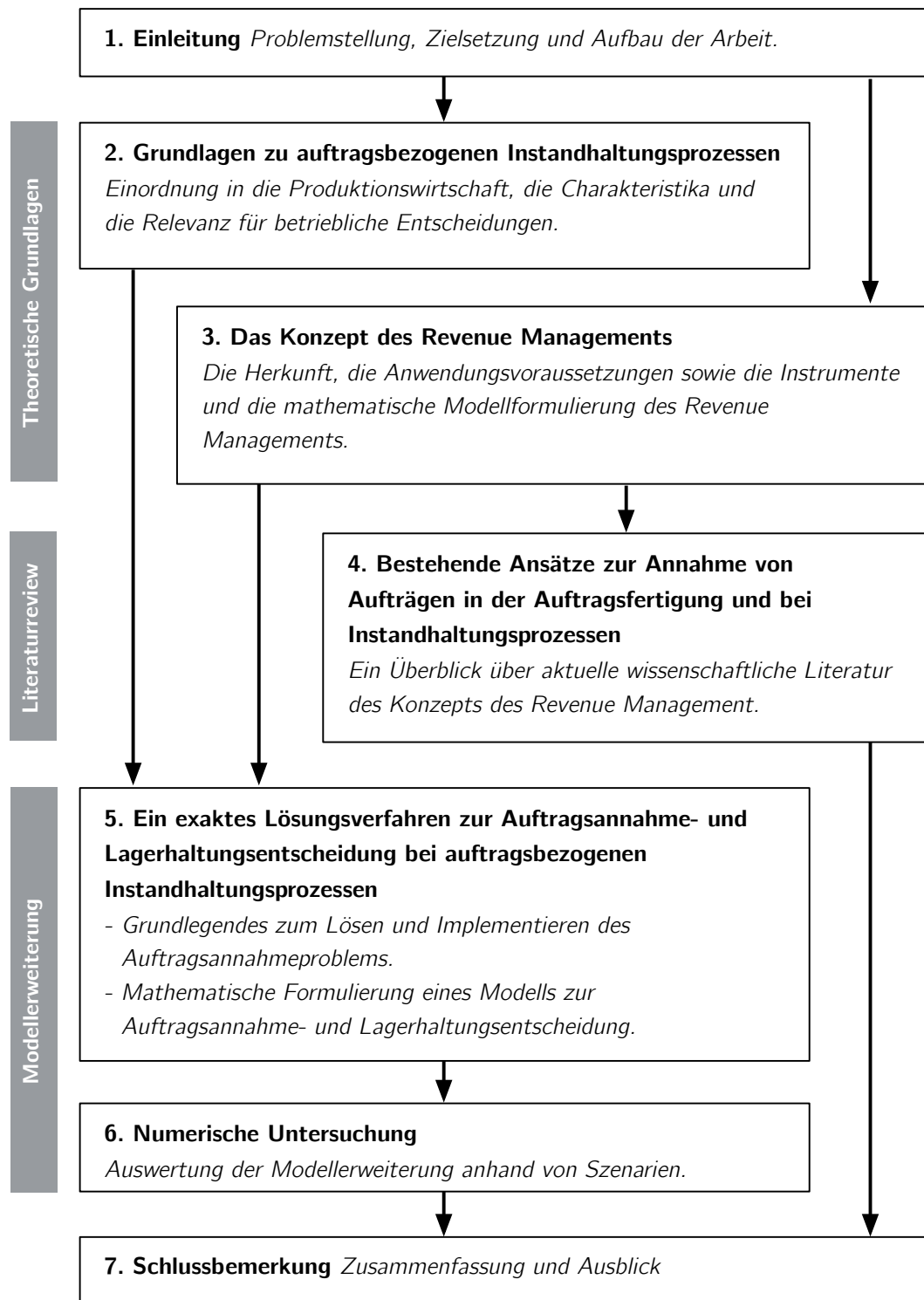


Abbildung 1 Grafische Darstellung der Gliederung der Arbeit

auf die numerische Untersuchung eingegangen.

Im letzten Kapitel sind die Schlussbemerkungen dieser Arbeit dargestellt. Es handelt dabei um eine Zusammenfassung der Ergebnisse der vorliegenden Arbeit und

um einen Ausblick für nachfolgende Forschung.

2 Grundlagen zu auftragsbezogenen Instandhaltungsprozessen

2.1 Einordnung in die Produktionswirtschaft und Begriffsbestimmung

Unter auftragsbezogenen Instandhaltungsprozessen (engl. maintenance-repair-and-overhaul; Abk. MRO)) wird ein nachgelagerter Prozess eines erweiterten Leistungsprogramms eines Unternehmens verstanden und kann damit als Teilsystem angesehen werden, welches den gesamten Wertschöpfungsprozess verlängert. Es handelt sich nicht um die interne oder beauftragte Instandhaltung von u. a. Anlagen und technischen Systemen, sondern versteht eine Leistung eines Unternehmens. Im weiteren Verlauf dieser Arbeit wird von MRO-Prozessen gesprochen. Da es bei MRO-Prozessen um eine Produktion einer Leistung handelt, werden diese der betriebswirtschaftlichen Betrachtung der Produktionswirtschaft zugeordnet.

Bei dem Begriff der Produktionswirtschaft handelt sich um ein Teilgebiet der Betriebswirtschaftslehre.⁵ In der Produktionswirtschaft wird der Fokus auf die Produktion von Leistung gelegt.⁶ Bei diesem ökonomischen Konzept wird die Transformation von materiellen und nichtmateriellen Inputgütern (Produktionsfaktoren) hin zu gewünschten Outputgütern (Leistung des Unternehmens) betrachtet. Bei den im Laufe der Zeit erweiterten betriebswirtschaftlichen Produktionsfaktoren nach Gutenberg (1959, S. 71) handelt es sich um die Elementarfaktoren Werkstoffe, Betriebsstoffe, Betriebsmittel und objektbezogene humane Arbeitsleistung sowie um die dispositiven Faktoren Betriebsführung, Organisation und Planung.⁷ Bei Outputgütern handelt es sich um Produkte in Form von Sach- oder Dienstleistungen die dem Markt und somit der potentiellen Nachfrage der Marktteilnehmer

⁵Neben der Teilgebiete Finanzwirtschaft, Marketing, Unternehmensführung, Unternehmensrechnung etc., vgl. dazu Dyckhoff und Spengler (2010), S. 3.

⁶???

⁷????

zur Verfügung gestellt werden.⁸ Die Transformation erfolgt durch bestimmte von Menschen veranlasste unternehmerischen Verfahrensweisen.⁹ Beispielsweise kann hier die industrielle Fertigung von Verbrauchs- oder Gebrauchsgütern genannt werden.

Bei der Transformation der Inputgüter erfolgt eine qualitative, quantitative, räumliche oder zeitlichen Veränderung der Objekte.¹⁰ Durch diese Veränderung kann seitens des Unternehmens eine Leistung auf dem Markt angeboten werden. Damit diese Leistung den Absatz bei potentiellen Konsumenten findet, muss die Leistung durch die Transformation eine Wertschöpfung erhalten. Der konzeptionelle Rahmen dieses Gedankens bildet die Wertschöpfungslehre (engl. supply chain management; Abk. SCM)).¹¹ Danach sollte das Ziel eines jeden Unternehmens das Betreiben von Wertschöpfung sein.¹² In der klassischen Auffassung der Wertschöpfungslehre durchläuft die Leistungserstellung alle (Teil-)Systeme des Unternehmens.¹³ Abbildung 2 zeigt in Teil a eine mögliche Abfolge der Systeme eines Unternehmens. Eine klassische Abfolge zur Leistungserstellung bzw. der Transformation von Inputgütern hin zu Outputgütern ist die Abfolge der Systeme Forschung/Entwicklung, Beschaffung, Produktion, Distribution sowie Verkauf. Damit ist die um die Wertschöpfung erhöhte Leistung auf dem Markt angekommen und das Unternehmen erzielt damit i. d. R. einen Ertrag.

Sofern das Unternehmen eine **Instandhaltung** bzw. einen MRO-Prozess als Leistungen anbieten kann, wird die Abfolge des Wertschöpfungsprozesses um dieses Unternehmenssystem erweitert. Für den gewerblichen Verkauf von Gütern an Privatkunden können gesetzliche Regelungen bestehen, womit ein Unternehmen gezwungen ist das Unternehmenssystem der Instandhaltung in den Wertschöpfungsprozess aufzunehmen.¹⁴ Andererseits kann ein Unternehmen auch die Instandhaltung seiner eigenen Güter als eigenständig angebotene Leistung bereitstellen und damit den Gesamtertrag des Unternehmens erhöhen (MRO-Prozess). Dieses wird oft von Unternehmen angeboten, die ihren Umsatz mit komplexen Produkten erzielen. Komplexe Produkte zeichnen sich durch ihre hohe Anzahl an hochtechnologischen Komponenten aus, die erst durch den vollständigen und meist kostenintensiven

⁸Vgl. Schmidt (2012), S. 1.

⁹Vgl. Tempelmeier und Günther (1994), S. 6, echt????

¹⁰Vgl. Dyckhoff und Spengler (2010), S. 3.

¹¹Vgl. ?, S. ??.

¹²Vgl. ?, S. ??.

¹³???

¹⁴Vgl. die Richtlinie 1999/44/EG des Europäischen Parlaments und des Rates vom 25. Mai 1999 zu bestimmten Aspekten des Verbrauchsgüterkaufs und der Garantien für Verbrauchsgüter.

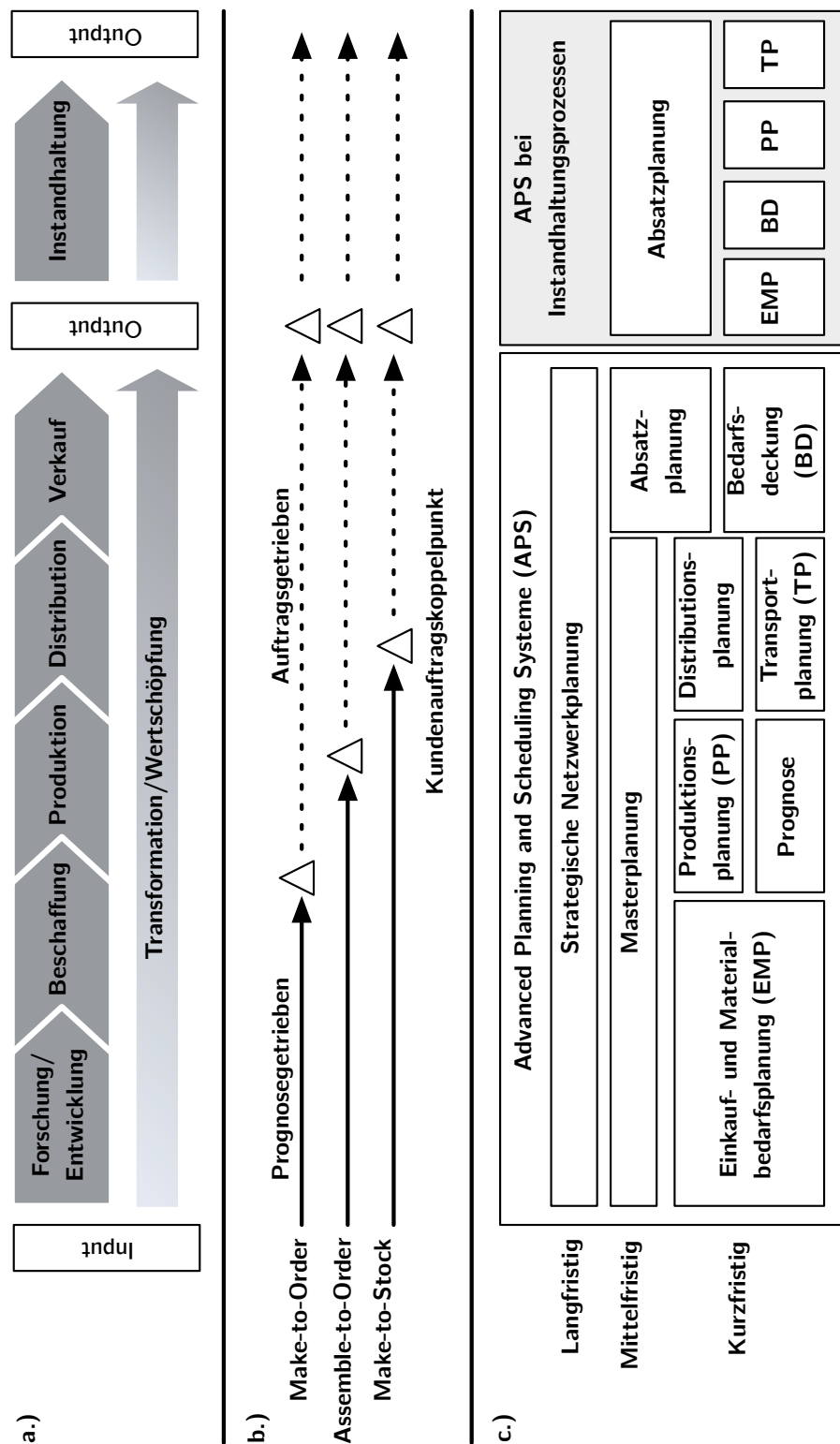


Abbildung 2 Grafische Veranschaulichung eines Wertschöpfungsprozesses

In Anlehnung an: Bach et al. (2012), S. 4-5; Quante (2009), S. 21-22; Meyr et al. (2015), S.

Zusammenschluss eine individuelle Bedürfnisbefriedigung des Nachfragers ermöglichen.¹⁵ In dieser Arbeit werden solche komplexen Produkte betrachtet, die aus einer Vielzahl von Komponenten bzw. Ressourcen bestehen. Dementsprechend handelt es sich bei den in dieser Arbeit betrachteten Instandhaltern um Anbieter von komplexen Produkten.

Weiter versteht Helbing unter Instandhaltung die „Gesamtheit der technischen und organisatorischen Mittel, Vorgänge und Maßnahmen zur Erhaltung, Verbesserung und Wiederherstellung des Funktions-, Leistungs- und Güteniveaus von materiellen Objekten während ihrer Wirkungs- und Lebenszeit durch Wartung, Inspektion und Instandsetzung.“¹⁶ In dieser Arbeit ist mit dem Begriff der MRO-Prozesse gemeint, dass für die betrachteten komplexen Produkte ein spezifischer Ausführungsmodus für die Anfragen verwendet wird, der eine erneute Integration von produkt-spezifischen Ressourcen vorsieht, damit die Funktionsfähigkeit des Produkts (Leistung) wiederhergestellt ist. Damit lässt sich das Verständnis des Begriffs als Leistung eines Unternehmens und als Unternehmenssystem ableiten, welches die Verlängerung und zugleich als Erweiterung des Wertschöpfungsprozesses versteht. Abbildung 2 zeigt im Teil a den erweiterten Wertschöpfungsprozess eines produzierenden Unternehmens inkl. der Integration des Teilsystems der Instandhaltung.

2.2 Charakteristika

Nach der DIN 310511 wird Instandhaltung insofern ausgeführt, wenn die Funktionsfähigkeit Betrachtungseinheit sichergestellt werden muss, damit der ursprüngliche Wert erhalten bleibt.¹⁷ Betrachtungseinheit können ganze Anlagen und Maschinen sein oder nur einzelne Komponenten.¹⁸

Erläuterung nach DIN 31051:¹⁹

- Instandhaltung ist die Kombination aller technischen und administrativen Maßnahmen des Managements während des Lebenszyklus einer Betrachtungseinheit.

¹⁵Vgl. Schmidt (2009), S. 97

¹⁶Vgl. Helbing, S. 984

¹⁷Vgl. Strunz (2012), S. 1.

¹⁸Vgl. Ryll und Freund (2010), S. 23.

¹⁹Unbedingt nachlesen!!!!

tungseinheit zur Erhaltung des funktionsfähigen Zustandes oder der Rückführung in diesen, so dass sie die geforderte Funktion erfüllen kann.

- Als Betrachtungseinheit (BE) wird jedes Bauelement, Gerät, Teilsystem, jede Funktionseinheit, jedes Betriebsmittel oder System, das für sich allein betrachtet werden kann, definiert.

Nach der DIN 31051 werden Einheiten betrachtet, die eine Wartung, Inspektion, Instandsetzung oder Verbesserung bedürfen.²⁰ Bei einer Wartung handelt es sich um Maßnahmen zur Verzögerung der Abnutzung. Die Inspektion umfasst alle Maßnahmen der Begutachtung sowie der Beurteilung des Ist-Zustandes einer Betrachtungseinheit und die Instandsetzung beinhaltet die Maßnahmen zur Wiederherstellung des Sollzustands. Mit der Verbesserung sind Maßnahmen gemeint, die den Soll-Zustand der Betrachtungseinheit erweitern, damit mögliche Defekte verhindert werden.

Instandhaltungsprozesse lassen sich weiter nach Ausführungszeitpunkt und Ausführungsort der Maßnahmen unterscheiden.²¹ Diese Unterscheidung spielt bei der Betrachtung von MRO-Prozessen eine untergeordnete Rolle.

Da der Absatz zeitlich vor der Produktion der Leistung stattfindet, handelt es sich bei dem hier betrachteten MRO-Prozessen um eine Auftragsfertigung.²² In der Literatur wird für die Auftragsfertigung oft der englischen Begriff *Make-to-Order*; Abk. *MTO*) verwendet. Abzugrenzen ist der Begriff von der Lagerfertigung (engl. *Make-to-Stock*; Abk. *MTS*) und der kundenindividuellen Fertigung mit standardisierten Komponenten (engl. *Assemble-to-Order*, Abk. *ATO*). Dies kann zum einen anhand des Kundenauftragskoppelpunkts und zum anderen anhand der Planungsgrundlage der Leistungserstellung getätigt werden. Der Kundenauftragskoppelpunkt zeigt den erstmaligen Kundenkontakt bzw. Auftragseingang auf. Mit Eingang des Kundenauftrags wechselt die Planungsgrundlage der Leistungserstellung von prognosegetriebener hin zu auftragsgetriebener Planung. Die prognosegetriebenen Planungsgrundlage für die Fertigung ist mit einem Prognosefehler für die Ausführung der unternehmerischen Tätigkeit behaftet, was der analytischen Betrachtung des Kundenauftragskoppelpunkts zur Bestimmung der notwendigen Ressourcenkapazität und der weiteren Auftragseingänge weiteres Gewicht verleiht.²³

²⁰Vgl. Ryll und Freund (2010), S. 23-24

²¹Vgl. Ryll und Freund (2010), S. 24-26; Hinsch (2010), S. 190-191

²²Vgl. Hax (1956), S. 247; Gutenberg (1965), S. 164-165

²³Vgl. Quante (2009), S. 21

Bei einer MTO trifft vor der eigentlichen Produktion der Leistung der Kundenauftrag ein. Damit sind die Forschungs/Entwicklung der Leistung und die Beschaffung der Komponenten bzw. Ressourcen hauptsächlich prognosegetrieben. Alle weiteren Teilsysteme des Unternehmens können sich speziell an den Forderungen des Auftrags richten. Ähnliche Rahmenbedingung besitzt eine ATO, die den Kundenauftragskoppelpunkt innerhalb des Produktionsablaufs hat. Da nur ein gewissen Umfang der Komponenten an auftragsspezifischen Eigenschaften angepasst wird, sind die Grundkomponenten der Produkte abhängig der Prognosemethoden des Unternehmens. Bei MTS erfolgt der Auftragseingang erst nach der Produktion, womit diese Fertigungsart den höchsten Anteil des Einsatzes von Prognosemethoden aufweist.²⁴ Abbildung 2 zeigt im Teil b die Unterschiede der Fertigungsarten im Verlauf des Wertschöpfungsprozesses.

Da bei MRO-Prozessen erst mit Auftragseingang die erforderlichen Produktionsschritte und der Ressourcenbedarf bekannt sind, werden diese mit MTO-Prozessen gleichgesetzt. Zur Erfüllung möglichst vieler Aufträge muss eine möglichst gute Prognose der benötigten Ressourcen vorliegen, damit kurze Liefer- und Durchlaufzeiten gewährleistet bleiben.²⁵ Sofern die Prognose fehlerhaft ist und ein zu geringer Bestand an Ressourcen vorhanden ist, konkurrieren die unterschiedlich eintreffenden Anfragen nach MRO-Prozessen des Unternehmens bzgl. der Ressourcen untereinander. Sofern ein zu hoher Ressourcenbestand vorhanden ist, stellt sich die Frage über den besten Mix der unterschiedlichen Aufträge, damit die Ressourcenkapazität optimal genutzt wird und dementsprechend der Ertrag maximiert wird. Im nächsten Abschnitt wird dieser Frage in Bezug von Produktionsplanungssystemen und der betrieblichen Entscheidungsfindung weiter nachgegangen.

2.3 Relevanz für betriebliche Entscheidungen

Die Entscheidung über die Ausgestaltung des MRO-Prozesses hat eine hohe Relevanz für den Erfolg der angebotenen Leistung sowie der Effizienz des gesamten SCM-Prozesses des Unternehmens und kann der nachhaltigen Konsumentenzufriedenheit dienen. Wichtiger Erfolgsfaktor für ein erfolgreiches SCM und somit der Konsumentenzufriedenheit ist der Einsatz von Advanced Planning and Scheduling Systemen (APS).²⁶

²⁴Vgl. Fleischmann und Meyr (2004), S. 300-303; Quante (2009), S. 21-22

²⁵Vgl. Thaler (2001), S. 68.

²⁶Vgl. Fleischmann und Meyr (2004), S. 298

Bei APS handelt es sich um computergestützte Systeme, mit denen die Planung der ganzheitlichen Wertschöpfung der Leistung mit Hilfe von mathematischer Modelle des Operations Research (OR) unterstützt wird.²⁷ Neben der reinen Planung der Produktion, helfen moderne APS mit ihrem modularen Aufbau auch bei der gesamten Auftragsabwicklung und unterstützen damit Unternehmensteilnehmer bei betrieblichen Entscheidungen innerhalb des SCM-Prozesses, wie z. B. bei dem Einkauf von Ressourcen, der Produktionsplanung und dem Absatz der Leistung.²⁸

Meyr et al. (2015) sortieren einige wichtige Module eines APS in den Dimensionen des SCM-Prozesses und des Planungshorizonts. Abbildung 2 in Teil c greift diese Sortierung auf. Bei dem Modul der strategischen Netzwerkplanung werden die Lieferanten, Werke und Lagerpunkte in einer langfristigen Betrachtungsweise festgelegt. Die Entscheidungen die Aufgrund der strategischen Netzwerkplanung getroffen werden, haben großen Einfluss auf die langfristige Rentabilität sowie Wettbewerbsposition des Unternehmens und haben räumlichen sowie zeitlichen Charakter.²⁹ D. h. das Unternehmen die Planungsmodelle und Entscheidungen aufgrund der strategischen Netzwerkplanung in Bezug auf bestimmte regionale Gegebenheiten und einem vordefinierten Betrachtungszeitraum festlegen????³⁰ Die Absatzplanung dient einerseits der Absatzprognose und andererseits der Analyse der notwendigen Sicherheitsbestände.³¹ Das Modul Masterplanung synchronisiert den Materialfluss entlang des gesamten SCM-Prozesses und unterstützt dadurch mittelfristige Entscheidungen über die effiziente Nutzung der Ressourcen, damit für einen kontinuierlichen Materialfluss größere Puffer gemieden werden.³² Mit dem Modul der Einkaufs- und Materialbedarfsplanung werden kurzfristig terminierte Pläne für Komponenten und Teile (Ressourcen) für die Fertigung berechnet.³³ Aufbauend auf diesen Plänen kann die Produktionsplanung und die weitere Prognose für die Produktion und Logistik erfolgen, die den organisatorischen Anforderungen des Produktionssystems gerecht werden müssen und einen kurzfristigen Planungshorizont aufweisen.³¹ Bei diesen Modulen werden in einer kurzfristigen Betrachtung u. a. die Maschinenverfügbarkeit und die Losgrößen geplant bzw. prognostiziert. Eine effiziente Produktionsplanung setzt eine gute Prognose voraus.³⁴ Bei den Modulen der Distributions- und Transportplanungen werden die Pläne für die Verkaufsstel-

²⁷????? steht eig auch bei Fleischmeyer, aber besser andere Quelle finden...

²⁸Vgl. Meyr et al. (2015), S. 99-100; Fleischmann und Meyr (2004), S. 298

²⁹Vgl. Goetschalckx und Fleischmann (2005), S. 117-118

³⁰Datengrundlage bilden u. a. Prognosen und Wirtschaftstrends.

³¹Vgl. Fleischmann und Meyr (2004), S. 298,

³²Vgl. Rohde und Wagner (2002), S. 143.

³³Vgl. Stadtler (2008), S. 217-218.

³⁴Vgl. Dickersbach (2004), S. 131.

len und die Logistik generiert.³¹ Zum Abschluss wird das Modul Bedarfsdeckung aufgeführt. Das Modul dient hauptsächlich der Kundenzufriedenheit, indem es die eintreffenden Kundenaufträge auf Realisierbarkeit prüft und die geforderte Lieferung der Leistung zusagt.³¹

Diese Module sind von Meyr et al. (2015) jedoch im Verhältnis eines typischen SCM-Prozess gesetzt. Es stellt sich die Frage, welche Systeme bei betriebliche Entscheidungen durch Einbeziehung von MRO-Prozessen in den SCM-Prozess relevant sind. Die strategische Netzwerkplanung scheint keinen direkten Einfluss auf MRO-Prozesse zu haben, da diese sich an der Grundleistung des Unternehmens orientiert. Anders formuliert bedeutet dies, dass nur dort eine Instandhaltung der Produkte angeboten wird, wo auch das eigentliche Produkt den Absatz findet. Inwieweit die Masterplanung ebenfalls den Materialfluss von MRO-Prozessen synchronisiert wird in dieser Arbeit nicht betrachtet. Aus dem Sachverhalt der Auftragsfertigung folgt, dass keine Distribution notwendig ist und dementsprechend finden die Distributionsplanung keine Relevanz bei MRO-Prozessen.

Bei MRO-Prozessen sind die Module der Absatzplanung, der Einkaufs- und Materialbedarfsplanung, der Produktionsplanung, die Transportplanung und der Bedarfsdeckung relevant. Ein effizientes APS bei MRO-Prozessen muss den erwarteten Absatz bzw. die erwarteten Auftragseingänge mittelfristig gut prognostizieren können. Aufbauend auf dieser Prognose kann eine Einkaufs- und Materialbedarfsplanung erfolgen. Dies erfolgt jedoch kurzfristig, da der Ressourcenbedarf abhängig der Kundenaufträge ist. Anschließend erfolgt bereits die Prüfung der Realisierbarkeit und abhängig dieser die Entscheidung über die Annahme des Auftrags über die Instandsetzung des Produkts mittels des Moduls der Bedarfsdeckung. Anschließend erfolgt die Produktions- und Transportplanung.

In dieser Arbeit wird der Fokus auf die Auftragsannahme mittels des Moduls der Bedarfsdeckung gelegt. Die Entscheidung, ob ein Auftrag zur Instandhaltung angenommen wird, ist abhängig der verfügbaren Ressourcenkapazität und des Lagerbestand an Teilen bzw. Fertigerzeugnissen (Produkten). Mit Ressourcen wird in dieser Arbeit entweder ein endlicher Bestand an Fertigungsteile zur Produktion der Produkte verstanden oder es handelt sich um regenerative Ressourcen, wie z. B. Maschinenkapazität oder Arbeitskraft. Erneuerbare Ressourcen zeichnen sich dadurch aus, dass sie mit Ablauf des Betrachtungszeitraums den vordefinierten Kapazitätswert wieder annehmen. Bei dem Lagerbestand handelt es sich wiederum um einen physischen Bestand an Teilen, die für die Fertigung bzw. Instandhal-

tung verwendet werden, oder um Fertigerzeugnisse, bei denen es sich um fertig erzeugte Leistungen in Form von Produkten handelt. Für die Entscheidungsträger muss das Modul der Bedarfsdeckung dementsprechend die Entscheidung über die Auftragsannahme unter Beachtung der Auftrags-, der Materialbedarfs- und der Produktionsplanung unterstützen. Im nächsten Kapitel wird eine OR-Methode zur klassischen Auftragsannahme bei konstanten Ressourcenkapazitäten vorgestellt und im Kapitel ?? wird dieses Modell um die Besonderheiten der MRO-Prozesse erweitert.

3 Das Konzept des Revenue Managements

3.1 Herkunft des Revenue Managements

Zur Entscheidungsunterstützung bei der Annahme von Kundenaufträgen wird in der aktuellen Forschung vermehrt auf das Konzept des Revenue Managements zurückgegriffen.⁵ Da eine kurzfristige Anpassung der mittelfristig bereitgestellten Kapazitäten einer Dienstleistungsproduktion an eine unsichere und schwankende Nachfrage nicht möglich ist, wird mit dem Konzept eine effizientere Auslastung der bestehenden Kapazitäten ermöglicht.⁶

Der Begriff *Revenue Management* wird im deutschsprachigen Raum meist mit *Ertragsmanagement* oder *Erlösmanagement* übersetzt.⁷ Yield Management wird als Synonym benutzt.⁸ Dabei greift der Begriff *Yield* zu kurz, da damit in der Luftverkehrsbranche der Erlös je Passagier und geflogener Meile bezeichnet wird.⁹ Der Term *Revenue Management* hat sich jedoch gegenüber Yield Management durchgesetzt, da der Yield (Durchschnittsertrag) sich theoretisch nur durch einen Passagier maximieren lässt und somit die Maximierung als Zielsetzung nicht sinnvoll ist.¹⁰ Erste Ansätze des RM sind in der Praxis entwickelt. Durch die Deregulierung des amerikanischen Luftverkehrsmarktes im Jahr 1978 mussten die traditionellen Fluggesellschaften ihre Wettbewerbsfähigkeit gegenüber Billiganbietern erhöhen und entwickelten das frühe RM.¹¹

Kimms und Klein (2007) versuchen durch eine umfangreiche Diskussion einige Erklärungsansätze aufzuzeigen (Warum? Wovon?). Das RM hat vor allem aus

⁵Vgl. Klein (2001), S. 246.

⁶Vgl. Spengler und Rehkopf (2005), S. 124.

⁷Vgl. z. B. Zehle (1991), S. 486

⁸Vgl. z. B. Kolisch und Zatta (2006), S. 319

⁹Vgl. z. B. Weatherford (1998), S. 69

¹⁰Vgl. Klein und Steinhardt (2008), S. 6; Spengler und Rehkopf (2005), S. 124-125.

¹¹Vgl. Petrick (2009), S. 1-3

dem älteren, englischsprachigen Bereich einen engen Bezug zu konkreten Anwendungsgebieten. Die Autoren zeigen auf, dass viele Autoren versuchen das komplexe Konzept des Revenue Managements in einer kurzen Erklärung zu überführen. Dieses läuft letztlich darauf hinaus, dass diese Autoren einige situative Merkmale und Instrumente des Managements vermischen, gleichzeitig aber versuchen, die Zielsetzung festzulegen und das Anwendungsgebiet auf bestimmte Branchen zu beschränken.

Weiter wird in der Literatur der Begriff des RM unterschiedlich definiert. Frieger (1996) bezeichnet das RM als *Preis-Mengen-Steuerung*, Daudel und Vialle (1992) als *Preis-Kapazitäts-Steuerung* und Talluri und van Ryzin (2004b) verstehen es als das gesamte *Management der Nachfrage*. Die beiden ersten Definitionen können als Synonym für eines der Instrumente des RM stehen und daher finden diese für das gesamte Konzept keine weitere Verwendung.¹² Nachfolgend wird die Definition von Klein (2001, S. 248) aufgegriffen:

„Revenue Management umfasst eine Reihe von quantitativen Methoden zur Entscheidung über Annahme oder Ablehnung unsicherer, zeitlich verteilt eintreffender Nachfrage unterschiedlicher Wertigkeit. Dabei wird das Ziel verfolgt, die in einem begrenzten Zeitraum verfügbare, unflexibel Kapazität möglichst effizient zu nutzen.“

Petrick (2009) definiert das RM als Ziel einer Unternehmung die Gesamterlöse zu maximieren, die sich aufgrund der speziellen Anwendungsgebiete ergeben. Damit definiert Petrick (2009) das RM als Zusammenfassung aller Interaktionen eines Unternehmens, die mit dem Markt, also der Absatz- oder Nachfrageseite, zusammenhängen. Im Kern lassen sich drei wichtige Perspektiven für eine Definition des Revenue Managements nach Petrick (2009), Stuhlmann (2000), Corsten und Stuhlmann (1999) übernehmen:

1. Ziel ist es die Gesamterlöse unter möglichst optimaler Auslastung der vorhandenen Kapazitäten zu maximieren.
2. Durch eine aktive Preispolitik wird das reine Kapazitäts- oder Auslastungsmanagement unterstützt.
3. Für die erfolgreiche Implementierung des Revenue Managements ist eine umfangreiche Informationsbasis notwendig. Es muss u. a. eine möglichst gute

¹²Vgl. z. B. Petrick (2009)

Prognose über die zukünftige Nachfrage und Preisbereitschaft der Kunden vorhanden sein.

Kimms und Klein (2007) weisen darauf hin, dass eine differenzierte Betrachtung des Konzepts notwendig ist: Einerseits im Hinblick auf die **Anwendungsvoraussetzungen** und andererseits im Hinblick auf die **Instrumente des Revenue Managements**, damit verdeutlicht dargestellt ist, in welchen Branchen das RM Potentiale liefert. Dabei sollten branchenspezifische Besonderheiten, neben den zahlreichen Ähnlichkeiten Berücksichtigung finden, sowie das begrenzte Kapazitätenkontingent, damit die Potentiale des RM zur Maximierung der Gesamterlöse in den Dienstleistungsbranchen erfolgen kann.¹³ In dem nachfolgenden Kapitel wird dieser Empfehlung gefolgt und die Anwendungsvoraussetzungen sowie Instrumente des RM vorgestellt.

3.2 Anwendungsvoraussetzungen und Instrumente des Revenue Managements

Petrick (2009) weist darauf hin, dass anhand von speziellen Anwendungsvoraussetzungen geprüft wird, ob das RM für die jeweilige Situation des Unternehmens (oder die gesamte Branche) zur Maximierung des Gesamterlöses beiträgt. Kimes (1989b) definiert die in der Literatur häufigsten Anwendungsvoraussetzungen:¹⁴

- „weitgehend fixe“ Kapazitäten
- „Verderblichkeit“ bzw. „Nichtlagerfähigkeit“ der Kapazitäten und der Leistung
- Möglichkeit zur Vorausbuchung von Leistungen
- stochastisch, schwankende Nachfrage
- hohe Fixkosten für die Bereitstellung der gesamten Kapazitäten bei vergleichsweise geringen variablen Kosten für Produktion einer Leistungseinheit
- Möglichkeit zur Marktsegmentierung und im Ergebnis dessen zur segment-

¹³Vgl. z. B. von Martens (2009), S. 11-24

¹⁴Vgl. u. a. Friege (1996), S. 616-622, und Weatherford und Bodily (1992), 831-832.

orientierten Preisdifferenzierung

Aufgrund der Anwendungsvoraussetzungen des RM kann das Konzept auch auf die Auftragsfertigung bzw. MRO-Prozesse übertragen werden.¹⁵ Bei der Auftragsfertigung wird ein bestimmter Buchungszeitraum betrachtet, in dem weitestgehend von fixen Kapazitäten der Ressourcen ausgegangen werden kann. Durch Ablauf des Buchungshorizonts verfällt die Kapazität, da es sich hauptsächlich um erneuerbaren Kapazitäten handelt (Arbeitskraft, Maschinenkapazität, usw.). Anders formuliert, die Ressourcen werden zur nächsten Leistungserstellung erneuert. Dabei beschreibt t einen Zeitpunkt des Buchungshorizonts. Die Gesamtlänge des Buchungshorizonts entspricht T und verläuft rückwärts bis zur Leistungserstellung des betrachteten Buchungshorizonts. Da der Buchungshorizont mit genau einer Leistungserstellung gekoppelt ist, können diese mit einem Parameter i beschrieben werden. Mit der Leistungserstellung werden die verbuchten Kapazitäten aus dem Buchungshorizont durch die Auftragsproduktion beansprucht. Mit dem Start der Leistungserstellung erfolgt ebenfalls die Vorausbuchungszeit für die nachfolgende Leistungserstellung $i + 1$. Bei der Leistungserstellung $i + 1$ sind die Kapazitäten der Ressourcen regeneriert. Abbildung 4 zeigt den Zusammenhang von Buchungshorizont, Leistungserstellung und abnehmenden Ressourcenkapazität.

??? Klein und Steinhardt (2008) setzen sich mit den Anwendungsvoraussetzungen von mehreren Autoren auseinander. Sie konnten Gemeinsamkeiten innerhalb der Definitionen der Autoren finden, aber zeigten auch die Unterschiede und die Kritiken auf. In ihrer Arbeit übernehmen sie die Anwendungsvoraussetzung von Corsten und Stuhlmann (1998): "Marktseitige Anpassungserfordernis steht unternehmensseitig unzureichendes Flexibilitätspotential hinsichtlich der Kapazität – bezogen auf Mittel- oder Zeitaufwand – gegenüber". Zugleich weisen sie jedoch darauf hin, dass zum Verständnis eines komplexen und interdisziplinären Ansatzes auch die Definitionen anderer Autoren im Hinblick auf das Verständnis der Anwendungsvoraussetzungen beitragen. ???

Auf Grundlage der von Friege (1996) beschriebenen Anwendungsvoraussetzungen hat Petrick (2009) drei Instrumente des RM bestimmt. Die Instrumente benötigen als Grundlage *Daten der Prognose*, damit sie zur Anwendung kommen.¹⁶ Zu den Instrumenten zählen die **segmentorientierte Preisdifferenzierung**, die **Kapazitätensteuerung** und die **Überbuchungssteuerung**. Es lassen sich un-

¹⁵Vgl. Hintsches et al. (2010), S.176-178; Kimes (1989b), S. 349-351.

¹⁶Die Prognose zählt laut Petrick (2009) nicht als eigenständiges Instrument des RM.

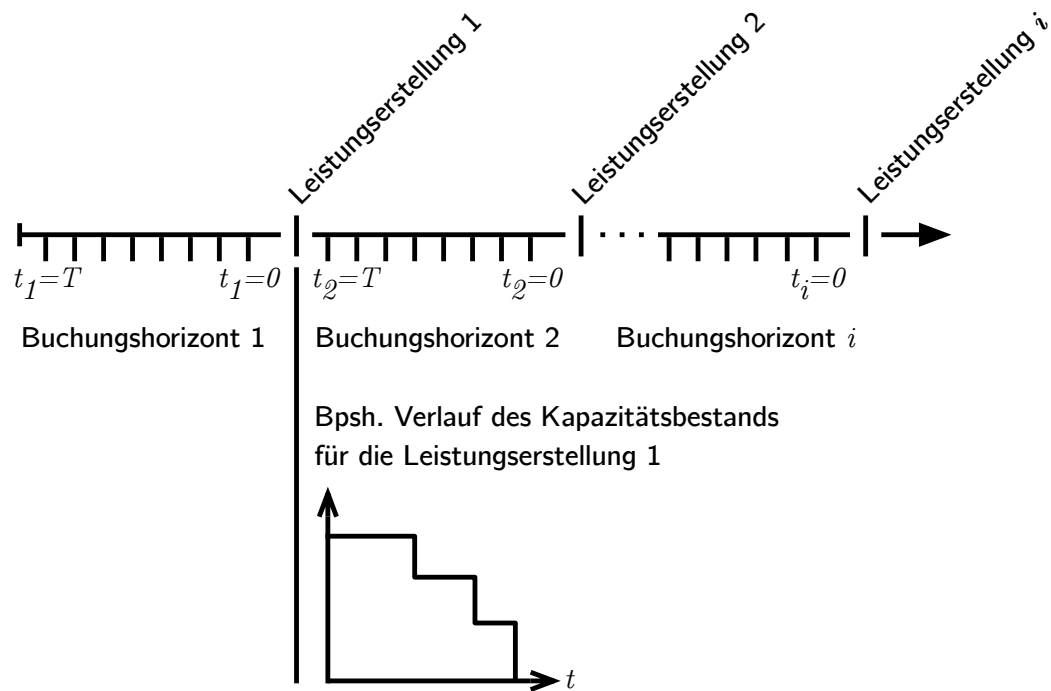


Abbildung 3 Grafische Darstellung des Buchungshorizonts bei der Auftragsfertigung
In Anlehnung an: ???.

terschiedliche Abhängigkeiten der Instrumente untereinander ermitteln.¹⁷

Erklärung segmentorientierte Preisdifferenzierung, Kapazitätssteuerung, Überbuchungssteuerung?

3.3 Mathematische Modellformulierung des Revenue Managements

Im Folgenden wird das dynamisch, stochastische Grundmodell des RM nach Talluri und van Ryzin (2004, S. 18-19) beschrieben. Ein Dienstleistungsnetzwerk eines Anbieters benötigt jeweils zur Erstellung der Dienstleistungen ein bestimmtes Kontingent an Ressourcen aus einer Menge an Ressourcen $\mathcal{H} = \{1, \dots, l\}$. Der Index h beschreibt dabei eine jeweilige Ressource und der Index l die gesamte Anzahl an möglichen Ressourcen. Die jeweilige Kapazität einer Ressource $h \in \mathcal{H}$ ist durch den Parameter c_h beschrieben und die gesamten Kapazitäten der Ressourcen ist

¹⁷Als Beispiel baut die Kapazitätensteuerung auf den Ergebnissen der Preisdifferenzierung auf und die Überbuchungssteuerung kann selten ohne Kapazitätensteuerung gelöst werden.

als Vektor $\mathbf{c} = (c_1, \dots, c_h, \dots, c_l)$ formuliert. Eine Anfrage nach einem Produkt (Dienstleistung) in dem Netzwerk ist durch den Parameter j aus der Menge an möglichen Produkthanfragen $\mathcal{J} = \{1, \dots, n\}$ beschrieben. Die gesamte Anzahl an Produkthanfragen ist durch den Parameter n definiert. Sobald eine Produkthanfrage $j \in \mathcal{J}$ akzeptiert und somit abgesetzt ist, fällt für den Absatz der Ertrag r_j an. Der jeweilige Verbrauch einer Ressource h durch Annahme einer Anfrage nach einem Produkt j ist anhand des Parameters a_{hj} beschrieben. Durch Vektorschreibweise kann der Ressourcenverbrauch für eine Anfrage nach einem Produkt j als Vektor $\mathbf{a}_j = (a_{1j}, \dots, a_{hj}, \dots, a_{lj})$ formuliert werden. Der Buchungshorizont entspricht T Perioden und kann jeweils in einzelne Perioden $t = 1, \dots, T$ aufgeteilt werden. Dabei muss Beachtung finden, dass der Buchungshorizont T gegenläufig verläuft. Die Wahrscheinlichkeit der Nachfrage eines Produkts j in der Periode t entspricht $p_j(t)$ und die Wahrscheinlichkeit, dass keine Nachfrage in der Periode t eintrifft, entspricht $p_0(t)$. Es gilt $\sum_{j \in \mathcal{J}} p_j(t) + p_0(t) = 1$ und somit kann $p_0(t)$ durch den Term $p_0(t) = 1 - \sum_{j \in \mathcal{J}} p_j(t)$ für die Periode t ermittelt werden.¹⁸ Die noch erwartete Nachfrage D_{jt} für ein bestimmtes Produkt j für eine beliebige Periode t lässt sich durch $\sum_{\tau=1}^t p_j(\tau)$ aggregieren.

Mit den vorangegangenen Parametern kann der maximal erwartete Ertragswert $V(\mathbf{c}, t)$ für eine Periode t bei einer noch vorhandenen Ressourcenkapazität \mathbf{c} als Bellman-Gleichung formuliert werden (**DP-op**):¹⁹

$$V(\mathbf{c}, t) = \sum_{j \in \mathcal{J}} p_j(t) \max[V(\mathbf{c}, t-1), r_j + V(\mathbf{c} - \mathbf{a}_j, t-1)] + p_0(t)V(\mathbf{c}, t-1) \quad (1)$$

Es handelt sich hier um die Modellformulierung der Dynamischen Programmierung (DP) im Netzwerk RM. Das Konzept der DP wurde von Bellman entwickelt und dient der Ermittlung der optimalen Politik in Bezug auf den aktuellen Zustand eines Systems.²⁰ Dabei bildet jeder Erwartungswert $V(\mathbf{c}, t)$ mit der Ressourcenkapazität \mathbf{c} zum Zeitpunkt t einen Systemzustand des Netzwerks ab. Eine derartige Formulierung eines Optimierungsproblems wird oft als sogenannte *Bellman'sche Funktionsgleichung oder Bellman-Gleichung* bezeichnet.²¹

¹⁸Vgl. Talluri und van Ryzin, S. 18

¹⁹???

²⁰Vgl. Bellman (1954), S. 4-5

²¹???

Die Gleichung weist die Grenzbedingungen

$$V(\mathbf{c}, 0) = 0 \text{ wenn } \mathbf{c} \geq 0 \text{ sowie} \quad (2)$$

$$V(\mathbf{c}, t) = -\infty \text{ wenn } c_j < 0 \forall j \in \mathcal{J} \quad (3)$$

auf, da eine jeweilig verbleibende Kapazität nach Bereitstellung des Produkts wertlos und eine negative Ressourcenkapazität nicht möglich ist.

Die Gleichung (1) lässt sich umformen, indem die Entscheidungen über die Annahme und Ablehnung einer Produkthanfrage separiert wird. Da $\sum_{j \in \mathcal{J}} p_j(t) + p_0(t) = 1$ gilt, kann die Gleichung weiter vereinfacht werden:²²

$$\begin{aligned} V(\mathbf{c}, t) &= \sum_{j \in \mathcal{J}} p_j(t) V(\mathbf{c}, t-1) + \sum_{j \in \mathcal{J}} p_j(t) \max[0, r_j - V(\mathbf{c}, t-1) \\ &\quad + V(\mathbf{c} - \mathbf{a}_j, t-1)] + p_0(t) V(\mathbf{c}, t-1) \\ &= V(\mathbf{c}, t-1) + \sum_{j \in \mathcal{J}} p_j(t) \max[0, \\ &\quad r_j - V(\mathbf{c}, t-1) + V(\mathbf{c} - \mathbf{a}_j, t-1)] \end{aligned} \quad (4)$$

Eine eintreffende Anfrage nach einem Produkt j ist demnach dann akzeptiert, wenn der Ertrag r_j größer gleich der Differenz des Erwartungswertes des Ertrags unter der Prämisse der Annahme der Produkthanfrage und des Erwartungswertes des Ertrag unter der Prämisse der Ablehnung der Produkthanfrage ist:

$$r_j \geq V(\mathbf{c}, t-1) - V(\mathbf{c} - \mathbf{a}_j, t-1) \quad (5)$$

Dabei kann der rechte Term (5) als Opportunitätskosten (OK) der Auftragsannahme angesehen werden:

$$OC_j = V(\mathbf{c}, t-1) - V(\mathbf{c} - \mathbf{a}_j, t-1) \quad (6)$$

Somit erfolgt die Akzeptanz einer Anfrage nach einem Produkt $j \in \mathcal{J}$ ausschließlich nur dann, sofern die OK des Ressourcenverbrauchs niedriger als der Ertrag ist. Der maximal mögliche Erwartungswert unter Beachtung des Kapazität \mathbf{c} zum Zeitpunkt t ist damit der Erwartungswert unter der Prämisse der Ablehnung der Anfrage zum nächsten Zeitpunkt $t-1$ inkl. der Summe der Erträge abzgl. der OK

²²Vgl. Spengler et al. (2007), S. 161.

durch Annahme der möglichen Anfragen über aller Produkte $j \in \mathcal{J}$ und lässt sich mathematisch wie folgt definieren:

$$V(\mathbf{c}, t) = V(\mathbf{c}, t - 1) + \sum_{j \in \mathcal{J}} p_j(t) \max[0, r_j - OC_j] \quad (7)$$

Zur Veranschaulichung des Netzwerk RM wird ein Netzwerk mit zwei Produkten $j \in \mathcal{J}$ und zwei Ressourcen $h \in \mathcal{H}$ betrachtet. Die Ressource $h = 1$ hat eine Kapazität von $c_1 = 2$ und die Ressource $h = 2$ hat eine Kapazität von $c_2 = 1$. Zur Ausführung des Produkt $j = 1$ wird die Ressource $h = 1$ mit einer Einheit benötigt und zur Ausführung des Produkt $j = 2$ wird wiederum eine Einheit der Ressource $h = 2$ gebraucht. Damit gilt $a_{11} = 1$ und $a_{22} = 1$. Durch Annahme einer Anfrage nach Produkt $j = 1$ wird der Ertrag $r_1 = 100$ und durch Annahme von Produkt $j = 2$ ein Ertrag von $r_2 = 200$ generiert. Der Buchungshorizont entspricht $T = 4$. Die Wahrscheinlichkeiten des Eintreffens einer Anfrage nach Produkt $j = 1$ über die Buchungsperioden $t \in T$ lässt sich als Vektor $p_1(t) = (0.5, 0.5, 0.5, 0.5)$ beschreiben. Analog lassen sich die Wahrscheinlichkeiten für das Eintreffen der Produkthanfragen $j = 2$ als Vektor $p_2(t) = (0.1, 0.1, 0.1, 0.1)$ definieren. Die Gegenwahrscheinlichkeiten, dass keine Anfragen eintreffen, lassen sich mit $p_0(t) = 1 - \sum_{j \in \mathcal{J}} p_j(t)$ berechnen und bilden den Vektor $p_0(t) = (0.4, 0.4, 0.4, 0.4)$. Durch Vereinfachung der Gleichung (1) zur Gleichung (4) werden die Gegenwahrscheinlichkeiten $p_0(t)$ nicht mehr benötigt und im weiteren Verlauf der Arbeit nicht mehr berücksichtigt.

Die Parameter lassen sich damit abschließend wie folgt definieren:

$$j = \{1, 2\}, h = \{1, 2\}, r_1 = 100, r_2 = 200, \text{Startperiode } t = 4,$$

$$\mathbf{c} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \mathbf{a}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{a}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, p_1(t) = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}, p_2(t) = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}$$

Die mathematische Modellformulierung des stochastisch, dynamisches Programms aus Gleichung (4) lässt sich als Graph darstellen, der die einzelnen Systemzustände für die DP aufzeigt.²³ Der Erwartungswert $V(\mathbf{c}, t)$ ist abhängig vom Erwartungswert $V(\mathbf{c}, t - 1)$ und vom Erwartungswert $V(\mathbf{c} - \mathbf{a}_j, t - 1)$. Der Erwartungswert

²³Vgl. DeMiguel und Mishra (2006), S. 8-13.

$V(\mathbf{c}, t - 1)$ ist wiederum abhängig vom Erwartungswert $V(\mathbf{c}, t - 2)$ und vom Erwartungswert $V(\mathbf{c} - \mathbf{a}_j, t - 2)$, usw. Abbildung 4 zeigt die erste rekursive Folge für die Gleichung (4) als gerichteten Graphen auf und im nachfolgenden wird auf die Notation eingegangen.

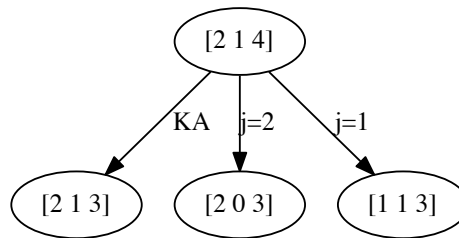


Abbildung 4 Beispielhafte Darstellung einer rekursive Folge eines Netzwerk RM

Ein Knoten repräsentiert einen Systemzustand des Netzwerks mit den vorhandenen Kapazitäten \mathbf{c} zum Zeitpunkt t . Bei dem Systemzustand handelt es sich um ein Teilproblem des Modells. Bei der Benennung eines solchen Knotens wird eine Zahlenfolge verwendet, bei der die ersten Einträge die Ressourcenkapazität \mathbf{c} in Länge der Ressourcen h entsprechen und der letzte Eintrag den Zeitpunkt t aufzeigt. Bspw. zeigt der Startknoten des Beispiels die Zahlenfolge $[2 \ 1 \ 4]$, da das Netzwerk noch die volle Ressourcenkapazität $\mathbf{c} = (2, 1)$ aufweist und sich im Zeitpunkt $t = 4$ befindet. Von diesem Systemzustand können jetzt nachfolgende Systemzustände abhängig der Produkthanfragen erreicht werden. In diesem Netzwerk gibt es zwei Produkte j und durch betrachten der Gleichung (4) wird klar, dass drei Optionen zum Erreichen des nachfolgenden Systemzustands zum Zeitpunkt $t - 1 = 3$ möglich sind. Diese Optionen bilden die Kanten des Graphen. Es kann keine Anfrage nach einem Produkt j eintreffend, dann wird der nachfolgende Systemzustand $[2 \ 1 \ 3]$ erreicht. Alternativ können Anfragen nach Produkt $j = 1$ oder $j = 2$ eintreffen und dementsprechend müssen die vorhandenen Kapazitäten \mathbf{c} um den Ressourcenverbrauch $a_{11} = 1$ bzw. $a_{22} = 1$ reduziert werden. Daraus folgt, dass der Systemzustand $[1 \ 1 \ 3]$ bzw. $[2 \ 0 \ 3]$ im Netzwerk erreicht werden kann. Die optimale Politik eines solchen Graphen wird abgebildet durch die Kanten. Sofern der Übergang des Systemzustands durch Annahme der jeweiligen Anfrage j gegen die Bedingung (6) verstößt, wird die Kante des Graphen als gestrichelter Pfeil dargestellt. Ein Übergang über die Kante zum nächsten Systemzustand wäre theoretisch möglich, aber entspricht nicht der optimalen Politik.

Aufbauen auf dieser rekursiven Logik wird ein gerichteter und gewichteter Multigraph aufgebaut. Er zeigt alle möglichen Systemzustände des Netzwerks. Die Rekursion wird abgebrochen, sofern ein Systemzustand aufgrund der Grenzbedingungen aus den Gleichungen (2) oder (3) nicht möglich ist. Abbildung 5 zeigt die möglichen Systemzustände für das eingeführte Beispiel.

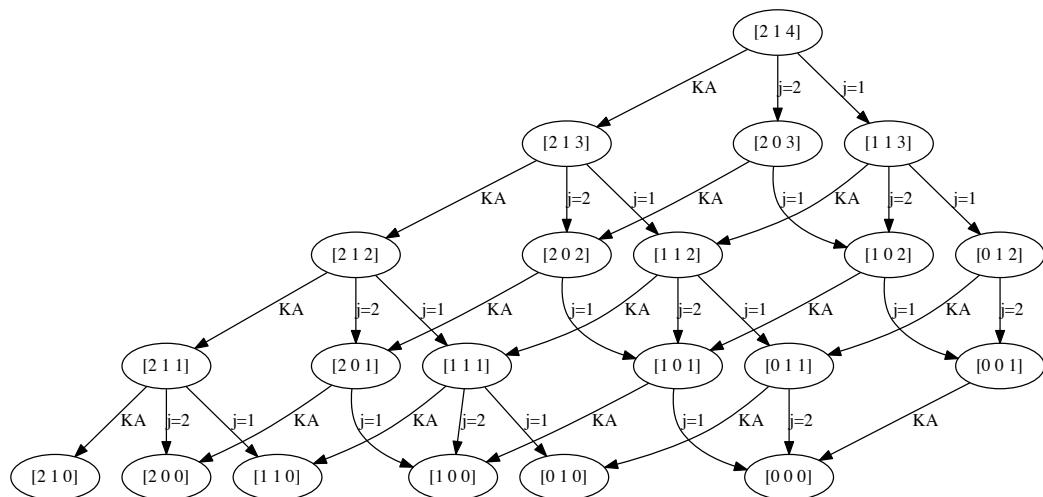


Abbildung 5 Darstellung der Systemzustände des beispielhaften Netzwerk RM

Anhand dieses Graphen mit den möglichen Systemzuständen aus der Abbildung 5 wird ersichtlich, welche Erwartungswerte (Teilprobleme) für das Entscheidungsproblem der Auftragsannahme durch Rückwärtsinduktion gelöst werden müssen.²⁴ Dafür wird im ersten Schritt die Grenzbedingung aus Gleichung (2) betrachtet. Alle Systemzustände zum Zeitpunkt $t = 0$ nehmen den Erwartungswert $V(\mathbf{c}, t = 0) = 0$ an. Mit dieser Bedingung lassen sich die zeitlich zuvorkommenden Systemzustände mit $t = 1$ berechnen. Es wird die Gleichung (4) angewendet, wobei beachtet werden muss, dass nicht für alle Systemzustände mit $t = 1$ alle Produkthanfragen möglich sind. Dies folgt aus der Grenzbedingung aus Gleichung (3). Zur Veranschaulichung wird der Erwartungswert des Systemzustands $[1 \ 0 \ 1]$ mittels der Gleichung (4)

²⁴Vgl. ???, S. ???.

Tabelle 1 Ergebnistabelle für das beispielhafte Netzwerk RM

c_1	c_2	t	ExpValue	$j = 1$	$j = 2$
2	1	4	231.25	1	1
2	1	3	191.6875	1	1
2	0	3	137.5	1	0
1	1	3	141.703125	1	1
1	0	2	75.0	1	0
2	1	2	138.0	1	1
0	1	2	38.0	0	1
2	0	2	100.0	1	0
1	1	2	113.0	1	1
1	1	1	70.0	1	1
2	0	1	50.0	1	0
0	1	1	20.0	0	1
2	1	1	70.0	1	1
1	0	1	50.0	1	0
0	1	0	0.0	0	0
2	1	0	0.0	0	0
1	0	0	0.0	0	0
1	1	0	0.0	0	0
2	0	0	0.0	0	0
0	0	0	0.0	0	0

berechnet.

$$\begin{aligned}
V(1, 0, 1) &= V(1, 0, 0) + p_1(1) \max[r_1 - V(1, 0, 0) + V(0, 0, 0), 0] \\
&\quad + p_2(1) \max[r_2 - V(1, 0, 0) + V(0, -1, 0), 0] \\
&= 0 + 0,5 \cdot \max[100 - 0 + 0, 0] + 0,1 \cdot \max[200 - 0 + (-\infty), 0] \\
&= 0 + 0,5 \cdot 100 + 0,1 \cdot 0 \\
&= 50
\end{aligned}$$

Nach dieser Vorgehensweise der Rückwärtsinduktion erfolgt die Ermittlung aller Erwartungswerte der möglichen Systemzustände. Tabelle 1 zeigt für alle möglichen Systemzustände $[c_1 \ c_2 \ t]$ den Erwartungswerten (ExpValue) und die optimale Politik. Die optimale Politik ist in den letzten zwei Spalten der Tabelle 1 jeweils für die zwei möglichen Anfragen $j \in \mathcal{J}$ aufgeführt. Sofern eine Anfrage nach dem Produkt j eintrifft, zeigt die Tabelle mit dem Wert „1“, dass die Anfrage unter den Bedingungen der vorhandenen Kapazitäten c und der verbleibenden Perioden $t \in T$ angenommen werden soll. Sofern die Anfrage j nicht der optimalen Politik entspricht oder in diesem Systemzustand nicht möglich ist, dann ist in den Spalten eine „0“ vermerkt.

Die optimale Politik lässt sich anhand der Gleichung (6) für jeden Systemzustand ermitteln. Für jede Kante des Graphens in Abbildung 5 kann damit der Ertrag abzgl. der OK hinterlegt werden ($r_j - OC_j$). Die optimale Politik im betrachteten Graphen der Abbildung 5 ist demnach die Kante bei der ein Ertrag größer gleich der OK erzielt wird. Abbildung 6 zeigt alle Systemzustände und Übergänge als Graphen mit den jeweiligen Erwartungswerten als Knoten und den Wert $r_j - OC_j$ als Kante. Dabei ist zu erkennen, dass zur optimalen Politik alle Übergänge bzw. die Entscheidungen der Annahme der jeweiligen Anfragen gehören.

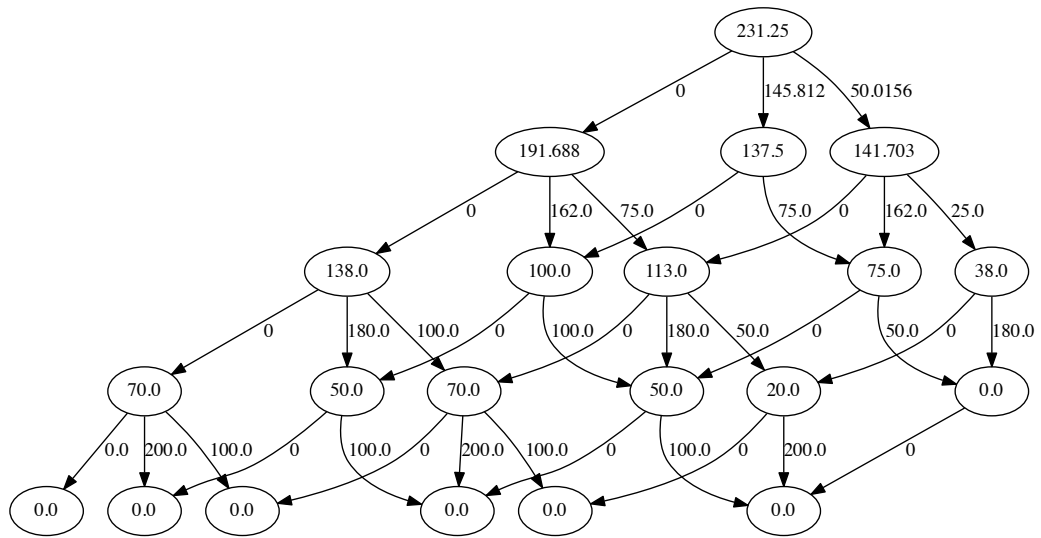


Abbildung 6 Darstellung der Systemzustände des beispielhaften Netzwerk RM (optimaler Auftragseingang)

Bei einem Fallbeispiel mit solchen Parameterwerten ist damit nur eine Auswertung des Erwartungswertes über den gesamten Verlauf des Buchungshorizonts T möglich. Zu erkennen ist, dass keine Anfrage zu einem beliebigen Systemzustand abgelehnt wird. Bei einem derartigen Ergebnis kann jedoch strategische Entscheidungen getroffen werden. Z. B. kann ein Unternehmen versuchen seine Instrumente der Marktbearbeitung anzupassen, damit ein bester Pfad der Auftragseingänge abgearbeitet wird. Der beste Pfad für das Eintreffen der Produkthanfragen lautet $[2 \ 1 \ 4] \rightarrow_{j=2} [2 \ 0 \ 3] \rightarrow_{j=1} [1 \ 0 \ 2] \rightarrow_{j=1} [0 \ 0 \ 1] \rightarrow_{KA} [0 \ 0 \ 0]$. Mit geschickter Ausgestaltung der Instrumente der Marktbearbeitung wird die prognostizierte Verteilung der Auftragseingänge zum Irrtum.

Anhand des Beispiels wird klar, dass die Prognose über die Wahrscheinlichkeiten

des Eintreffens einer Anfrage nach den Produkten $j \in \mathcal{J}$ die Erwartungswerte stark beeinflussen, sowie der potentielle Ertrag r_j und die OC_j . Dies kann durch das Abwandeln der Parameter gezeigt werden:

$$j = \{1, 2\}, h = \{1\}, r_1 = 100, r_2 = 200, T = 4$$

$$c_1 = 2, a_{11} = 1, a_{12} = 1, p_1(t) = \begin{pmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \end{pmatrix}, p_2(t) = \begin{pmatrix} 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \end{pmatrix}$$

Bei dieser beispielhaften Ausgestaltung der Parameter konkurrieren die Typen der Produkthanfragen j mit der einzigen verfügbaren Ressource h . Der nachfolgende Abbildung 7 zeigt den zugehörigen Graphen mit den Systemzuständen, den möglichen Entscheidungen und der grafischen Darstellung der optimalen Politik und die Tabelle 2 zeigt die berechneten Erwartungswerte sowie die optimale Politik für jeden Systemzustand $[c_1, t]$.

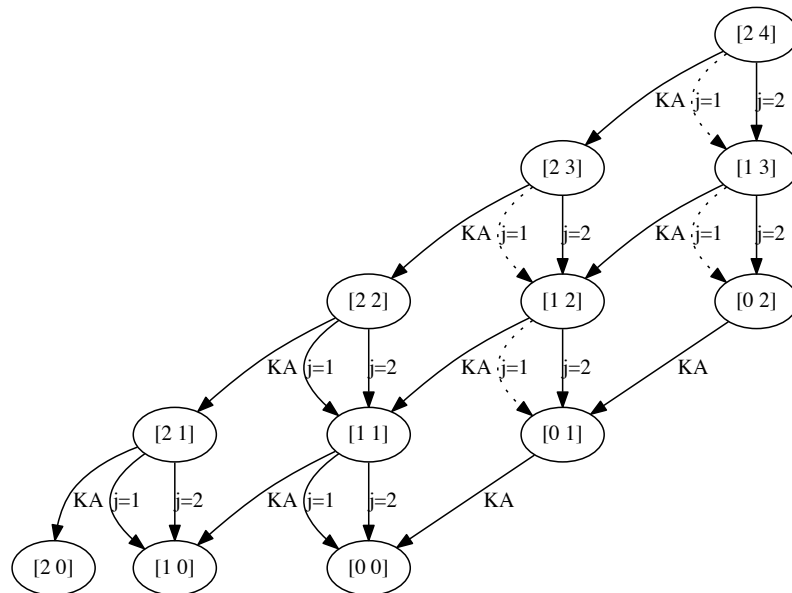


Abbildung 7 Darstellung der Systemzustände des Netzwerk RM mit konkurrierenden Anfragen

Tabelle 2 Ergebnistabelle für das beispielhafte Netzwerk RM mit konkurrierenden Anfragen

c_1	t	ExpValue	$j = 1$	$j = 2$
2	4	397.125	0	1
2	3	388.796875	0	1
1	3	199.19921875	0	1
2	2	360.0	1	1
1	2	196.0	0	1
2	1	180.0	1	1
1	1	180.0	1	1
0	1	0.0	0	0
2	0	0.0	0	0
1	0	0.0	0	0
0	0	0.0	0	0

Mit dem Beispiel wird klar, dass aufgrund des höheren Ertrags und der höheren Wahrscheinlichkeit $p_j(t)$ über den gesamten Verlauf des Buchungshorizonts T die Annahme des Produktauftrags $j = 2$ im Systemzustand [3 4] die optimale Politik ist. Sofern eine Anfrage nach dem Produkt $j = 2$ eintrifft, dann sollte diese abgelehnt werden. Dies resultiert aus der Tatsache, dass die OC_j den Ertrag r_j des Auftrags übersteigen. In den OC_j des Auftrags $j = 2$ sind die nachfolgenden Erwartungswerte gebündelt. Aufgrund der Parameterausprägung für das Netzwerk (Wahrscheinlichkeitsverteilung $p_j(t)$ und der potentielle Ertrag r_j für die Anfragen $j \in \mathcal{J}$) ist die potentielle Annahme einer Anfrage $j = 2$ möglich und für den Gesamterlös des Netzwerks ertragreicher. Jedoch ist in der Abbildung 7 ebenfalls zu erkennen, dass sofern am Anfang des Buchungshorizonts keine Anfrage eintreffen, dass die optimale Politik sich verändert. Die OC_j einer Anfrage nach dem Produkt $j = 1$ nehmen im Zeitverlauf ab und wird die Option der Annahme dieser Anfrage zur optimalen Politik aufgenommen. Für eine strategische Entscheidung ist der beste Pfad des Netzwerks $[2\ 4] \rightarrow_{j=2} [1\ 3] \rightarrow_{j=2} [0\ 2] \rightarrow_{KA} [0\ 1] \rightarrow_{LA} [0\ 0]$. Ein Unternehmen muss damit das Ziel verfolgen, dass die Verteilung der Auftragseingänge sich zu Gunsten der Produkthanfrage $j = 2$ bewahrheiten. Damit wäre ein Gesamterlös in Höhe von 300 GE generiert.

4 Bestehende Ansätze zur Annahme von Aufträgen in der Auftragsfertigung und bei Instandhaltungsprozessen

Das Konzept des Revenue Managements (RM) zur Annahme von Aufträgen bei Dienstleistungen findet bereits über mehrere Dekaden in der wissenschaftlichen Literatur Anwendung.⁵ Neue Veröffentlichungen versuchen das Konzept auf die Problemstellung der Annahme von Anfragen der Auftragsfertigung bzw. Kundeneinzelfertigung zu übertragen.⁶ Wie in Kapitel 2 dargelegt, kann der Instandhaltungsprozess eines Dienstleistungsunternehmens einer Auftragsfertigung gleichgesetzt werden. Kimms und Klein (2007) geben einen Überblick über das traditionelle Konzept des RM über verschiedene Branchen. Dabei schreiben die Autoren, dass das Konzept des RM vermehrt Anwendung findet, damit Unternehmen eine Unterstützung in der Entscheidungsfindung erhalten, welche Aufträge zur Auftragsfertigung akzeptiert werden sollen.⁷ Quante (2009) gibt einen Überblick über relevante Literatur des traditionellen RM. Tabelle 3 zeigt eine Anlehnung der Übersicht von Quante (2009) mit den Publikationen zum traditionellen RM in der Fertigungsindustrie. Die Tabelle zeigt jeweils zur Publikation den Kundenauftragskoppelpunkt, die Anzahl der berücksichtigten Konsumerklassen und die Methode aufgeführt ist. Die Konsumerklassen resultieren aus der für das Konzepts des Revenue Management notwendigen Marktsegmentierung von Kunden bzw. Auftragsstypen. Einige Konzepte und Modelle berücksichtigen daher explizit die Anzahl solcher Klassen.

H. deB. Harris und Pinder (1995) beziehen die RM-Komponenten der differenzierten Preispolitik und eine multiklassen Kapazitätsallokation in ihr Modell mit ein. Sie zeigen, dass sofern stochastische Nachfrage auf fixe und kurzfristige Kapazität trifft, dass es zu Lagerfehlbeständen bei Niedrig-Preis-Segmenten führt. Jedoch

⁵Vgl. Klein (2001), S. 246 NACHLESEN SONST FALSCH!!!!

⁶Vgl. ???

⁷Vgl. Kimms und Klein (2007), S. 1

Tabelle 3 Überblick über Publikationen des traditionellen Konzepts des Revenue Managements in der Fertigungsindustrie

Autoren	KAKP	#Klassen	Methode
H. deB. Harris und Pinder (1995)	ATO	2	K, M
Kalyan (2002)	MTO/ATO/MTS	–	K
Rehkopf und Spengler (2005)	MTO	mehrere	K, M
Rehkopf (2007)	MTO	–	K, M, L
Spengler et al. (2007)	MTO	mehrere	M
Volling et al. (2012)	MTO	–	K, M
Kimms und Klein (2005)	MTO	–	K
Guhlich et al. (2015)	ATO	–	M, L
Kolisch und Zatta (2006)	MTO	–	K
Barut und Sridharan (2005)	MTO	mehrere	M
Kuhn und Defregger (2004)	MTO	2	F
Specht und Gruß (2008)	ATO	–	K, F
Quante et al. (2009)	MTO	–	L
Cheraghi et al. (2010)	MTO/MTS	–	L
Sucky (2009)	MTO	–	M, F

In Anlehnung an: Quante (2009), S. 44.

Legende: KAKP: Kundenauftragskoppelpunkt, F: Fallstudie, K: Konzeption, L: Literaturüberblick, M: Simulations-/Analysemodell.

rechtfertigen diese Lagerfehlbestände eine Premiumpreisstrategie bei den Kundengruppen mit höherer Preisbereitschaft, was letztendlich zu Umsatzsteigerungen führt.⁸ Kalyan (2002) beschäftigt sich mit der Bestätigung der Anwendungsvoraussetzungen des RM für die Auftragsfertigung. Das vom Autor beschriebene Konzept sieht die Einführung eines minimalen Akzeptanzwert vor. Sofern dieser Wert bekannt ist, kann ein Unternehmen bei jedem Auftragseingang die Entscheidung treffen, welche Anfrage angenommen oder abgelehnt werden soll.

Der vom Autor Kalyan (2002) eingeführte minimalen Akzeptanzwert wird in der wissenschaftlichen Literatur im Kontext des Konzepts des Revenue Managements als sogenannten Bid-Preis bezeichnet. Bei dem Bid-Preis handelt es sich um einen variierenden Parameter in Abhängigkeit der Zeit (bzw. der Periode) und der verfügbaren Ressourcenkapazität des betrachteten Netzwerks. Er kann als statische Information der optimalen Lösung für die eindimensionalen Probleme angesehen werden. Die Ermittlung des Bid-Preises erfolgt bei den traditionellen Modellformulierungen des RM anhand des *deterministische lineare Programm (DLP)* unter

⁸Vgl. H. deB. Harris und Pinder (1995), S. 307-308.

deterministisch eintreffenden Nachfragen.⁹ Er fungiert als Schwellenpreis für eine jede Ressource im Netzwerk und ist normalerweise beschrieben als geschätzte marginale Kosten aufgrund des nächsten sukzessiven Verbrauchs einer Einheit der Ressourcenkapazität.¹⁰ Laut Gönsch und Steinhardt (2013) erfolgt der Ansatz erstmalig von Talluri (2001) in Verbindung der Optimierung von Passagier Routen. Das Verfahren des Bid-Preises ist ein einfacher Weg, die Kapazitäten der Ressourcen in einem Netzwerk eines Anbieters zu kontrollieren.¹¹ Viele Konzepte der neueren Veröffentlichungen im Bereich des Netzwerk RM sehen die Verwendung des Bid-Preise vor.¹²

Rehkopf und Spengler (2005) zeigen durch das Lösen eines linearen Modells die Kapazitätsallokation für die Problemformulierung des Netzwerk-RM im Fall von MTO-Prozessen. Die Autoren fokussieren sich dabei auf die Branche der Eisen- sowie Stahlindustrie und veröffentlichen zwei Publikationen mit ihren Forschungsergebnissen. Dabei findet der im vorherigen Absatz definierte Bid-Preis Anwendung.

Rehkopf (2007) verfasste nach Veröffentlichung der ersten Publikation eine Schrift zu einem umfangreichen RM-Konzept zur Auftragsannahme bei kundenindividueller Produktion. Auch hier wird als Beispiel der Anwendungsfall in der Branche der Eisen und Stahl erzeugenden Industrie untersucht. Es werden zwei Fallstudien betrachtet, die zum einen die taktisch-operative Allokation der Kapazität und zum anderen die operative Annahmeentscheidung beinhaltet. Bei der Entwicklung einer geeigneten RM-Methodik in der Stahl erzeugenden Industrie kommt zum Tragen, dass in dieser Branche eine strikt divergente Produktionsstruktur vorliegt und dadurch die Auftragsfertigung in einstufige Leistungserstellung zerlegt werden kann.¹³ Dabei konnte die Fallstudie zeigen, dass bei der taktisch-operativen Allokation der Kapazität das vorgestellte Verfahren des RM mit einem FCFS-Verfahren eine ausgesprochene Dominanz gegenüber eines deterministischen Ansatzes mit FCFS-Verfahren aufweist.¹⁴ Für die operative Entscheidungsunterstützung formuliert der Autor ein Netzwerk-RM-Modell, welches durch ein spezielles Verfahren die Restkapazität bestimmt. Das Verfahren berechnet die Restkapazität nach einem real-time Ansatz für den zusicherbaren Bestand¹⁵ aus der kapazitierten Haupt-

⁹Vgl. Talluri und van Ryzin (2004), S. 107-108

¹⁰Vgl. Talluri und van Ryzin (2004b), S. 89

¹¹Vgl. Talluri und van Ryzin (2004b), S. 86-87

¹²Vgl. Petrick et al. (2010), S. 2028; Gönsch und Steinhardt (2013), S. 98-100.

¹³Vgl. Rehkopf (2007), S. 113

¹⁴Vgl. Rehkopf (2007), S. 127-134

¹⁵Vgl. den Ansatz „Available-to-promise“

produktionsprogrammplanung und den bereits eingeplanten Aufträgen des Netzwerks.¹⁶ Ebenso wird hier der Ansatz des Bid-Preises eingesetzt, welcher durch ein duales lineares Optimierungsmodell ermittelt wird.¹⁷ Dabei konnte er in der Fallstudie das Potential der Anwendung eines Bid-Preises und für den angewandte Ansatz zur Ermittlung der Restkapazität feststellen.¹⁸ Die Ergebnisse aus dieser Monografie fließen in die zweite Veröffentlichung von Spengler et al. (2007). In dieser Arbeit wird in einer Fallstudie ebenfalls gezeigt, dass durch eine Bid-Preis-Strategie der Gesamtdeckungsbeitrag verbessert wird.¹⁹ Dabei werden in der Publikation die Bid-Preise jeweils mit einer linearen Modellformulierung sowie einer multi-dimensionalen Knacksack-Modellformulierung berechnet und verglichen. Dabei wird deutlich, dass beide Verfahren den Deckungsbeitrag ähnlich verbessern, wobei sich das Lösen anhand der Knacksack-Modellformulierung als robuster darstellt.²⁰ In der Arbeit zeigten die Autoren, dass durch Anwenden der Heuristik sich der Gesamterlös im Vergleich zu einer einfachen Reihenfolgeannahme²¹ um 5,3% erhöhen lässt.²²

Volling et al. (2012) beschäftigen sich ebenfalls mit dem Auftragsannahmeproblem bei MTO-Prozessen. Dabei legt die Arbeit den Fokus auf die Ermittlung und den Zeitpunkt der Neuberechnung des Bid-Preises für ein Netzwerk RM. Es werden mögliche Szenarien gezeigt, zu welchem Zeitpunkt im Buchungsverlauf der Bid-Preis Neuberechnet werden kann. Dabei erfolgt insbesondere die Untersuchung der Anwendbarkeit bei MTO-Prozessen. Die Notwendigkeit der Neuberechnung resultiert aus der Vereinfachung der Problemstellung des Auftragsannahmeproblems im Netzwerk RM. Im ersten Schritt des in der Arbeit präsentierten Verfahrens erfolgt die Berechnung der anfänglichen Bid-Preise. Im zweiten Schritt werden szenarioabhängige Bid-Preise berechnet, die für den weiteren Verlauf des Buchungshorizonts gelten. Die Autoren nutzen zur Neuberechnung dieser szenarioabhängigen Bid-Preise ein Verfahren der „Neuronalen Netze“. Dabei wird im ersten Schritt anhand der beobachteten und gebuchten Nachfrage die weitere erwartete Nachfrage berechnet. Anschließend wird diese Kennzahl mit der übrigen Ressourcenkapazität ins Verhältnis gesetzt. Sofern die erwartete Restnachfrage größer als die übrige Kapazität des Netzwerks ist, erfolgt die Neuberechnung der Bid-Preise. Dabei wird der Bid-Preis und die erwartete Nachfrage in eine nicht lineare Beziehung gesetzt, bis

¹⁶Vgl. Rehkopf (2007), S. 140; Spengler et al. (2007), S. 160.

¹⁷Vgl. Rehkopf (2007), S. 144.

¹⁸Vgl. Rehkopf (2007), S. 177

¹⁹Vgl. Spengler et al. (2007), S. 157–171.

²⁰Vgl. Spengler et al. (2007), S. 168-169.

²¹„First come, first served.“

²²Vgl. Spengler et al. (2007), S. 170.

beide Betrachtungsobjekte eine gleiche Nachfrageverteilung aufweisen.²³ Laut der in der Arbeit durchgeführten numerischen Untersuchung wird der Deckungsbeitrag durch Anwenden des Verfahrens um 7,8-20,1% gegenüber eines FCFS-Verfahrens verbessert.

Einen umfassenden Branchenüberblick über Anwendungsmöglichkeit des RM liefern Kimms und Klein (2005) in ihrer Veröffentlichung. Dabei wird exemplarische das RM-Instrument der Kapazitätssteuerung untersucht, indem jeweils ein Entscheidungsmodell formuliert wird. In dem Branchenüberblick werden Konzepte zur Modellformulierung für das Luftverkehrswesen (Passagier und Fracht), der Touristikbranche (Hotellerie, Gastronomie und Autovermietung) und eben der Fertigungsindustrie vorgestellt. Das Modell für die Auftragsfertigung sieht explizit keine Lagerhaltung von Kapazitäten vor, wobei die eintreffenden Kundenaufträge über unterschiedliche Ausführungsmodi realisiert werden. Mit den unterschiedlichen Modi wird in dem Modell z. B. eine unterschiedliche Betriebsintensität zur Ausführung einer angenommenen Anfrage verstanden. Mit der Arbeit legen die Autoren dar, welche branchenübergreifenden Voraussetzungen zur Anwendung von RM erforderlich sind. U. a. Einschränkung operativer Flexibilität und Heterogenität des Nachfrageverhaltens. Dabei definieren die Autoren das RM im weitesten Sinne auch als mögliches Instrument für strategisch-taktische Entscheidungsunterstützung.²⁴

Kolisch und Zatta (2006) diskutieren in ihrer Arbeit die Voraussetzungen für die Anwendung von RM in der Vermarktung von Sachleistungen für Geschäftskunden. Zusätzlich werden die Komponenten eines RM-Systems dargestellt und Ergebnisse einer empirischen Studie zum Einsatz von RM in der Prozessindustrie präsentiert. Zu den Komponenten eines RM-Systems zählen lt. den Autoren die Datenanalyse, die Nachfrageprognose, die Optimierung und die Steuerung. Letzteren zwei beziehen sich auf das Lösen des mathematischen Optimierungsmodells zur Ermittlung des Bid-Preises. Durch diese Komponenten eines RM-Systems kann ein verbessertes Kapazitäts- und Preismanagement erfolgen. Die präsentierte empirische Studie belegt, dass Unternehmen den Einsatz von RM-Instrumenten positiv gegenüber stehen und einen zunehmenden Einsatz bestätigen. Dabei geben ca. 80% die befragten Unternehmen an, dass der Einsatz von RM über Systemlösungen erfolgt und dass das Preismanagement gegenüber des reinen Kapazitätsmanagements in den letzten Jahren an Bedeutung gewonnen hat.²⁵

²³Vgl. Volling et al. (2012), S. 1026.

²⁴Vgl. Kimms und Klein (2005), S. 24.

²⁵Vgl. Kolisch und Zatta (2006), S. 40-41.

Anders als die vorher aufgeführten Autoren befassen sich die Autoren Barut und Sridharan (2005) nicht nur mit dem Auftragsannahmeproblem, welches sich im Konzept des RM ergibt, sondern auch um die Planung und der Bestimmung des genauen Zeitpunkts der Fertigung. Die Autoren stellen eine Heuristik vor, die Aufträge in verschiedene Lose sortiert. Dabei werden mehrere Konsumerklassen beachtet. Die Basisidee des Verfahrens ist die Beachtung der relativen Gewinnspannen der Aufträge, damit der Gesamtdeckungsbeitrag erhöht wird.²⁶ Unter Einsatz der Heuristik zeigen die Autoren, dass ein höherer Gewinn aufgrund einer effizienten Nutzung der verfügbaren Kapazität erzielt wird. Dies kommt zustande, da eine Unterscheidung der Aufträge in Bezug von verschiedene Produktklassen mit unterschiedlichen Deckungsbeiträge erfolgt.²⁷

Auch die Autoren Gühlich et al. (2015) beschäftigen sich zusätzlich zur optimalen Kapazitätsallokation mit der Ermittlung der möglichen Fertigstellungszeit eines Auftrags im Kontext des RM-Konzepts. Das von den Autoren vorgestellte Modell berücksichtigt bei jeder Anfrage einen Zeitpunkt der angebotenen Fertigstellung. Dabei betrachtet das Modell Buchungsperioden innerhalb von Planungszeiträumen.²⁸ Dadurch ist es dem Modell gestattet, zum Anfragezeitpunkt den angebotenen Fertigstellungszeitpunkt ebenfalls für die Auftragsproblematik zu berücksichtigen. Dies erfolgt in der Form, dass sich zur Herstellung des gewünschten Produkts ein Zeitraum der Produktion ergibt. Damit kann eine Entscheidung getroffen werden, in Abhängigkeit der Kapazitäten und der Fertigstellungszeit, zu welchem Zeitpunkt die Herstellung starten soll. Dabei ist mit der Annahme der Anfrage der angebotene Fertigstellungszeitpunkt für den Kunden bestätigt. Das Modell berücksichtigt auch Lager- und Fertigstellungsrückstandkosten in dem möglichen Produktionszeitraum und bedarf einer Ermittelt einer möglichen Durchführbarkeit. Für die Durchführungsprüfung wird ein lineare Modell gelöst. Dies ist notwendig, damit der Zeitpunkt der angebotenen Fertigstellungszeitpunkt und Planungsgrundlage geprüft werden kann.²⁹ Auch hier werden die Opportunitätskosten mittels der Ermittlung eines Bid-Preises approximiert.³⁰ In einer numerischen Untersuchung ist als Feststellung dargelegt, dass das Modell der Autoren im Vergleich zu anderen Algorithmen eine niedrigere Abweichung zur optimalen Lösung aufweist.³¹

Kuhn und Defregger (2004) beschreiben die Anwendung des RM anhand eines

²⁶Vgl. Barut und Sridharan (2005), S. 291

²⁷Vgl. Barut und Sridharan (2005), S. 310.

²⁸Vgl. Gühlich et al. (2015), S. 10.

²⁹Vgl. Gühlich et al. (2015), S. 12-14.

³⁰Vgl. Gühlich et al. (2015), S. 18-19.

³¹Vgl. Gühlich et al. (2015), S. 22-24.

Papierherstellers, der eine Auftragsfertigung mit zwei unterschiedlichen Klassen anbietet. Eine Klasse ist dabei eine mit höheren und die andere mit niedrigen Erlösen. Dabei nehmen die Autoren an, dass durch Annahme der Aufträge die Maschinen eine Belegungszeit zur Fertigung des geforderten Produkts haben. Außerdem unterscheiden sich die Aufträge durch ihre individuelle Lieferzeit. Durch diese Einschränkungen kommt es zu einer differenzierten Möglichkeit der Auftragsannahme, da untersucht werden muss, ob Aufträge aufgrund blockierter Maschinen innerhalb der geforderten Lieferzeit möglich sind und ob genügend Kapazität verfügbar ist. In der Fallstudie wird das Konzept des FCFS-Verfahrens mit der linearen Programmierung für die optimale Auftragsannahmepolitik verglichen. Auch in dieser Fallstudie konnten die Autoren die Anwendung des RM für MTO-Prozesse durch eine Gesamterlösverbesserung bestätigen.

Specht und Gruß (2008) schreibt in ihrem Artikel über die Anwendung des Revenue Management im Bereich der Automobilindustrie. Dabei wird als Betrachtungselement das Unternehmen „Ford Motor Company“ herangezogen, wobei die Autoren zum Zeitpunkt der Erstellung der Publikation keinen Zugriff auf internen Dokumente des Unternehmens hatten. Die Autoren beschreiben die Anwendung von mehreren Teilsystemen die Kundenwünsche systematisch abfragen. Dem Unternehmen ist es daraufhin möglich sehr nah an den Bedürfnissen des Marktes zu produzieren.³² Es konnten aber keine Anhaltspunkte über eine mögliche Verbesserung der Kapazitätsallokation der Werke ermittelt werden. Specht und Gruß (2008) kommen daher zum Ergebnis, dass es sich eher um preisbasierte und nicht um kapazitätsbasierte RM-Systeme handeln muss. Sie konnten aber grundsätzlich die Anwendung des RM in der Automobilindustrie bestätigen.

Die Problematik der wachsenden Anzahl an möglichen Anwendungsfeldern, der unterschiedlichen Modelle und der verfügbaren Software in Bezug zum RM nehmen sich die Autoren Quante et al. (2009) in einem zusammenfassenden Überblick an. Zur den Anwendungsfelder zählen aus der Sicht der Autoren die anwendungsorientierte Lieferkettenmerkmale und die Anwendungstypen. Bei den Lieferkettenmerkmalen werden die einzelnen Systeme der Lieferkette in Bezug der Merkmale des RM untersucht. Dabei wird eine beispielhafte Übersicht aufgezeigt, welche Merkmale des RM sich in Abhängigkeit des Anwendungsfelds ergeben. Abhängig des Anwendungsfalls kann mit diesem Verfahren eine Charakterisierung der relevanten Merkmale für das RM erfolgen. Bei der in dieser Arbeit betrachteten Auftragsfertigung kann der in der Veröffentlichung von Quante et al. (2009) beschriebenen Fertigung von Ma-

³²Vgl. Specht und Gruß (2008), S. 66.

schinen gleichgesetzt werden. Bei der Maschinenfertigung ist der KAKP vor der Produktion, daher handelt es sich um ein MTO-Prozess. Die Flexibilität der Kapazität ist hoch. Diese Merkmalsausprägung resultiert aufgrund der Betrachtung der Verwendung der Kapazität innerhalb des SCM-Systems der Produktion. Bei den Maschinen handelt es sich um langlebige Produkte mit einem langen Lebenszyklus. Bei dem Anwendungsfall der Auftragsfertigung ist Preisflexibilität ab dem Zeitpunkt des Auftragseingangs gegeben. Die Autoren bestimmen des Weiteren die Profitheterogenität. Bei einem MTO-Prozess der Auftragsfertigung ist der Profit lt. den Autoren abhängig vom Zeitpunkt und der eintreffenden Kundenaufträge. Zusammengesetzt wird er durch den erzielten Ertrag, der ebenfalls abhängig des Zeitpunkts und des Auftrags ist. Der Ertrag wird um die Kosten reduziert und um einen Wert der strategische Bedeutung erhöht. Diese beiden Werte sind wiederum nur abhängig des Kundenauftrags.³³

Ebenfalls wird in der Arbeit eine mögliche Klassifizierung der Modelle des RM anhand der Lieferkette getätigt. Dabei werden zu den einzelnen Systemen der Lieferkette mögliche Ausprägungen der Parameter der Modelle angegeben. Dabei werden verschiedene Modelle in ihren Ausprägungen der eingesetzten Parametern verglichen, wie z. B. bei den Modellen der stochastischen Lagerbestandskontrolle, des „vielversprechendsten Auftrags“ und des traditionellen RM.³⁴ Als letzten Teil der Übersicht werden verschiedene softwarebasierte Lösungen für das RM aufgeführt. Anschließend erfolgt jeweils für die Branchen der Serviceindustrie, des Handels und der Fertigung ein zusammenfassendes Ergebnis der Anwendungsfelder, der Modelle und der Softwarelösungen. Für MTO-Prozesse kommen die Autoren zum Schluss, dass die verfügbaren Modelle und Softwarelösungen nicht das Problem darstellen, sondern die Lösungsqualität bzw. Zuverlässigkeit und die kurze Reaktionszeiten der Auftragsannahme.³⁵

Aufgrund erhöhter Unternehmensforschung im Bereich des RM verfassen die Autoren Cheraghi et al. (2010) ein Literaturüberblick über RM-Systeme in der Fertigungsindustrie anhand der Dimensionen des Kapazitäts- und Preismanagements sowie der Marktsegmentierung. Neben der Kapazitätssteuerung befassen sich die Autoren auch um Verfahren der Segmentierung und der Bepreisung, da die richtige Ansprache der unterschiedlichen Kundengruppen eines Fertigungsunternehmens hohe Relevanz aufweist.³⁶ Durch die Marktsegmentierung ist es einem Unterneh-

³³Vgl. Quante et al. (2009), S. 37.

³⁴Vgl. Quante et al. (2009), S. 43-44.

³⁵Vgl. Quante et al. (2009), S. 56-57.

³⁶Vgl. Cheraghi et al. (2010), S. 64-65.

men möglich, die Produkte unter unterschiedlichen Rahmenbedingungen anzubieten. Infolgedessen kann ein verbessertes Preismanagement erfolgen. Dabei stellen die Autoren die Besonderheit des Preismanagements in der Fertigung heraus. Anders als bei Fluggesellschaften, verfällt/verdirbt die Kapazität nicht aufgrund eines ablaufenden Planungsperiode. Die Kapazität könnte weiterhin abgesetzt werden. Eine Verderblichkeit der Kapazität erfolgt jedoch in der Hinsicht, da ein möglicher Abschluss eines Verkaufsgeschäfts mit einem potentiellen Kunden nicht eingetroffen ist.³⁷ In Bezug des Literaturüberblick zum Kapazitätsmanagement fassen die Autoren ihre Ergebnisse in den Abschnitten „Auftragswahl“ und „Kapazitätsallokation“ zusammen. Bei dem Überblick zu den Modellen kann insbesondere die Arbeit von Defregger und Kuhn (2007) genannt werden, da sich die Arbeit mit der Annahme von Aufträgen eines Unternehmens mit endlichem Lagerbestand beschäftigt. In dieser Arbeit wird ein Modell für MTO-Prozesse modelliert, bei dem das Auftragsannahmeproblem mittels zeitdiskreten Markov-Entscheidungsprozesses und eines heuristischen Verfahrens untersucht wird. Das in der Arbeit vorgestellte Modell basiert dabei auf Sequenzen von Entscheidungen, in welchen Fällen eine Anfrage akzeptiert und zur Reduktion des Lagerbestands führt. Dabei werden die Auftragsarten in Klassen sortiert und mittels einer Heuristik untersucht. Sofern eine Anfrage zu der Klasse mit niedrigen Erträgen gehört, wird diese abgelehnt. Dabei konnte sich die Heuristik in einer numerischen Untersuchung gegenüber eines FCFS-Verfahrens behaupten.³⁸

Sucky (2009) zeigt das Potential vom RM in der Auftragsfertigung anhand eines einfachen Beispiels eines Modells zur Kapazitätssteuerung. Dabei erfolgt die Kapazitätssteuerung auf einem Konzept der Beachtung eines Buchungslimits. Damit versucht der Autor konkrete Anwendungsfälle des RM in der Auftragsfertigung zu ermitteln. Dabei führt der Autor auch die möglichen Hemmnisse des RM in der Auftragsfertigung auf. Zum Beispiel wird der Fall aufgeführt, dass Anfragen von wichtigen Geschäftskunden aufgrund niedrigerer Stückdeckungsbeiträge abgelehnt werden, da diese tendenziell weniger für die gleiche Leistung zahlen.

³⁷Vgl. Cheraghi et al. (2010), S. 65.

³⁸Vgl. Defregger und Kuhn (2007), S. 150-154.

5 Ein exaktes Lösungsverfahren zur Auftragsannahme- und Lagerhaltungsentscheidung bei auftragsbezogenen Instandhaltungsprozessen

5.1 Grundlegendes zum Lösen und Implementieren des Auftragsannahmeproblems

In Kapitel 3 ist das Grundmodell des Netzwerk RM für die Annahme von Kundenanfragen als *Bellman'sche Funktionsgleichung* dargestellt. Gleichung (1) zeigt die mathematische Modellformulierung und Gleichung (4) zeigt eine Vereinfachung der Modellformulierung. In diesem Kapitel wird dieses Modellformulierung in ein Computersystem implementiert und mittels dieser Implementierung erfolgt das exakte Lösen von vordefinierten Szenarien. Dabei wird das Grundmodell mit der Entscheidung über eine mögliche Lagerhaltung erweitert. Bei der Implementierung wird in dieser Arbeit das Grundmodell mit neuen Parametern einer möglichen Lagerhaltungsentscheidung bei der Annahme von Kundenaufträgen ergänzt. Zum einen soll eine Verdeutlichung der Funktionsweise der Modellerweiterung der *Bellman'sche Funktionsgleichung* erfolgen und zum anderen wird das Ergebnis dieser Modellerweiterung für eine numerische Untersuchung verwendet.

Bei der *Bellman'sche Funktionsgleichung* (1) aus Abschnitt 3.3 handelt es um ein stochastisch-dynamisches Optimierungsproblem in rekursiver Form.⁸ Das Optimierungsmodell besteht aus verschiedenen gleichartigen Teilproblemen und zur

⁸Vgl. ???, S. ???

Lösung des gesamten Optimierungsmodells müssen alle Teilprobleme gelöst werden.⁹ Zur Lösung solcher rekursiven Modellformulierung wird auf das vom amerikanischen Mathematiker Richard Bellman entwickelte Konzept der **Dynamischen Programmierung (DP)** zurückgegriffen.¹⁰ Bei diesem Konzept werden die berechneten Teilergebnisse gespeichert und bei der weiteren Berechnung des Optimierungsmodells verwendet.¹¹ Durch eine solche Implementierung kann die rekursive Berechnung des Modells verbessert werden, da auf bereits berechnete Teilergebnisse zurückgegriffen wird, anstelle diese neu zu berechnen.

Eine Form der Implementierung des Konzepts der DP ist das Anwenden einer *Memofunktion*. Mit der Memofunktion werden die Teilprobleme des Konstruktors des Optimierungsmodells überführt, ob diese bereits schon berechnet sind.¹² Bei dem Konstrukt handelt es sich um ein Graphen mit allen Teilproblemen und den einzelnen Übergängen in die nachfolgenden Teilprobleme. Zur Verdeutlichung einer Memofunktion im Netzwerk RM in der Auftragsannahme wird ein einfaches Beispiel eingeführt: $j = \{1, 2\}$, $h = \{1\}$, Startperiode $t = 2$, $c_1 = 2$, $a_{11} = 1$, $a_{12} = 1$.

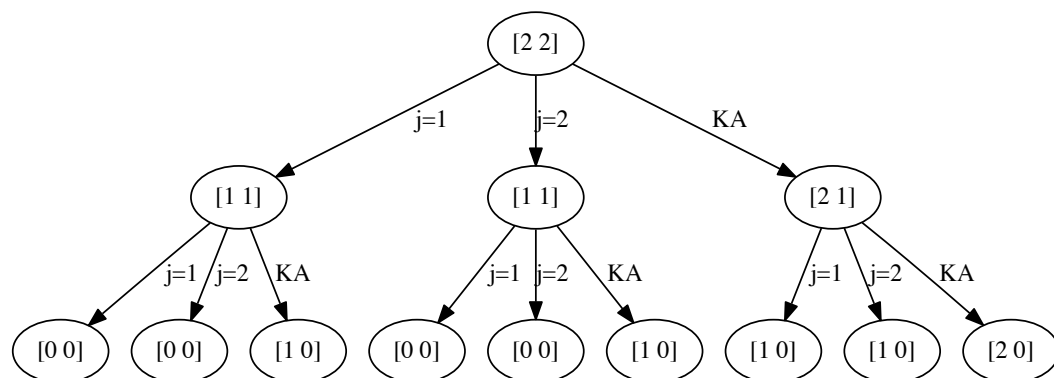


Abbildung 8 Rekursive Übergänge der Systemzustände ohne Memofunktion

Quelle: ???? Legende: Annahme einer Produktauftrag entspricht 'j', KA='Kein Auftrag'

Abbildung 8 zeigt die möglichen Systemzustände in Form der Knoten und Übergänge in Form der Kanten. Dabei ist ein Knoten definiert als Zahlenfolge $[c_h t]$ und zeigt damit den Kapazitätsbestand c_h zum Zeitpunkt t . Durch Annahme einer Produkthanfrage $j \in \mathcal{J}$ oder sofern keine Anfrage eintrifft, wird ein Systemzustand verlassen. Diese rekursive Folge des Graphen wird konstruiert durch Anwenden der Gleichung (1). Sofern keine Memofunktion Anwendung findet, werden die einzelnen

⁹Vgl. ???, S. ???

¹⁰Vgl. ???, S. ???

¹¹Vgl. ???, S. ???

¹²Vgl. ???, S. ?

Teilprobleme jeweils mehrfach ermittelt, da sie jeweils für das vorherige Teilproblem notwendig sind.

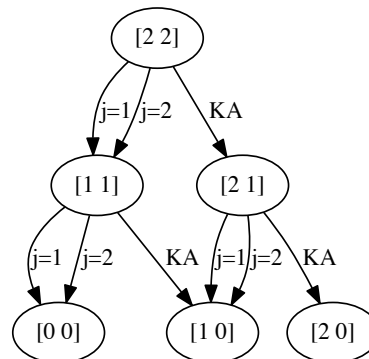


Abbildung 9 Rekursive Übergänge mit Memofunktion

Quelle: ???? **Legende:** Annahme einer Produktauftrag entspricht 'j', KA='Kein Auftrag'

Durch Anwenden einer Memofunktion können die möglichen Übergänge der Systemzustände des Beispiels vereinfacht werden, wie Abbildung 9 zeigt. Jedes Systemzustand ist nur einmal im Netzwerk vorhanden und sofern der rekursive Verlauf auf ein bereits ermitteltes Teilproblem trifft, erfolgt das Abrufen der bereits gespeicherten Lösung. Anders formuliert bedeutet dies, dass nachdem der Systemzustand bzw. das Teilproblem $[1 \ 1]$ aufgrund der Anfrage nach Produkt $j = 1$ gelöst ist, die Berechnung des gleichen Systemzustands $[1 \ 1]$ aufgrund der Anfrage nach Produkt $j = 2$ nicht mehr notwendig ist. Das Ergebnis des Teilproblems wird direkt aus der Memofunktion abgerufen.

Der in dieser Arbeit verwendete **Algorithmus** zum exakten Lösen des Auftragsannahmeproblems im Netzwerk RM verwendet eine solche Memofunktion. Der Algorithmus berechnet den Erwartungswert des maximal möglichen Ertrags für ein Netzwerk RM auf Basis der rekursiven Form der Gleichung (4). Er durchläuft alle möglichen Teilprobleme bzw. Systemzustände des Netzwerks, indem der Algorithmus sich selbst mit angepassten Parametern aufruft. Als Grenzen für die rekursive Abfolge werden die Grenzbedingungen (2) und (3) hinterlegt. Sofern eine dieser Grenzen erreicht ist bzw. die Lösungen für die notwendigen nachfolgenden Teilprobleme vorhanden sind, kann das Teilproblem gelöst werden. Somit erfolgt das Lösen des Optimierungsproblems durch Rückwärtsinduktion der rekursiven Folge, wobei bei jedem Teilproblem geprüft wird, ob bereits eine Lösung in der Memofunktion vorliegt. Nachfolgend wird der verwendete Algorithmus als Pseudocode dargestellt.

Pseudocode: Auftragsannahmeproblem im Netzwerk RM;

Ermittlung des Erwartungswerts $V(\mathbf{c}, t)$

Eingabe: *Memofunktion*

Eingabe: \mathcal{J} , \mathbf{c} , t sowie \mathbf{a}_j , r_j und $p_j(t) \forall j \in \mathcal{J}$

wenn $V(\mathbf{c}, t) \notin \text{Memofunktion}$ **dann**

wenn $t \neq 0$ **dann**

$value^{reject} = V(\mathbf{c}, t - 1)$

für jedes $j \in \mathcal{J}$ **tue**

wenn $\mathbf{c} - \mathbf{a}_j \geq \mathbf{0}$ **dann**

$value_j^{accept} = p_j(t) \cdot \max[r_j - value^{reject} + V(\mathbf{c} - \mathbf{a}_j, t - 1), 0]$

sonst

 Grenzbedingung (3): $V(\mathbf{c} - \mathbf{a}_j, t - 1) = -\infty$

$value_j^{accept} = p_j(t) \cdot \max[r_j - value^{reject} - \infty, 0]$

Ende

$V(\mathbf{c}, t) = value^{reject} + \sum_{j \in \mathcal{J}} value_j^{accept}$

sonst

 Grenzbedingung (2): $V(\mathbf{c}, t) = 0$

$Memofunktion = Memofunktion + V(\mathbf{c}, t)$

sonst $V(\mathbf{c}, t) \in \text{Memofunktion}$;

Ausgabe: $V(\mathbf{c}, t)$

Zur Implementierung des Algorithmus zum Lösen des Grundmodells des Auftragsannahmeproblems im Netzwerk RM wird die Programmiersprache Python/2.7.1 verwendet. Es handelt sich um eine höhere Programmiersprache mit einem Interpreter.¹³ Als Kommandozeileninterpreter wird IPython/3.2.1 genutzt. Zur Verbesserung der Laufzeit der Implementierung wird auf die Programmbibliothek NumPy/1.9.2 zurückgegriffen. Mit dieser Programmbibliothek ist es möglich multidimensionale Datenstrukturen zu formen und mit den integrierten numerischen Algorithmen sowie mathematischen Werkzeugen zu bearbeiten.¹⁴ Dabei ist NumPy/1.9.2 ein Bestandteil von SciPy/0.15.1.¹⁵ Die Laufzeitverbesserung kommt zu Stande, da die Funktionen der Programmbibliothek hauptsächlich in der Programmiersprache C geschrieben sind.¹⁶ Zusätzlich wird die Programmbibliothek NetworkX/1.9.1 verwendet, um die ermittelten Erwartungswerte in eine Netzwerkdatenstruktur zu überführen. Diese Netzwerkdatenstruktur ist explizit für Multigraphen geeignet, was für diese Problemstellung notwendig ist. Die Teilprobleme werden miteinander in Beziehung gesetzt, indem ein Teilproblem ein Knoten bildet

¹³Vgl. ???, S. ??

¹⁴Vgl. ???, S. ???

¹⁵Vgl. ???, S. ??

¹⁶Vgl. ???, S. ??

und die für das Teilproblem notwendigen nachfolgenden Teilprobleme die Kanten der Netzwerkdatenstruktur bilden. Dadurch wird das Optimierungsproblem als Netzwerk interpretierbar. Weiter kann dieser Netzwerkdatenstruktur grafisch exportiert werden, indem eine für das Programm GraphViz/2.38.0 verwertbare Datei erzeugt wird. Dadurch ist das Rendern der jeweiligen Netzwerkdatenstruktur möglich. Sämtliche in dieser Arbeit abgebildeten Graphen sind mit dem Programm GraphViz/2.38.0 erstellt. Zusätzlich erfolgt die Verwaltung der Daten durch die Datenanalyse-Programmbibliothek Pandas/0.16.2. Nachfolgend wird das Grundmodell auf Basis der *Bellman'schen Funktionsgleichung* mit der Lagerhaltungsentscheidung erweitert und der hier aufgeführte Algorithmus mit den aufgeführten Erweiterungen umgeformt. Im Anschluss dieses Kapitels erfolgt die abschließende numerische Untersuchung von Szenarien....

5.2 Mathematische Formulierung eines Modells zur Auftragsannahme- und Lagerhaltungsentscheidung

Die Gleichung (1) wird bei Modellerweiterung der Auftragsannahme- und Lagerhaltungsentscheidung um den Parameter des Lagerbestands y_h erweitert. In Bezug des Ausgangsproblems der Lagerhaltungsentscheidung ist dieser Schritt der Modellerweiterung nötig, damit ein Parameter für bereits reparierte Produkte im Modell vorhanden ist. Es existiert für jede Ressource $h \in \mathcal{H}$ ein möglicher Lagerbestand y_h . Die Obergrenze eines jeden Lagerbestands einer Ressource $h \in \mathcal{H}$ wird als Parameter y_h^{max} beschrieben. Die einzelnen Lagerbestände können als Vektor y zusammengefasst werden.

Die Modellerweiterung soll es ermöglichen ertragsarme Anfragen j^- nach auftragsbezogenen Instandhaltungsprozessen abzulehnen und eine Erhöhung des Lagerbestands y_h über des Parameters a_{hj} zu ermöglichen, sofern diese Anfragen im weiteren Verlauf des Buchungshorizonts T relevante Ressourcenkapazitäten c_h für ertragreichere Anfragen j^+ reduzieren. Weiter ist eine Annahme von Anfragen j nach Instandhaltungsprozessen über den Lagerparameter y_h möglich, indem anstelle der Kapazitätsreduktion eine Lagerreduktion erfolgt. Der Parameter y_h wird bei der Annahme via Lagerbestand um den Parameter a_{hj} reduziert. Damit fungiert in diesem Modell der Parameters a_{hj} als Bestandsveränderung der Kapazitäten oder

der Lagerbestände.

In der Modellerweiterung werden Anfragen nach Produkten betrachtet, die für bestimmte Leistungserstellungszeitpunkte vorgesehen sind. Ein Leistungserstellungszeitpunkt wird durch den Parameter \hat{t} definiert. Damit existiert für eine jede Ressourcen h eine Kapazität c_h und ein Lagerbestand y_h für einen jeden Leistungserstellungszeitpunkt \hat{t} . Den Parametern c_h und y_h kann der hochgestellte Index zum zugehörigen Buchungsabschnitt \hat{t} ergänzt werden. Die Menge aller Leistungserstellungszeitpunkte wird mit dem Parameter \hat{T} abgebildet. Zur Vereinfachung des Modells wird jedoch nur eine einzelne Ressource h über den gesamten Buchungshorizont betrachtet. Damit gilt $c_{h=\hat{t}}$. Der jeweilige Eintrag im Vektor \mathbf{c} entspricht somit der Kapazität c_h der Ressource h zum Leistungserstellungszeitpunkt \hat{t} .

Der Kapazitätsverbrauch a_j einer Produkthanfrage j beschreibt mit dem jeweiligen Eintrag für die Ressource h damit den Zeitpunkt der Erfüllung der Anfrage. Sei \tilde{j} eine Anfrage nach einem Produkt für diese Modellerweiterung, dann beschreibt der Parameter $\mathbf{a}_{\tilde{j}} = (0, 1, 0)$, dass die Erfüllung der Anfrage für den Leistungserstellungszeitpunkt $\hat{t} = 2$ vorgesehen ist. Die bis zur Leistungserstellung benötigten Perioden können ermittelt werden durch Division des Buchungshorizonts T und der Anzahl an Leistungszeitpunkten \hat{t} . Wobei T durch die Anzahl der Leistungszeitpunkten \hat{t} ganzzahlig teilbar sein muss. Daher wird bei dieser Modellerweiterung strikt ein Buchungshorizont gewählt, der in Fünfer-Schritten den nächsten Zeitpunkt der Leistungserstellung vorsieht. Sofern für eine Anfrage j mit einem Kapazitätsverbrauch $\mathbf{a}_{\tilde{j}} = (0, 1, 0)$ ein Buchungshorizont von $T = 15$ vorgesehen ist, dann erfolgt die Leistungserstellung der Anfrage j mit Ablauf der sechsten Periode t .

Das Modell erlaubt die vorzeitige Inanspruchnahme von Kapazitäten c_h für Anfragen j für nachfolgende Leistungserstellungszeitpunkte \hat{t} , was die Annahme der Instandhaltungsauftrags oder das Instandsetzen von Ressourcen entspricht. Dies resultiert aus der Tatsache, dass die Bestandsveränderung \mathbf{a}_j einer Anfrage j bereits vor des eigentlichen Leistungserstellungszeitpunkts \hat{t} den Zugriff auf die Kapazität \mathbf{c}_j oder den Lagerbestand \mathbf{y}_j erlaubt. Jedoch darf kein Zugriff des Parameters \mathbf{a}_j nach Überschreitung des Leistungserstellungszeitpunkts \hat{t} auf die jeweiligen Bestände \mathbf{c} bzw. \mathbf{y} erfolgen. Anders formuliert bedeutet dies, dass nicht beanspruchte Kapazitäten c_h für vergangene Leistungserstellungszeitpunkte \hat{t} nicht mehr für weitere Auftragsannahme verwendet werden dürfen. Diese Eigenschaft wird mit der Wahrscheinlichkeit $p_j(t)$ für die jeweiligen Anfragen j gesteuert. Dabei ist der Zeitpunkt an dem die erste Anfrage nach einem Produkt j unerheblich. Abbildung

10 zeigt die Rahmenbedingungen der Modellerweiterung als grafische Darstellung. Dabei soll der Zusammenhang von Buchungsperioden und Leistungserstellungszeitpunkte verdeutlicht werden, aber auch aufgezeigt werden, welche Produkthanfragen in welchen Zeiträumen eintreffen können.

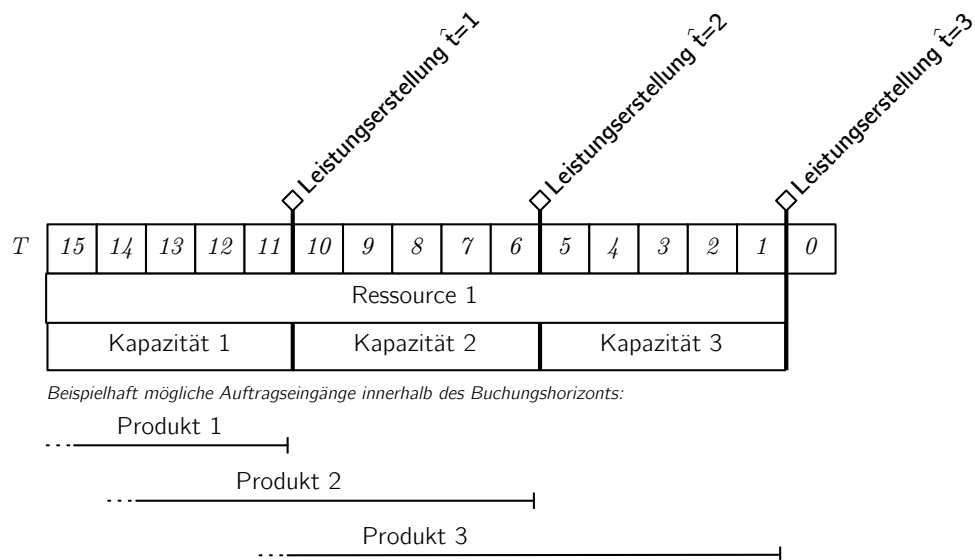


Abbildung 10 Zusammenhang von Buchungsperioden und den Buchungsabschnitten
In Anlehnung an: ????

Bei der Modellformulierung erfolgt die Anpassung der *Bellman'schen Funktionsgleichung* (1) um die Lagerentnahme in Form des Terms $r_j + V(\mathbf{c}, \mathbf{y} - \mathbf{a}_j, t - 1)$. Der Term beschreibt die Annahme mittels des Lagerbestands. Damit ist es dem Unternehmen möglich entweder die Kapazität oder den Lagerbestand zur Annahme einer Anfrage eines Instandhaltungsprozess in Anspruch zu nehmen. Die gewollte Ablehnung einer Anfrage j zur Aufarbeitung einer Ressource h für den Lagerbestand y_h wird mit dem Term $V(\mathbf{c} - \mathbf{a}_j, \mathbf{y} + Y(\mathbf{a}_j), t - 1)$ dargestellt. Damit ist dem Modell die Entscheidung über die Ablehnung einer Anfrage j gestattet, wobei bei dieser Ablehnung der Anfrage eine Reduktion der Ressourcenkapazität \mathbf{c} und die Erhöhung des Lagerbestands \mathbf{y} um jeweils des Parameters \mathbf{a}_j einhergeht. Die Erhöhung des Lagerbestands \mathbf{y} einer jeden Ressource h erfolgt über den gesamten Buchungshorizont. Sei der aktuelle Lagerbestands $\mathbf{y} = (0, 0, 0)$ über alle Leistungserstellungszeitpunkte \hat{t} , wobei aufgrund der Vereinfachung des Modells $h = \hat{t}$ entspricht. Dann erfolgt durch die Annahme einer Anfrage j mit der Bestandsveränderung \mathbf{a}_j eine Erhöhung des Lagerbestands auf $\mathbf{y} = (0, 1, 1)$, da die Leistung erst nach Erstellung in $\hat{t} = 1$ verfügbar ist.

Zur Ermittlung des Lagerbestands aufgrund der Annahme der Aufarbeitung von

Ressourcen ist die Hilfsfunktion $Y(\mathbf{a}_j)$ erforderlich, die den Parameter für den Kapazitätsverbrauch \mathbf{a}_j über alle nachfolgenden Leistungserstellungszeitpunkten kumuliert.¹⁷ Als Beispiel wird der Kapazitätsverbrauch $\mathbf{a}_j = (0, 1, 1, 0)$ betrachtet. Ein Auftrag j mit einem Verbrauch des vorher aufgeführten Parameters \mathbf{a}_j zeigt eine Leistungserstellung über mehrere Perioden auf. Die Hilfsfunktion zur Ermittlung der Lagerbestandsveränderung berechnet damit eine Lageraufstockung in Höhe des Vektors $Y(\mathbf{a}_j) = (0, 0, 1, 2)$. Es muss beachtet werden, dass ein Auftrag j mit einem Kapazitätsverbrauch von $\mathbf{a}_j = (0, 0, 1, 2)$ eine Lagererhöhung $Y(\mathbf{a}_j) = (0, 0, 0, 1)$ verursacht. Diese Erhöhung liegt außerhalb des Betrachtungszeitraum. Damit kann zwar ein Teil der Bestandsveränderung für den aktuellen Betrachtungszeitraum genutzt werden, aber der andere Teil geht in dieser Modellformulierung verloren. Eine zusätzliche Betrachtung einer Beanspruchung von Beständen außerhalb des Betrachtungszeitraums (Perioden $t \leq 0$) in Form von z. B. Lagerentnahmen findet in dieser Arbeit keine Anwendung.

Die mathematische Formulierung der *Bellman'schen Funktionsgleichung* für Auftragsannahme- und Lagerhaltungsentscheidungen bei auftragsbezogenen Instandhaltungsprozessen wird damit wie folgt beschrieben:

$$\begin{aligned} V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t) = & \sum_{j \in \mathcal{J}} p_j(t) \max[V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1), r_j + V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j, \mathbf{y}^{\hat{t}}, t-1), \\ & r_j + V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}} - \mathbf{a}_j, t-1), V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j, \mathbf{y}^{\hat{t}} + Y(\mathbf{a}_j), t-1)] \\ & + p_0(t) \max_{\substack{j \in \mathcal{J}, \\ p_j(t) > 0}} [V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1), V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j, \mathbf{y}^{\hat{t}} + Y(\mathbf{a}_j), t-1)] \end{aligned} \quad (8)$$

Die Gleichung (8) zeigt im ersten Term, welche Entscheidungen bzgl. der optimalen Politik möglich sind, sofern eine Anfrage $j \in \mathcal{J}$ in Abhängigkeit der Wahrscheinlichkeitsverteilung $p_j(t)$ eintrifft. Es wird das Maximum über die Entscheidungen „Ablehnung des Auftrags“, „Annahme via Ressourcenkapazität“, „Annahme via Lagerbestand“ sowie „Ablehnung und Aufarbeitung von Ressourcen“. Damit geht aus der Gleichung (8) hervor, dass es zu jedem Erwartungswert der rekursiven Folge, somit für jedes Teilproblem des Netzwerks, für eine jede Anfrage j die vier Optionen bzw. Entscheidungen existieren. Es gibt damit für eine jede Produktanfrage j im gesamten Netzwerk einen Wert, der die optimale Politik beim Eintreffen einer Anfrage beschreibt. Diese Option der optimalen Politik wird als Parameter $d_j(\mathbf{c}, \mathbf{y}, t) \forall j \in \mathcal{J}$ definiert. Dieser Parameter existiert für jedes Produkt und ist abhängig des Systemzustands. Ein Systemzustands wird weiterhin durch die ver-

¹⁷Vgl. Lars

bleibende Kapazität c und durch den Lagerbestand \mathbf{y} zur Periode t beschrieben. Der Parameter kann ebenfalls als Vektor $\mathbf{d}(\mathbf{c}, \mathbf{y}, t)$ interpretiert werden und zeigt damit für ein Teilproblem des Netzwerks die optimale Politik über alle möglichen Anfragen. Inhalt des Vektors ist der jeweilige Index der optimalen Option bzw. Politik. Es nimmt den Wert $d_j(\mathbf{c}, \mathbf{y}, t) = 1$ an, sofern die Option der Auftragsannahme via Kapazität den höchsten Erwartungswert liefert und den Wert $d_j(\mathbf{c}, \mathbf{y}, t) = 2$, wenn die Auftragsannahme via Lagerbestand der optimalen Politik entspricht. Sofern die Ablehnung des Auftrags erfolgen soll, aber inkl. der Entscheidung der Aufarbeitung der Ressourcen, dann nimmt der Parameter $d_j(\mathbf{c}, \mathbf{y}, t)$ den Wert 3 an. Sofern keine Anfrage akzeptiert werden soll, nimmt der Parameter den Wert $d_j(\mathbf{c}, \mathbf{y}, t) = 0$ an.

Für die verschiedenen Optionen existieren unterschiedliche OC_j , wie aus der Gleichung (8) zu erkennen ist. Damit trägt der Parameter OC_j den Superskript d für die jeweilig möglichen Optionen aufgrund der Modellformulierung. In dieser Modellerweiterung existieren vier unterschiedliche Optionen, die den möglichen Ausprägungen der optimalen Politiken des Parameters $d_j(\mathbf{c}, \mathbf{y}, t)$ entsprechen. Sofern die Entscheidung über die verschiedenen möglichen Optionen im betrachteten Systemzustand gleichwertig sind, welches abhängig von den Parametern r_j und OC_j^d ist, wird die Annahme einer Anfrage über die Ressourcenkapazität bevorzugt. Anschließend erfolgt die Bevorzugung der Annahme einer Anfrage via Lagerbestand, falls die anderweitigen Optionen einen gleichwertigen Wert aufweisen.

Es gelten die Grenzbedingungen (2) sowie (3) aus Kapitel 3 und bzgl. der optimalen Politik zur Annahme von Anfragen nach Produkt j gilt folgende Annahme:

$$d_j(\mathbf{c}, \mathbf{y}, t) := \begin{cases} 1, & \text{für } r_j - OC_j^1 \geq r_j - OC_j^2 \\ 2, & \text{für } r_j - OC_j^2 \geq OC_j^3 \\ 3, & OC_j^3 > 0 \\ 0, & \text{sonst} \end{cases} . \quad (9)$$

Die *Bellman'schen Funktionsgleichung* (8) zeigt ebenfalls den Term an, sofern keine Anfragen eintreffen. Bei der Auftragsannahme von Kundenaufträgen unter Berücksichtigung von Lagerhaltungsentscheidung muss beachtet werden, dass auch Entscheidungsalternativen beim Eintreffen keiner Anfragen nach Produkten $j \in \mathcal{J}$ bestehen. Sofern keine Anfragen nach Produkten $j \in \mathcal{J}$ eintreffen, gibt es die Optionen bzgl. einer Aufarbeitung von Ressourcen $h \in \mathcal{H}$ über alle möglichen Produkthanfragen $j \in \mathcal{J}$. Damit existiert auch eine optimale Politik, sofern keine

Anfragen nach den Produkten j eintreffen. Für diese optimale Politik wird der Parameter $d_0(\mathbf{c}, \mathbf{y}, t)$ verwendet. Es hat die gleiche Abhängigkeit wie der Parameter für die optimale Politik bei Auftragseingang ($d_j(\mathbf{c}, \mathbf{y}, t)$). Er zeigt jedoch nicht einen Index über die optimale Entscheidung, sondern er zeigt den Index der Produkthanfrage j , sofern eine Lagerproduktion durch Aufarbeitung von Ressourcen $h \in \mathcal{H}$ erfolgen soll, und sonst nimmt er den Wert 0 an, wenn keine Lagerproduktion bzw. die Aufarbeitung erfolgen soll. Damit kann der Parameter wie folgt definiert werden:

$$d_0(\mathbf{c}, \mathbf{y}, t) := \begin{cases} j, & \text{für } \max_{\substack{j \in \mathcal{J}, \\ p_j(t) > 0}} OC_j^3 \\ 0, & \text{sonst} \end{cases}. \quad (10)$$

Dabei wird für den Parameter $d_0(\mathbf{c}, \mathbf{y}, t)$ zur Vereinfachung des Modells angenommen, dass nur dann eine Lagererhöhung erfolgen kann, sofern Anfragen nach Produkten $j \in \mathcal{J}$ möglich sind ($p_j(t) > 0$).

Die Gleichung (8) kann analog der Gleichung (1) umgeformt werden:

$$\begin{aligned} V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t) &= \sum_{j \in \mathcal{J}} p_j(t) V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1) + \sum_{j \in \mathcal{J}} p_j(t) \max[0, \\ &\quad r_j - V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1) + V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j, \mathbf{y}^{\hat{t}}, t-1), \\ &\quad r_j - V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1) + V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}} - \mathbf{a}_j, t-1), \\ &\quad V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j, \mathbf{y}^{\hat{t}} + Y(\mathbf{a}_j), t-1) - V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1)] \\ &\quad + p_0(t) V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1) \\ &\quad + p_0(t) \max_{\substack{j \in \mathcal{J}, \\ p_j(t) > 0}} [0, V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j, \mathbf{y}^{\hat{t}} + Y(\mathbf{a}_j), t-1)] \\ &= V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1) + \sum_{j \in \mathcal{J}} p_j(t) \max[0, \\ &\quad r_j - V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1) + V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j, \mathbf{y}^{\hat{t}}, t-1), \\ &\quad r_j - V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1) + V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}} - \mathbf{a}_j, t-1), \\ &\quad V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j, \mathbf{y}^{\hat{t}} + Y(\mathbf{a}_j), t-1) - V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1)] \\ &\quad + p_0(t) \max_{\substack{j \in \mathcal{J}, \\ p_j(t) > 0}} [0, V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j, \mathbf{y}^{\hat{t}} + Y(\mathbf{a}_j), t-1)] \end{aligned} \quad (11)$$

Die Beschreibung der Funktionsweise der Modellerweiterung wird anhand nachfolgendem Beispiels getätigt:

$$j = \{1, 2, 3\}, h^{\hat{t} \forall \hat{T}} = \{1\}, r_1 = 100, r_2 = 200, r_3 = 5000, \\ \text{Startperiode } t = 3, \hat{T} = \{1 : \{3\}, 2 : \{2\}, 3 : \{1\}\},$$

$$\mathbf{c}^{\hat{t}} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{a}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{a}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{a}_3 = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}, p_j(t) = \begin{pmatrix} 0.3 & 0 & 0 \\ 0.3 & 0.3 & 0 \\ 0.3 & 0.3 & 0.3 \end{pmatrix}, \\ \mathbf{y} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \mathbf{y}^{max} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$$

Bei dem Beispiel wird ein Netzwerk mit drei Produkte $j \in \mathcal{J}$ betrachtet. Es existieren drei Leistungserstellungszeitpunkte \hat{t} für eine Ressource $h^{\hat{t}} \in \mathcal{H}$. Die Produkte generieren einen Ertrag in Höhe von $r_1 = 100, r_2 = 200$ und $r_3 = 5000$. Die Kapazitäten für die Ressourcen betragen $\mathbf{c} = (1, 1, 1)$ und die Bestandsveränderungen für die Anfragen betragen $\mathbf{a}_1 = (1, 0, 0)$, $\mathbf{a}_2 = (0, 1, 0)$ sowie $\mathbf{a}_3 = (0, 0, 2)$. Der Buchungshorizont beträgt $T = 3$. Es existiert kein Lagerbestand \mathbf{y} und es kann nur ein maximaler Lagerbestand $\mathbf{y}^{max} = (2, 2, 2)$ zu jeder Leistungserstellung aufgebaut werden. In diesem Beispiel sind nicht alle Produkthanfragen $j \in \mathcal{J}$ zu jedem Zeitpunkt $t \in T$ möglich. Dies zeigt die Matrix $p_j(t)$ mit den Eintrittswahrscheinlichkeiten für jedes Produkt j zum jeweiligen Zeitpunkt t . Für das Produkt $j = 1$ treffen Anfragen zu den Zeitpunkten $t \in T$ mit den Wahrscheinlichkeiten $p_1(t) = (0.3, 0, 0)$ ein. Für die Produkte $j = 2$ und $j = 3$ betragen die Wahrscheinlichkeiten $p_2(t) = (0.3, 0.3, 0)$ bzw. $p_3(t) = (0.3, 0.3, 0.3)$. Bei den Wahrscheinlichkeiten $p_j(t)$ der Produkthanfrage j zum Zeitpunkt t muss beachtet werden, dass der Buchungshorizont rückwärts verläuft. Damit gilt $p_j(t) = (p_j(3), p_j(2), p_j(1))$. Abbildung 11 zeigt für das Beispiel die möglichen Systemzustände mit allen Übergängen. Dabei wird ein Systemzustand im Netzwerk als Zahlenfolge $[c_1 \ c_2 \ c_3 \ y_1 \ y_2 \ y_3 \ t]$ beschrieben.

Wie in der Abbildung 11 zu erkennen ist, wäre die optimale Politik $d_j(\mathbf{c}, \mathbf{y}, t)$ im Systemzustand $[1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 3]$ eine Anfrage nach einem Produkt $j = 1$ abzulehnen und eine Bestandsveränderung des Lagers zu bewilligen (Lagerproduktion) sowie die Annahme der Produkthanfrage j anhand der Kapazitäten. Eine Anfrage $j = 3$ kann in diesem Systemzustand nicht akzeptiert werden, da nicht genügend Kapazität vorhanden ist und daher wird der Übergang in einem Systemzustand mit einer negativen Ressourcenkapazität nicht dargestellt. Sofern eine Anfrage $j = 1$ im Systemzustand $[1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 3]$ eintrifft, erreicht das Netzwerk den System-

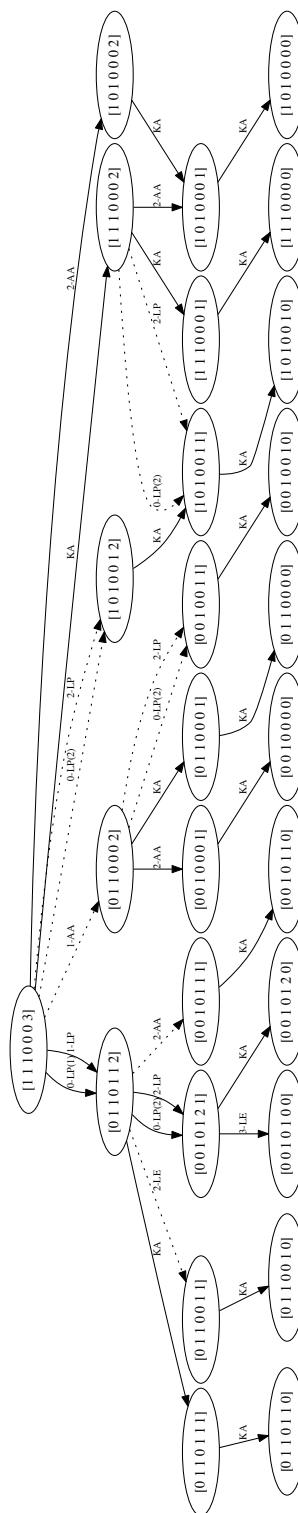


Abbildung 11 Darstellung der Systemzustände des Netzwerk RM unter Beachtung der Möglichkeit der Auftragsannahme- und Lagerhaltungsentscheidung

Legende: Die Zahlen stehen für den Auftrag j , AA='Auftragsannahme', LE='Lagerentnahme', LP='Lagerproduktion', KA='Kein Auftrag', ...='Anfrage ablehnen'

zustand $[0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 2]$. Wie zu erkennen, ist die Kapazität $c_1^{\hat{t}=1}$ auf 0 reduziert und nach dem Leistungserstellungszeitpunkt $\hat{t} = 1$ ist ein Ressourcenbestand auf dem Lagerhaltungsparameter \mathbf{y} verfügbar. Zum Systemzustand $[0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 2]$ ist aufgrund der Wahrscheinlichkeitsverteilung $p_j(t)$ keine Anfrage nach Produkt $j = 2$ mehr möglich und ebenfalls ist eine Annahme einer Anfrage nach Produkt $j = 3$ aufgrund der Kapazitätsbestände ausgeschlossen. In diesem Systemzustand ist, sofern die Anfrage eintrifft, das Ablehnen einer Produkthanfrage $j = 2$ und die aus der Ablehnung mögliche Lagerproduktion alleiniges Bestandteil der optimalen Politik. Aufgrund einer solchen Entscheidung erfolgt der Übergang in den Systemzustand $[0 \ 0 \ 1 \ 0 \ 1 \ 2 \ 1]$. Wie die Zahlenfolge $[0 \ 0 \ 1 \ 0 \ 1 \ 2 \ 1]$ des Systemzustands andeutet, ist anhand des Lagerhaltungsparameters $\mathbf{y} = (0, 1, 2)$ erstmalig eine Annahme der Produkthanfrage $j = 3$ möglich. Sofern diese aufgrund der Wahrscheinlichkeitsverteilung $p_j(t)$ einzig mögliche Anfrage eintrifft, ist die optimale Politik des Netzwerks die Annahme der Anfrage $j = 3$ anhand der Entscheidung der Lagerentnahme $(d_3((0, 0, 1)^T, (0, 1, 2)^T, 1) = 2)$. Erfolgt eine solche Reihenfolge des Eintreffens der Aufträge, dass würde ein Unternehmen unter Einhaltung der optimalen Politik einen Gesamtdeckungsbeitrag in Höhe von 5000 GE erzielen.

Sofern im Ausgangssystemzustand $[1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 3]$ eine Anfrage nach einem Produkt $j = 2$ eintrifft, ist die optimale Politik die Annahme der Anfrage über die Ressourcenkapazität $\mathbf{c}^{\hat{t}}$. Aufgrund dieser Entscheidung ist ein Gesamtdeckungsbeitrag in Höhe von 200 GE erzielt. Eine Entscheidung über die Ablehnung und Aufarbeitung von Ressourcen (Lagerproduktion) unter diesen Parametergegebenheiten hätte in Bezug der Zielsetzung der Maximierung des Gesamtdeckungsbeitrags keinen Nutzen. Wie in der Abbildung 11 zu erkennen ist, würde eine Lagerproduktion zwar den Bestand an Ressourcen auf Lager erhöhen, jedoch wäre im weiteren Verlauf keine Annahme einer anderen Produkthanfrage $j \in \mathcal{M}$ möglich. Sofern ein Unternehmen sich gegen die optimale Politik entscheidet, würde kein Ertrag im weiteren Verlauf des Netzwerks generiert.

Trifft im Ausgangssystemzustand $[1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 3]$ keine Anfrage j ein, dann ist die optimale Politik $d_0(\mathbf{c}, \mathbf{y}, t) = 1$. D. h. sofern keine Anfragen nach einem Produkt j eintreffen, wäre es die optimale Politik des Unternehmens die Bestandsveränderung \mathbf{a}_j der Produkthanfrage $j = 1$ zu verwenden, damit der Lagerbestand \mathbf{y} sich ab der Leistungsperiode $\hat{t} = 2$ um eine Einheit erhöht. Anders formuliert, würde das Unternehmen die vorhandene Kapazität nicht verfallen lassen und eine Lagerproduktion veranlassen. Die weitere optimale Politik für dieses Netzwerk ist, sofern keine Produkthanfragen eintreffen, das Verfolgen einer optimalen Politik, die eine

Tabelle 4 Optimale Politik für das beispielhafte Netzwerk RM unter Beachtung der Möglichkeit der Auftragsannahme- und Lagerhaltungsentscheidung

c^1	c^2	c^3	y^1	y^2	y^3	t	ExpValue	d_0	d_1	d_2	d_3
1	1	1	0	0	0	3	497.999969482	1	3	1	0
1	0	1	0	0	0	2	0.0	0	0	0	0
1	0	1	0	0	1	2	0.0	0	0	0	0
0	1	1	0	1	1	2	1050.0	2	0	3	0
0	1	1	0	0	0	2	60.0	0	0	1	0
1	1	1	0	0	0	2	60.0	0	0	1	0
0	0	1	0	1	2	1	1500.0	0	0	0	2
1	1	1	0	0	0	1	0.0	0	0	0	0
0	1	1	0	1	1	1	0.0	0	0	0	0
1	0	1	0	0	0	1	0.0	0	0	0	0
1	0	1	0	0	1	1	0.0	0	0	0	0
0	1	1	0	0	0	1	0.0	0	0	0	0
0	1	1	0	0	1	1	0.0	0	0	0	0
0	0	1	0	0	1	1	0.0	0	0	0	0
0	0	1	0	1	1	1	0.0	0	0	0	0

Annahme der Produkthanfrage $j = 3$ ermöglicht. Tabelle 4 zeigt zusammenfassend die berechneten Erwartungswerte und die optimale Politik für alle Systemzustände des Beispiels.

Die eigentliche Funktionsweise des hier vorgestellten Modells geht anhand des vorher aufgeführten Beispiels jedoch teilweise verloren, da die Anzahl an Buchungsperioden $t \in T$ der Anzahl der möglichen Buchungsabschnitte $\bar{t} \in \bar{T}$ gleich ist. Zur besseren Veranschaulichung wird ein umfangreicheres Beispiel berechnet:

$$j = \{1, 2, 3, 4\}, h^{\hat{t} \forall \hat{T}} = \{1\}, r_1 = 100, r_2 = 5000, r_3 = 100, r_4 = 5000, \\ \text{Startperiode } t = 10, \hat{T} = \{1 : \{10, 9, 8, 7, 6\}, 2 : \{5, 4, 3, 2, 1\}\},$$

$$\mathbf{c}^{\hat{t}} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{a}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{a}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{a}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \mathbf{a}_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

$$p_j(t) = \begin{pmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{pmatrix},$$

$$\mathbf{y} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mathbf{y}^{max} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Für dieses Beispiel ist eine grafische Auswertung unter Beachtung der Parameter und der Menge an möglichen Systemübergängen nicht mehr zielführend. Eine Auswertung der optimalen Politik anhand einer tabellarischen Auswertung ist jedoch weiterhin möglich, wie Tabelle 5 zeigt. Die Tabelle zeigt die jeweiligen Erwartungswerte und die optimalen Politik für jeden Systemzustand. Dabei wird in diesem Beispiel ein Systemzustand als Zahlenfolge $[c_1^{\hat{t}} c_2^{\hat{t}} y_1^{\hat{t}} y_2^{\hat{t}} t]$ definiert.

Wie aus der Tabelle 5 zu erkennen ist, ermittelt der Algorithmus keine optimale Politik der Instandsetzung von Lagerbeständen für die Aufträge $j = 3$ und $j = 4$. Dies liegt daran, dass die Bereitstellung des aufgewerteten Lagerbestands erst außerhalb des Buchungshorizonts T verfügbar ist. Es liegt eine Dominanz der Annahmen von $j = 2$ bzw. $j = 4$ für das gesamte Netzwerk vor, sofern die Wahrscheinlichkeit des Auftragseingangs besteht. Zur den anfänglichen Buchungsperiode gehört die Annahme der ertragsarmen Anfragen $j = 1$ und $j = 3$ nicht zur optimalen Politik. Jedoch könnte die Kapazität $c_{h^{\hat{t}}}^{\hat{t}}$ der Produkthanfrage $j = 1$ genutzt werden, damit ein Lagerbestand $y_{h^{\hat{t}}}$ über die Ressource $h^{\hat{t}}$ generiert wird. Die Kapazität wäre demnach nach dem Leistungserstellungszeitpunkt $\hat{t} = 1$ verfügbar. Erfolgt zum Zeitpunkt $t = 10$ der Auftragseingang von $j = 1$, dann ist eine Annahme über die Kapazitäten jedoch nicht die optimale Politik, sonder die komplette Ablehnung des Auftrags. D. h. der Kapazitätsverbrauch a_j der Anfrage j wird nicht aufgewendet um den Lagerbestand y um $Y(a_j)$ zu erhöhen. Die Entscheidung über eine Lagerproduktion, die aufgrund des Eingangs einer Produkthanfrage $j = 1$ getätigt werden kann, erfolgt erst ab der Periode $t = 7$. Für das Netzwerk wandeln sich ab diesem Zeitpunkt die OK in der Form, dass eine Übertragung der Kapazität bzgl. der Zielvorgabe der Maximierung des zu erwartenden Gesamtertrags sinnvoller ist. Damit wäre ein Lagerbestand für weitere Produkthanfragen $j = 2$ (und $j = 4$) möglich und die Kapazität $c_{h^{\hat{t}}}^{\hat{t}}$ aus $\hat{t} = 1$ wäre aufgrund des ablaufenden Buchungshorizonts weiterhin verfügbar.

Sofern keine Anfragen in dem Beispiel eintreffen, ergibt sich ein differenziertes Bild bzgl. der optimalen Politik. Sofern Kapazitäten die $c_{h^{\hat{t}}}^{\hat{t}}$ zu den Zeitpunkten $t < 5$ vollständig verfügbar sind, wird keine Lagerproduktion durchgeführt. Das Modell versucht weiter den Ertrag durch die möglichen Erträge r_j der Produktnachfragen $j = 2$ und $j = 4$ zu maximieren. Sofern jedoch in diesem Zeitabschnitt eine Anfrage nach dem Produkt $j = 2$ oder $j = 4$ eingetroffen und dementsprechend die Kapazität reduziert ist, wird das Modell die Kapazität $c_{h^{\hat{t}}}^{\hat{t}-1} = 1$ zum Lagerbestand $y_{h^{\hat{t}}}^{\hat{t}-1}$ umschichten. Dies erfolgt mit dem Bestandveränderungsparameter a_j für $j = 1$ und nur dann, wenn keine Anfrage nach den Produkten $j \in \mathcal{J}$ in den Zeitpunkten

Tabelle 5 Optimale Politik für das zweite beispielhafte Netzwerk RM unter Beachtung der Möglichkeit der Auftragsannahme- und Lagerhaltungsentscheidung

c^1	c^2	y^1	y^2	t	ExpValue	d_0	d_1	d_2	d_3	d_4
1	1	0	0	10	8528.45019531	0	0	1	0	1
1	0	0	0	9	4245.92529297	1	3	1	0	0
0	1	0	0	9	4332.25	0	0	0	0	1
1	1	0	0	9	8021.45019531	0	0	1	0	1
0	1	0	1	9	7161.75	0	0	0	0	1
0	0	0	1	8	4165.375	0	0	0	0	2
1	1	0	0	8	7331.45019531	0	0	1	0	1
0	1	0	1	8	6661.0	0	0	0	0	1
0	1	0	0	8	4165.375	0	0	0	0	1
0	0	0	0	8	0.0	0	0	0	0	0
1	0	0	0	8	3949.65014648	1	3	1	0	0
1	0	0	0	7	3417.60009766	1	3	1	0	0
1	1	0	0	7	6428.45019531	1	3	1	0	1
0	0	0	1	7	3956.75	0	0	0	0	2
0	1	0	0	7	3956.75	0	0	0	0	1
0	1	0	1	7	6087.0	0	0	0	0	1
0	0	0	0	7	0.0	0	0	0	0	0
1	0	0	0	6	2348.0	1	3	1	0	0
0	0	0	1	6	3696.0	0	0	0	0	2
1	1	0	0	6	5228.20019531	1	3	1	0	1
0	1	0	1	6	5434.5	0	0	0	0	1
0	0	0	0	6	0.0	0	0	0	0	0
1	0	0	0	5	0.0	0	0	0	0	0
0	1	0	1	5	4701.0	0	0	0	0	1
0	1	0	0	5	3370.0	0	0	0	0	1
0	0	0	0	5	0.0	0	0	0	0	0
1	1	0	0	5	3370.0	0	0	0	0	1
0	0	0	1	5	3370.0	0	0	0	0	2
0	0	0	1	4	2962.5	0	0	0	0	2
0	1	0	0	4	2962.5	0	0	0	0	1
0	1	0	1	4	3886.0	0	0	0	0	1
1	0	0	0	4	0.0	0	0	0	0	0
1	1	0	0	4	2962.5	0	0	0	0	1
0	0	0	0	4	0.0	0	0	0	0	0
1	0	0	0	3	0.0	0	0	0	0	0
1	1	0	0	3	2453.0	0	0	0	0	1
0	1	0	1	3	2995.0	0	0	0	0	1
0	1	0	0	3	2453.0	0	0	0	0	1
0	0	0	1	3	2453.0	0	0	0	0	2
0	0	0	0	3	0.0	0	0	0	0	0
0	0	0	1	2	1816.0	0	0	0	0	2
0	1	0	1	2	2040.0	0	0	0	1	1
0	0	0	0	2	0.0	0	0	0	0	0
1	1	0	0	2	1816.0	0	0	0	0	1
1	0	0	0	2	0.0	0	0	0	0	0
0	1	0	0	2	1816.0	0	0	0	0	1
0	0	0	1	1	1020.0	0	0	0	2	2
0	1	0	0	1	1020.0	0	0	0	1	1
0	1	0	1	1	1020.0	0	0	0	1	1
0	0	0	0	1	0.0	0	0	0	0	0
1	1	0	0	1	1020.0	0	0	0	1	1
1	0	0	0	1	0.0	0	0	0	0	0

$t > 5$ eintreffen. Damit folgt die optimale Politik $d_0(\mathbf{c}, \mathbf{y}, t)$ der optimalen Politik $d_j(\mathbf{c}, \mathbf{y}, t)$.

6 Numerische Untersuchung

Für die numerische Untersuchung wurden mehrere Berechnungen der optimalen Politik für bestimmte Kombinationen von Eigenschaften der Produkthanfragen und verschiedener Wahrscheinlichkeitsverteilungen durchgeführt. Ziel ist es zu untersuchen, ob durch die Möglichkeit der Lagerhaltung eine Veränderung in der optimalen Politik resultiert und sofern es eine Veränderung gibt, in welcher Ausgestaltungsform diese eintritt. Die Ergebnisse der Berechnungen, die in dieser Arbeit vorgestellt sind, wurden mithilfe des Clustersystems an der Leibniz Universität Hannover berechnet. Tabelle 6 zeigt die verwendete Rechnersysteme, wobei eine Berechnung jeweils auf einem Knoten der verfügbaren Rechnersysteme durchgeführt wurde.

Tabelle 6 Überblick über Publikationen des traditionellen Konzepts des Revenue Managements in der Fertigungsindustrie

Cluster	Knoten	Prozessoren	Kerne/ Knoten	Speicher/ Knoten (GB)
Tane	96	2x Intel Westmere-EP Xeon X5670 (6-cores, 2.93GHz, 12MB Cache, 95W)	12	48
Taurus	54	2x Intel Westmere-EP Xeon X5650 (6-cores, 2.67GHz, 12MB Cache, 95W)	12	48
SMP	9	4x Intel Westmere-EX Xeon E7-4830 (8-cores, 2.13GHz, 24MB Cache, 105W)	32	256
	9	4x Intel Backton Xeon E7540 (6-cores, 2.00GHz, 18MB Cache, 105W)	24	256
	3	4x Intel Westmere-EX Xeon E7-4830 (8-cores, 2.13GHz, 24MB Cache, 105W)	32	1024

Die in der Arbeit verwendete Implementierung des Auftragsannahmeproblems im Netzwerk RM mit Lagerhaltungsentscheidung sieht die Berechnung der möglichen Systemzustände und des maximal möglichen Erwartungswerts vor. Mithilfe der

möglichen Systemzustände wird die Memofunktion gespeist (Vgl. Kapitel ???), die wiederum bei der Berechnung der Erwartungswerte für die benötigten Systemzustände notwendig ist. Bei der Berechnung der Erwartungswerte handelt es sich um das DP. Zusätzlich erfolgt nach Berechnung der Erwartungswerte die Übertragung und Speicherung der optimalen Politik für jeden Systemzustand in eine Datenbank. Aufgrund der zwei Berechnungsschritte werden für beide Berechnungen jeweils die ermittelten Daten bzw. Berechnungskennzahlen angegeben. Auf dem Rechnersystem wurden folgende Versionen der verwendeten Software-Pakete genutzt:

```
Python Version 2.7.9 (default, Jul 2 2015, 11:24:04) [GCC 4.9.2]
Numpy Version 1.9.2
Matplotlib Version 1.4.3
Pandas Version 0.16.2
```

Nachfolgend sind alle berechneten Beispiele anhand von Szenarien dargestellt. Je Szenario gibt es eine Anzahl an Buchungsperioden T und eine Anzahl an Leistungserstellungszeitpunkten \hat{T} . In dieser Arbeit wird nur eine Ressource $h \in \mathcal{H}$ betrachtet, was sich ebenfalls in den Szenarien widerspiegelt. Für einen jeden Leistungserstellungszeitpunkt $\hat{t} \in \hat{T}$ gibt es eine Anzahl an Kapazitäten $c_h^{\hat{t}}$, einen Lagerbestand $y_h^{\hat{t}}$ und einen maximal möglichen Lagerbestand $y_h^{\hat{t},max}$ der Ressource $h = 1$. Die unterschiedlichen Produkthanfragen $j \in \mathcal{J}$ haben dabei einen individuellen Ertrag r_j , einen Parameter für die Inanspruchnahme bzw. Bestandsveränderung der Kapazitäten $a_h j$ der Ressource $h = 1$ und einen Zeitpunkt der Leistungserstellung \hat{t} . Die Szenarien besitzen für die Buchungsperioden $t \in T$ eine bestimmte Wahrscheinlichkeitsverteilung $p_j(t)$ über den Auftragseingang einer jeden Produkthanfrage $j \in \mathcal{J}$. Die Verteilungen ist jeweils beim Produkt j angegeben und über alle Produkte $j \in \mathcal{J}$ des jeweiligen Szenarios als grafischer Kurvenverlauf abgebildet.

Im ersten Szenario werden über einen Buchungszeitraum $T = 100$ insgesamt zehn Produkthanfragen $j \in \mathcal{J}$ betrachtet. Jeweils zwei Produkthanfragen j sind für eine der fünf möglichen Leistungserstellungen vorgesehen und die Anfragen wechseln sich dabei jeweils zwischen dem möglichen Ertrag $r_j = 100$ und $r_j = 1000$ ab. Die Wahrscheinlichkeitsverteilung einer jeden Produkthanfrage verläuft dabei linear mit dem konstanten Wert $p_j(t)$ über den Buchungshorizont T , wobei mit Überschreitung der zur Produkthanfrage j zugehörigen Leistungserstellung \hat{t} die Eintrittswahrscheinlichkeit der Produkthanfrage auf $p_j(t) = 0$ sinkt (Vgl. Abbildung 12).

Tabelle 7 Szenario 1

T	\hat{T}	h	$c_h^t \forall t \in \hat{T}$	$y_h^t \forall t \in \hat{T}$	$y_h^{t,max} \forall t \in \hat{T}$
100	5	1	1	0	2
j	r_j	a_{1j}	\hat{t}	Verteilung $p_j(t)$	
1	100	1	1	Linear verlaufend (0.09)	
2	1000	1	1	Linear verlaufend (0.09)	
3	100	1	2	Linear verlaufend (0.09)	
4	1000	1	2	Linear verlaufend (0.09)	
5	100	1	3	Linear verlaufend (0.09)	
6	1000	1	3	Linear verlaufend (0.09)	
7	100	1	4	Linear verlaufend (0.09)	
8	1000	1	4	Linear verlaufend (0.09)	
9	100	1	5	Linear verlaufend (0.09)	
10	1000	1	5	Linear verlaufend (0.09)	
Rechenzeit Systemzustände (h):				0:00:00.170329	
Anzahl möglicher Systemzustände:				785376	
Anzahl benötigter Systemzustände:				76397	
Rechenzeit DP (h):				16:24:47.128239	
Max. Erwartungswert (GE):				10858.2	

Tabelle 7 zeigt die verwendeten Parameter des Szenarios und das Ergebnis der Berechnung. Das vollständige Ergebnis ist im Anhang ??? nachzulesen.

Tabelle 8 zeigt die Auswertung der optimalen Politik die sich aufgrund der Parameter und der Wahrscheinlichkeitsverteilung des Szenarios 1 ergeben. Es handelt sich um eine Auswertung der absoluten Anzahl der Entscheidungen für die optimale Politik sofern keine Anfrage eintrifft (d_0) und sofern Anfragen eintreffen ($d_j \forall j \in \mathcal{J}$). Wie in der Tabelle 8 zu erkennen, werden nur die Bestandsveränderungsparameter a_j mit niedrigem Ertrag r_j für eine mögliche Lagerhaltungsentscheidung verwendet. Dabei kann die Annahme getroffen werden, dass bei fortschreiten des Buchungshorizonts T die Entscheidung über eine mögliche Lagerproduktion öfters getroffen

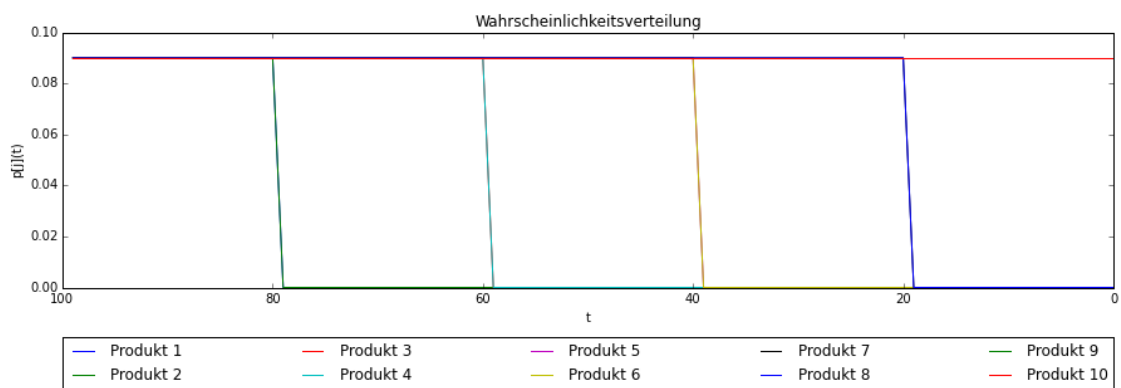


Abbildung 12 Wahrscheinlichkeitsverteilung Szenario 1

Tabelle 8 Auswertung des Szenarios 1

j	0	1	2	3	4	5	6	7	8	9	10
d_0	57767	1461	0	5048	0	7960	0	3367	0	0	0

d_j	Auftrags- ablehnung (0)	Auftrags- annahme (1)	Lager- entnahme (2)	Lager- produktion (3)
d_1	74038	104	0	1461
d_2	73850	292	0	1461
d_3	69657	208	528	5210
d_4	60889	404	11262	3048
d_5	65333	304	1136	8830
d_6	46350	659	24820	3774
d_7	69725	216	2583	3079
d_8	27421	10707	37475	0
d_9	70577	2440	2586	0
d_{10}	13013	29235	33355	0

wird. Bei der optimalen Politik d_j ergibt sich ein ähnliches Bild. Tendenziell werden öfters Anfragen mit niedrigem Ertrag abgelehnt und eine Lagerproduktion durchgeführt.

7 Schlussbemerkung

Anhang

.1 Programmcode

Quellcode 1 Parameter Definition für die Dynamische Programmierung im
Netzwerk Revenue Management

```
# coding: utf-8

__version__ = '1.0'
__author__ = "Robert Matern (r.matern@stud.uni-hannover.de)"
__date__ = 'Sonntag, 16. August 2015 (MESZ)'
__copyright__ = "(C) 2015 Robert Matern"

import numpy as np
import datetime

# Parameter

def Product(j):
    '''Produkte 'j' aus 'J'. '''
    return np.arange(j+1, dtype=np.uint16)

def Resource(h):
    '''Ressourcen 'h' aus 'H'. '''
    return np.arange(h+1, dtype=np.uint16)

def Capacity(h):
    '''Kapazität 'c[h]' der Ressourcen 'h' aus 'H'. '''
    matrix = np.zeros(shape=(len(h)), dtype=np.int16)
    return matrix

def Consumption(j, h):
```

```

'''Matrix 'A' mit den Parametern der Ressourcen-
verbräuchen 'a[j,h]' für die Produkte 'j' aus 'J'
und die Ressourcen 'h' aus 'H'.'''
matrix = np.zeros(shape=(len(j),len(h)), dtype=np.uint16)
return matrix

def Revenue(j):
    '''Erträge 'r[j]' der Produkte 'j' aus 'J'.'''
    vector = np.zeros(shape=(len(j)), dtype=np.uint16)
    return vector

def Time(t):
    '''Buchungshorizont 'T' mit einem Buchungs-
zeitpunkt 't'.'''
    time = np.arange(t+1, dtype=np.uint16)
    time = time[::-1]
    return time

def Prob(j, t):
    '''Matrix 'Prob' mit den Parametern 'p[j,t]', die
den Wahrscheinlichkeiten der Nachfrage nach einem
Produkt 'j' in der Buchungsperiode 't' entspricht.

    ACHTUNG: Perioden aufsteigend sortiert!!!'''
    matrix = np.zeros(shape=(len(j),len(t)), dtype=np.float32)
    return matrix

def Against_Prob(prob):
    '''Gegenwahrscheinlichkeit keiner Produkthanfrage.
Dabei entspricht der Paramter 'prob'
der Wahrscheinlichkeit der Produkthanfragen.'''
    matrix = prob.T
    matrix2 = np.zeros(shape=(len(matrix)), dtype=np.float32)
    for i in range(0, len(matrix)):
        if np.greater(np.sum(matrix[i]), np.float32(1.0)):
            raise ValueError('Wahrscheinlichkeit ist > 1 in der ←
Buchungsperiode %i.' % i)
        else:
            matrix2[i] = np.subtract(np.float32(1.0), np.float32←
(np.sum(matrix[i])))
    return matrix2

```


Mögliche Systemzustände

```
def cartesian(arrays, out=None):
    """
    Generate a cartesian product of input arrays.

    Quelle: http://stackoverflow.com/a/1235363/4913569

    Parameters
    _____
    arrays : list of array-like
        1-D arrays to form the cartesian product of.
    out : ndarray
        Array to place the cartesian product in.

    Returns
    _____
    out : ndarray
        2-D array of shape (M, len(arrays)) containing cartesian←
        products
        formed of input arrays.
    """

    arrays = [np.asarray(x) for x in arrays]
    dtype = arrays[0].dtype

    n = np.prod([x.size for x in arrays])
    if out is None:
        out = np.zeros([n, len(arrays)], dtype=dtype)

    m = n / arrays[0].size
    out[:,0] = np.repeat(arrays[0], m)
    if arrays[1:]:
        cartesian(arrays[1:], out=out[0:m,1:])
        for j in xrange(1, arrays[0].size):
            out[j*m:(j+1)*m,1:] = out[0:m,1:]

    return out

def Condition(capacities, resources, times):
    '''Berechnung der möglichen Systemzustände eines Netzwerks
    mit c Kapazitäten, r Ressourcen und t Perioden.'''
```

```
a = datetime.datetime.now()

arrays = [range(i) for i in capacities[1:]+1]
arrays.append(times)
for i in arrays:
    np.append(arrays, np.array(i))

conditions = cartesian(arrays)

b = datetime.datetime.now()
print (b-a)
return conditions[::-1]

# Lagerparameter

def Stock_Resource(h):
    '''Lagerbestand 'y[h]' der Ressource 'h' aus 'H'. '''
    vector = np.zeros(shape=(len(h)), dtype=np.int16)
    return vector

def Max_Stock_Resource(h):
    '''Max. Lagerbestand 'ymax[h]' der Ressource 'h' aus 'H'. '''
    vector = np.zeros(shape=(len(h)), dtype=np.int16)
    return vector

def Condition_Storage(capacities, resources, stocks, times):
    '''Berechnung der möglichen Systemzustände eines Netzwerks
    mit c Kapazitäten, r Ressourcen und t Perioden.'''

    a = datetime.datetime.now()

    arrays = [range(i) for i in capacities[1:]+1]
    stock_arrays = [range(i) for i in stocks[1:]+1]
    for i in stock_arrays:
        arrays.append(i)
    arrays.append(times)

    conditions = cartesian(arrays)

    b = datetime.datetime.now()
    print 'Rechenzeit für Systemzustände:', (b-a)
    return conditions[::-1]
```

```
# Leistungsperioden

def Performance_Time(times, resources):
    array = np.split(times[:-1], len(resources[:-1]))
    return array
```

Quellcode 2 Python-Algorithmus für die Dynamische Programmierung im Netzwerk Revenue Management

```
# coding: utf-8

__version__ = '1.0'
__author__ = "Robert Matern (r.matern@stud.uni-hannover.de)"
__date__ = 'Sonntag, 16. August 2015 (MESZ)'
__copyright__ = "(C) 2015 Robert Matern"

import numpy as np
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
import graphviz as gv
import datetime

# Dynamisches Programm

def DP_Stock(solutions, conditions, products, resources, ←
capacities, consumptions, times, revenues, probs, stocks, ←
max_stocks):
    '''Berechnung der Erwartungswerte des ←
    Auftragsannahmeproblems inkl. Lagerhaltung.'''

    # Regeneration der Kapazitäten in dieser Rechnung nicht mö←
    # glich!!!

    # Leere Wert für den Erwartungswert des aktuellen ←
    # Systemzustands.
    value = 0

    #Aktueller Systemzustand wird generiert.
    condition = np.array(np.hstack((capacities[1:], stocks[1:], ←
    times[0])), dtype=np.int16)

    #Aktueller Systemzustand wird in dem 'np.array' aller mö←
```

```

    gleichen Systemzustände gesucht.
state = np.where((conditions == condition).all(axis=1))←
    [0][0]

# Memofunktion: Die Rechnung wird nur fortgeführt, sofern es←
    nicht schon berechnet wurde.
if state not in solutions:

    # Sofern es sich nicht um einen Endknoten des ←
        Entscheidungsbaums handelt,
    # werden folgende Schritte eingeleitet:
    if times[0]!=0:
        # Der Erwartungswert t-1 ohne Akeptanz wird ←
            ermittelt und im Wert 'reject' gespeichert.
        reject = DP_Stock(solutions, conditions, products, ←
            resources, capacities, consumptions, times[1:], ←
            revenues, probs, stocks, max_stocks)

        # Für die Aufträge wird ein Numpy-Array in der Länge←
            der Anzahl an Produkten erstellt.
        max_order = np.zeros(shape=(len(products[1:])), ←
            dtype=np.float16)
        accept = np.zeros(shape=(len(products[1:])), dtype=←
            np.float16)
        index_order = np.zeros(shape=(len(products)), dtype=←
            np.uint8)
        oc_order = np.zeros(shape=(len(products[1:])), dtype←
            =np.float16)

        reject_list = np.zeros(shape=(len(products[1:])), ←
            dtype=np.float16)

        # For-Schleife über alle Produkte, sofern die ←
            Kapazitäten keinen negativen Werte annehmen.
        product = (j for j in products[1:] if probs[j][times←
            [0]] > 0)
        for j in product:

            change = capacities-consumptions[j]
            decrease_stock = stocks-consumptions[j]

            if np.all((change) >= np.float32(0)):
                # Initialisierung der Rechnung für t-1 mit ←
                    Akeptanz jeweils für ein Produkt j.
                accept_j = DP_Stock(solutions, conditions, ←

```

```

        products, resources, change, consumptions, ←
        times[1:], revenues, probs, stocks, ←
        max_stocks)
    oc = reject-accept_j
    order = np.float32(revenues[j]-reject+←
        accept_j)
else:
    # Erwartungswert für ein Produkt j entspricht
    # der Grenzbedingung  $V(c,y,t)=\text{infinitely}$ , ←
    # falls  $c[j] < 0$ .
    order = np.float16(0)
    oc = np.float16(0)

if np.all((decrease_stock) >= np.float32(0)):
    # Initialisierung der Rechnung für t-1 mit ←
    # Akeptanz jeweils für ein Produkt j.
    accept_stock_j = DP_Stock(solutions, ←
        conditions, products, resources, ←
        capacities, consumptions, times[1:], ←
        revenues, probs, decrease_stock, ←
        max_stocks)
    oc_stock = reject-accept_stock_j
    order_stock = np.float32(revenues[j]-reject+←
        accept_stock_j)
else:
    # Erwartungswert für ein Produkt j entspricht
    # der Grenzbedingung  $V(c,y,t)=\text{infinitely}$ , ←
    # falls  $c[j] < 0$ .
    order_stock = np.float16(0)
    oc_stock = np.float16(0)

change_stock = np.cumsum(np.insert([np.array(←
    consumptions[j], dtype=np.uint8)], 0, 0)[: -1])
increase_stock = np.copy(stocks)
increase_stock = np.array(increase_stock+←
    change_stock, dtype=np.int16)

if np.all((change) >= np.float32(0)) and np.all(←
    ((increase_stock) <= (max_stocks)) and np.←
    count_nonzero(change_stock)!=0:
    # Initialisierung der Rechnung für t-1 mit ←
    # Akeptanz jeweils für ein Produkt j.
    accept_workup_j = DP_Stock(solutions, ←
        conditions, products, resources, change, ←
        consumptions, times[1:], revenues, probs, ←
        increase_stock, max_stocks)

```

```

        oc_workup = reject-accept_workup_j
        order_workup = np.float32(accept_workup_j-↵
            reject)
        reject_list[j-1] = order_workup
    else:
        # Erwartungswert für ein Produkt j entspricht
        # der Grenzbedingung  $V(c,y,t)=\text{infinitely}$ , ↵
        # falls  $c[j] < 0$ .
        order_workup = np.float16(0)
        oc_workup = np.float16(0)
        reject_list[j-1] = order_workup

    list_order = np.hstack((0, order, order_stock, ↵
        order_workup))
    max_order[j-1] = np.amax(list_order)
    index_order[j] = np.argmax(list_order)
    accept[j-1] = probs[j][times[0]]*max_order[j-1]
    list_oc = np.hstack((0, oc, oc_stock, oc_workup)↵
        )
    oc_order[j-1] = list_oc[index_order[j]]

# Sofern keine Anfrage eintrifft, kann ein ↵
# beliebiger Bestand verwendet werden.
list_reject = np.hstack((0, reject_list))
max_reject = np.amax(list_reject)
index_reject = np.argmax(list_reject)
index_order[0] = index_reject

# Summierung der Erwartungswerte ohne und mit ↵
# Akzeptanz.
value = np.float32(reject + np.sum(accept) + (probs↵
    [0][times[0]]*max_reject))
# Einzelne Optionen werden nach ihrem erw. Ertrag ↵
# indiziert.
index = np.argmax(max_order)
if np.all(max_order == 0):
    solutions[state] = [conditions[state], value, 0,↵
        0, 0, index_order]
else:
    solutions[state] = [conditions[state], value, ↵
        index+1, index_order[index+1], revenues[index↵
        +1]-oc_order[index], index_order]
print 'c[h]:', solutions[state][0][:len(resources)↵
    -1], '- y[h]:', solutions[state][0][len(resources↵
    )-1:-1], '- t:', solutions[state][0][-1], '- V(c,↵
    y,t):', solutions[state][1], '- d[0]:', solutions↵

```

```

        [state][5][:1], '- d[j]:', solutions[state][5][1:], '- j*:', solutions[state][2], '- OC[j*]:', solutions[state][4]
# Sofern es sich um einen Endknoten handelt, werden folgende Schritte eingeleitet:
else:
    # Erwartungswert entspricht der Grenzbedingung  $V(c, y, 0) = 0$ , für  $c \geq 0$ .
    value = np.float32(0)
    # Ein Endzustand wird mit einem Erwartungswert 0 in das Dict "solutions" gespeichert.
    solutions[state] = [conditions[state], value, 0, 0, 0, np.zeros(len(products), dtype=np.uint8)]
    #print 'c[h]:', solutions[state][0][:len(resources)-1], '- y[h]:', solutions[state][0][len(resources)-1:-1], '- t:', solutions[state][0][-1], '- d(c, y, t):', solutions[state][5], '- V(c, y, t):', solutions[state][1], '- j*:', solutions[state][2], '- OC[j*]:', solutions[state][4]

# Memofunktion: Sofern das Ergebnis bereits berechnet wurde, wird der Wert aus dem Dict "solutions" verwendet.
else:
    value = solutions[state][1]

return value

# Erstellung der Datenstruktur als NetworkX-Graph

def Structure_Stock(solutions, conditions, products, resources, consumptions, revenues, probs, stocks, max_stocks):
    '''Generierung der Stuktur der Problemstellung.'''

    if len(solutions) < 100:
        # MultiDiGraph erstellen.
        graph=nx.MultiDiGraph()

        # Knoten für alle Lösungen (solutions) erstellen.
        for key in solutions.iterkeys():
            graph.add_node(key, label=solutions[key][0], value=solutions[key][1],
                           capacity=solutions[key][0][:len(resources)-1], time=solutions[key][0][-1],
                           stock=solutions[key][0][len(resources)-1])

```

```

    )-1):-1])

# Endknoten zur Nutzung der NetworkX-Algorithmen anlegen
graph.add_node("end", label='end', op=0, value=0,
               capacity=0, stock=0, time=0)

# Kanten erstellen.
# Schleife über alle Lösungen.
for i in solutions.iterkeys():
    # Handelt es sich um einen Zeitpunkt 0, dann wird der Systemzustand mit dem 'end'-Knoten verbunden.
    if solutions[i][0][-1] == 0:
        graph.add_edge(i, "end", key=0, modus=0, weight=0, revenue=0, time=0)
    # Sonst führe die Schleife aus.
    else:
        # Aufgrund der differenzierten Auftragsannahme erfolgt eine Schleife über alle Produkte.
        for j in products:
            # Kapazitätsveränderung aufgrund der Anfragen nach Produkt 'j' wird erfasst.
            change = np.copy(solutions[i][0][:len(resources)-1])
            change = change-consumptions[j][1:]

            # Zielzustand zur Ermittlung der Opportunitätskosten wird ermittelt.
            no_reduction = np.append(solutions[i][0][-1], solutions[i][0][-1]-1)
            reject = np.where((conditions == no_reduction).all(axis=1))[0][0]

            if j == 0:
                graph.add_edge(i, reject, key='KA', modus='KA', label='KA', weight=0, weight_goal=solutions[reject][1], revenue=0, goal=solutions[reject][0], time=solutions[i][0][-1])
            else:
                # Keine negativen Kapazitätsbestände möglich.
                if np.all((change) >= np.float32(0)):
                    # Zielzustand wird ermittelt.
                    reduction = np.hstack((change, solutions[i][0][len(resources)-1]-1]))

```



```

        )-1:-1],solutions[i↵
        ][0][-1]-1))
    state = np.where((conditions == ↵
        reduction).all(axis=1))[0][0]
    # Prüfung und Verknüpfung der ↵
    Anfragen
    if probs[j][solutions[i][0][-1]]↵
        > 0:
        # Abfrage, ob die Kante der ↵
        optimalen Politik ↵
        entspricht.
        if solutions[i][5][j]==1:
            graph.add_edge(i, state,↵
                key=str(j)+'-AA', ↵
                modus='AA', label=str↵
                (j)+'-AA', weight=↵
                revenues[j]-solutions↵
                [reject][1]+solutions↵
                [state][1], ↵
                weight_goal=solutions↵
                [state][1], revenue=↵
                revenues[j], goal=↵
                solutions[state][0], ↵
                time=solutions[i↵
                ][0][-1])
        else:
            graph.add_edge(i, state,↵
                key=str(j)+'-AA', ↵
                modus='AA', label=str↵
                (j)+'-AA', style="↵
                dotted", weight=0, ↵
                weight_goal=solutions↵
                [state][1], revenue↵
                =0, goal=solutions[↵
                state][0], time=↵
                solutions[i][0][-1])

    # Lagerveränderung aufgrund der Anfragen↵
    nach Produkt 'j' wird erfasst.
    decrease_stock = np.copy(solutions[i↵
        ][0][(len(resources)-1):-1])
    decrease_stock = decrease_stock-↵
        consumtions[j][1:]
    if np.all((decrease_stock) >= np.float32↵
        (0)):
        # Zielzustand wird ermittelt.

```

```

reduction_stock = np.hstack((↵
    solutions[i][0][: len(↵
        resources) - 1], decrease_stock, ↵
    solutions[i][0][-1] - 1))
state_stock = np.where((↵
    conditions == reduction_stock↵
).all(axis=1))[0][0]
# Prüfung und Verknüpfung der ↵
# Anfragen
if probs[j][solutions[i][0][-1]]↵
    > 0:
    # Abfrage, ob die Kante der ↵
    # optimalen Politik ↵
    # entspricht.
    if solutions[i][5][j]==2:
        graph.add_edge(i, ↵
            state_stock, key=str(↵
                j)+'-LE', modus='LE',↵
            label=str(j)+'-LE', ↵
            weight=revenues[j]-↵
            solutions[reject][1]+↵
            solutions[state_stock↵
            ][1], weight_goal=↵
            solutions[state_stock↵
            ][1], revenue=↵
            revenues[j], goal=↵
            solutions[state_stock↵
            ][0], time=solutions[↵
            i][0][-1])
    else:
        graph.add_edge(i, ↵
            state_stock, key=str(↵
                j)+'-LE', modus='LE',↵
            label=str(j)+'-LE', ↵
            style="dotted", ↵
            weight=0, weight_goal=↵
            =solutions[↵
            state_stock][1], ↵
            revenue=0, goal=↵
            solutions[state_stock↵
            ][0], time=solutions[↵
            i][0][-1])

# Lagererhöhung aufgrund der Anfragen ↵
# nach Produkt 'j' wird erfasst.
change_stock = np.cumsum(np.insert(np.↵

```

```

        array(consumptions[j][1:]), 0, 0)↵
        [: -1])
    increase_stock = np.copy(solutions[i↵
        ][0][(len(resources) - 1): -1])
    increase_stock = increase_stock + ↵
        change_stock
    if np.all((change) >= np.float32(0)) and ↵
        np.all((increase_stock) <= (↵
        max_stocks[1:])) and np.count_nonzero↵
        (change_stock) != 0:
        # Zielzustand wird ermittelt.
        propagation_stock = np.hstack((↵
            change, increase_stock, solutions[i↵
            ][0][ -1] - 1))
        state_storage = np.where((conditions↵
            == propagation_stock).all(axis↵
            = 1))[0][0]
        # Prüfung und Verknüpfung der ↵
        Anfragen
        if probs[j][solutions[i][0][ -1]] > ↵
            0:
            # Abfrage, ob die Kante der ↵
            optimalen Politik entspricht.
            if solutions[i][5][j] == 3:
                graph.add_edge(i, ↵
                    state_storage, key=str(j)↵
                    + '-LP', modus='LP', label↵
                    =str(j) + '-LP', weight=↵
                    solutions[state_storage↵
                    ][1] - solutions[reject↵
                    ][1], weight_goal=↵
                    solutions[state_storage↵
                    ][1], revenue=0, goal=↵
                    solutions[state_storage↵
                    ][0], time=solutions[i↵
                    ][0][ -1])
            else:
                graph.add_edge(i, ↵
                    state_storage, key=str(j)↵
                    + '-LP', modus='LP', label↵
                    =str(j) + '-LP', style="↵
                    dotted", weight=0, ↵
                    weight_goal=solutions[↵
                    state_storage][1], ↵
                    revenue=0, goal=solutions↵
                    [state_storage][0], time=↵

```

```

        solutions[i][0][-1])
# Kante für LP ohne Auftrag.
if probs[0][solutions[i][0][-1]] <
    > 0:
    if solutions[i][5][0]==j:
        graph.add_edge(i, <
            state_storage, key='<
0-LP(' +str(j)+' )', <
            modus='LP', label='0-<
LP(' +str(j)+' )', <
            weight=solutions[<
            state_storage][1] -<
            solutions[reject][1], <
            weight_goal=<
            solutions[<
            state_storage][1], <
            revenue=0, goal=<
            solutions[<
            state_storage][0], <
            time=solutions[i<
            ][0][-1])
    else:
        graph.add_edge(i, <
            state_storage, key='<
0-LP(' +str(j)+' )', <
            modus='LP', label='0-<
LP(' +str(j)+' )', <
            style="dotted", <
            weight=0, weight_goal<
            =solutions[<
            state_storage][1], <
            revenue=0, goal=<
            solutions[<
            state_storage][0], <
            time=solutions[i<
            ][0][-1])

draw = graph.copy()
draw.remove_node('end')
# Quelle: http://stackoverflow.com/a/11484144/4913569
# ———
# write dot file to use with graphviz
# run "dot -Tpng test.dot >test.png"
nx.write_dot(draw, 'graph.dot')
dot = gv.Source(nx.to_agraph(draw), filename='graph.<

```

```

        gv')

        # same layout using matplotlib with no labels
        #pos=nx.graphviz_layout(draw,prog='dot')
        #plt.figure(figsize=(20,20))
        #plt.title("Entscheidungsbaum")
        #nx.draw(draw,pos,with_labels=True, node_size=500)
        #plt.savefig('Graph.png')
    return graph
else:
    print 'Graph hat %i Knoten. Datenstruktur durch NetworkX↵
        ist nicht erstellt.' % len(solutions)

# Optimalen Politik als Datenbank speichern.

def Opt_Politic_Stock(solutions, resources, stocks, products):

    db = pd.DataFrame(index=solutions)

    db_res = []
    db_sto = []
    db_time = []
    db_sol = []
    db_op = []

    #array = np.array(solutions.values()).T
    #print array[0].T

    # Spalten für die Ressourcenkapazitäten erstellen.
    for res in resources[1:]:
        db_res.append('$c^{'+str(res)+'}$')
    db_cap = []
    for c in range(len(db_res)):
        cap = []
        for s in solutions:
            cap.append(solutions[s][0][c])
        db_cap.append(cap)

    # Spalten für die Lagerbestände erstellen.
    for sto in range(len(stocks[1:])):
        db_sto.append('$y^{'+str(sto+1)+'}$')
    db_stos = []
    for i in range(len(db_sto)):
        stos = []
        for s in solutions:

```

```

        stos.append(solutions[s][0][i+len(resources[1:])])
    db_stos.append(stos)

    for s in solutions:
        db_time.append(solutions[s][0][-1])
        db_sol.append(solutions[s][1])

    # Spalten für die optimale Politik.
    for j in range(len(products)):
        db_op.append('$d_{'+str(j)+'}$')
    db_ops = []
    for i in range(len(products)):
        ops = []
        for s in solutions:
            ops.append(solutions[s][5][i]*1)
        db_ops.append(ops)

    for i, res in enumerate(db_res):
        db[res] = db_cap[i]
    for k, sto in enumerate(db_sto):
        db[sto] = db_stos[k]
    db['$t$'] = db_time
    db['ExpValue'] = db_sol
    for i, op in enumerate(db_op):
        db[op] = db_ops[i]

    # Tabelle ausgeben.
    db = db.sort(['$t$'], ascending=0)
    db = db[db['$t$'] != 0]
    db.to_csv('Table_Optimal'+str(datetime.datetime.now())+'.csv',
             index=0)

    return db

```

.2 Szenarien

.2.1 Rechnung

Literaturverzeichnis

- Bach, N.; Brehm, C.; Buchholz, W. und Petry, T. (2012): Wertschöpfung als zentrales Ziel von Unternehmensführung. Gabler Verlag. S. 1–25.
- Barut, M. und Sridharan, V. (2005): Revenue Management in Order-Driven Production Systems. In: Decision Sciences. Bd. 36, Nr. 2, S. 287–316.
- Bellman, R. (1954): The theory of dynamic programming. Arbeitspapier. DTIC Document.
- Cheraghi, S.H.; Dadashzadeh, M. und Venkitachalam, P. (2010): Revenue management in manufacturing: a research landscape. In: Journal of Business & Economics Research (JBER). Bd. 8, Nr. 2.
- Corsten, H. und Stuhlmann, S. (1998): Yield Management: Ein Ansatz zur Kapazitätsplanung und-steuerung in Dienstleistungsunternehmen. In: Schriften zum Produktionsmanagement. Bd. 18.
- Corsten, H. und Stuhlmann, S. (1999): Yield Management als Ansatzpunkt für die Kapazitätsgestaltung von Dienstleistungsunternehmen. In: Wettbewerbsfaktor Dienstleistung: Produktion von Dienstleistungen-Produktion als Dienstleistung, Vahlen, München, S. 79–107.
- Daudel, S. und Vialle, G. (1992): Yield-Management: Erträge optimieren durch nachfrageorientierte Angebotssteuerung. Campus-Verlag.
- Defregger, F. und Kuhn, H. (2007): Revenue management for a make-to-order company with limited inventory capacity. In: . Bd. 29, Nr. 1, S. 137–156.
- DeMiguel, V. und Mishra, N. (2006): What multistage stochastic programming can do for network revenue management. Arbeitspapier. Working paper, London Business School, Optimization Online URL: http://www.optimizationonline.org/DB_HTML/2006/10/1508.html.

- Dickersbach, J. (2004): Production Planning. In: Supply Chain Management with APO. Springer Berlin Heidelberg, S. 131–234.
- Dyckhoff, H. und Spengler, T.S. (2010): Springer-Lehrbuch. Bd. 0. Springer Berlin Heidelberg. S. 3–11.
- Fleischmann, B. und Meyr, H. (2004): Customer Orientation in Advanced Planning Systems. In: H. Dyckhoff; R. Lackes und J. Reese (Hrsg.), Supply Chain Management and Reverse Logistics. Springer Berlin Heidelberg, S. 297–321.
- Friege, C. (1996): Yield-Management. In: WiSt–Wirtschaftswissenschaftliches Studium. Bd. 25, S. 616–622.
- Goetschalckx, M. und Fleischmann, B. (2005): Strategic network planning. In: Supply chain management and advanced planning. Springer, S. 117–137.
- Gönsch, J. und Steinhardt, C. (2013): Using dynamic programming decomposition for revenue management with opaque products. In: BuR-Business Research. Bd. 6, Nr. 1, S. 94–115.
- Guhlich, H.; Fleischmann, M. und Stolletz, R. (2015): Revenue management approach to due date quoting and scheduling in an assemble-to-order production system. In: OR Spectrum, S. 1–32.
- Gutenberg, E. (1959): Enzyklopädie der Rechts- und Staatswissenschaft. Springer Berlin Heidelberg. S. 34–74.
- Gutenberg, E. (1965): Die dispositiven Faktoren. In: Grundlagen der Betriebswirtschaftslehre. Abteilung Staatswissenschaft. Springer Berlin Heidelberg, S. 130–285.
- H. deB. Harris, F. und Pinder, J.P. (1995): A revenue management approach to demand management and order booking in assemble-to-order manufacturing. In: Journal of Operations Management. Bd. 13, Nr. 4, S. 299 – 309.
- Hax, K. (1956): Industriebetrieb. In: Handwörterbuch der Sozialwissenschaften, Stuttgart ua.
- Helbing, K. (2010): Instandhaltung. In: Handbuch Fabrikprojektierung. Springer Berlin Heidelberg, S. 983–989.

- Hinsch, M. (2010): Instandhaltung. In: Industrielles Luftfahrtmanagement. Springer Berlin Heidelberg, S. 189–221.
- Hintsches, A.; Spengler, T.S.; Volling, T.; Wittek, K. und Priegnitz, G. (2010): Revenue management in make-to-order manufacturing: case study of capacity control at ThyssenKrupp VDM. In: BuR-Business Research. Bd. 3, Nr. 2, S. 173–190.
- Kalyan, V. (2002): Dynamic Customer Value Management: Asset Values under Demand Uncertainty using Airline Yield Management and Related Techniques. In: . Bd. 4, Nr. 1, S. 101–119.
- Kimes, S.E. (1989b): Yield management: a tool for capacity-considered service firms. In: Journal of operations management. Bd. 8, Nr. 4, S. 348–363.
- Kimms, A. und Klein, R. (2007): Revenue management. In: . Bd. 29, Nr. 1, S. 1–3.
- Kimms, A. und Klein, R. (2005): Revenue Management im Branchenvergleich. In: Zeitschrift für Betriebswirtschaft / Special Issue. , Nr. 1, S. 1–30.
- Klein, R. (2001): Revenue Management: Quantitative Methoden zur Erlösmaximierung in der Dienstleistungsproduktion. Arbeitspapier. Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL).
- Klein, R. und Steinhardt, C. (2008): Revenue Management : Grundlagen und mathematische Methoden. Springer, Berlin [u.a.].
- Kolisch, R. und Zatta, D. (2006): Revenue-Management in der Sachgüterproduktion. In: Marketing Journal. Bd. 12, S. 38–41.
- Kuhn, H. und Defregger, F. (2004): Revenue Management in der Sachgüterproduktion. In: Wirtschaftswissenschaftliches Studium. Bd. 33, Nr. 5, S. 319–324.
- von Martens, T. (2009): Kundenwertorientiertes Revenue Management im Dienstleistungsbereich. Gabler, Wiesbaden, 1. Aufl.. Aufl.
- Meyr, H.; Wagner, M. und Rohde, J. (2015): Structure of advanced planning

- systems. In: Supply chain management and advanced planning. Springer, S. 99–106.
- Petrick, A. (2009): Multimodale Produkte im Revenue Management : Potenziale und Ansätze zur Realisierung einer Kapazitätssteuerung. Südwestdeutscher Verl. für Hochschulschriften, Saarbrücken.
- Petrick, A.; Gönsch, J.; Steinhardt, C. und Klein, R. (2010): Dynamic control mechanisms for revenue management with flexible products. In: Computers and Operations Research. Bd. 37, Nr. 11, S. 2027–2039.
- Quante, R. (2009): Management of stochastic demand in make-to-stock manufacturing. Bd. 37. Peter Lang.
- Quante, R.; Meyr, H. und Fleischmann, M. (2009): Revenue management and demand fulfillment: matching applications, models, and software. In: OR spectrum. Bd. 31, Nr. 1, S. 31–62.
- Rehkopf, S. (2007): Revenue Management-Konzepte zur Auftragsannahme bei kundenindividueller Produktion: Am Beispiel der Eisen und Stahl erzeugenden Industrie. Springer-Verlag.
- Rehkopf, S. und Spengler, T. (2005): Revenue Management in a Make-to-Order Environment. Bd. 2004. Springer Berlin Heidelberg. S. 470–478.
- Rohde, J. und Wagner, M. (2002): Master Planning. In: H. Stadtler und C. Kilger (Hrsg.), Supply Chain Management and Advanced Planning. Springer Berlin Heidelberg, S. 143–160.
- Ryll, F. und Freund, C. (2010): Grundlagen der Instandhaltung. In: M. Schenk (Hrsg.), Instandhaltung technischer Systeme. Springer Berlin Heidelberg, S. 23–101.
- Schmidt, G. (2012): Grundlagen betriebswirtschaftlicher Prozesse. Springer Berlin Heidelberg. S. 1–29.
- Schmidt, S. (2009): Komplexe Produkte und Systeme. In: Die Diffusion komplexer Produkte und Systeme. Gabler, S. 77–121.

- Specht, D. und Gruß, C.F. (2008): Revenue Management —eine Strategie für die Produktion in der Automobilindustrie?. Gabler. S. 61–72.
- Spengler, T. und Rehkopf, S. (2005): Revenue Management Konzepte zur Entscheidungsunterstützung bei der Annahme von Kundenaufträgen. In: Zeitschrift für Planung und Unternehmenssteuerung. Bd. 16, Nr. 2, S. 123–146.
- Spengler, T.; Rehkopf, S. und Volling, T. (2007): Revenue management in make-to-order manufacturing—an application to the iron and steel industry. In: . Bd. 29, Nr. 1, S. 157–171.
- Stadtler, H. (2008): Purchasing and Material Requirements Planning. In: H. Stadtler und C. Kilger (Hrsg.), Supply Chain Management and Advanced Planning. Springer Berlin Heidelberg, S. 217–229.
- Strunz, M. (2012): Gegenstand, Ziele und Entwicklung betrieblicher Instandhaltung. Springer Berlin Heidelberg. S. 1–35.
- Stuhlmann, S. (2000): Kapazitätsgestaltung in Dienstleistungsunternehmen: eine Analyse aus der Sicht des externen Faktors. Deutscher Universitäts-Verlag.
- Sucky, E. (2009): Revenue Management bei Auftragsfertigung—Ein Ansatz zur Kapazitätssteuerung. In: Logistik Management. Springer, S. 77–96.
- Talluri, K. und van Ryzin, G.J. (2004): Revenue management under a general discrete choice model of consumer behavior. In: Management Science. Bd. 50, Nr. 1, S. 15–33.
- Talluri, K.T. (2001): Airline revenue management with passenger routing control: A new model with solution approaches. In: International Journal of Services Technology and Management. Bd. 2, Nr. 1, S. 102–115.
- Talluri, K.T. und van Ryzin, G.J. (2004b): The Theory and Practice of Revenue Management. International Series in Operations Research and Management Science, vol. 68. Springer.
- Tempelmeier, H. und Günther, H.O. (1994): Produktion und Logistik. Springer.
- Thaler, K. (2001): Supply chain management: Prozessoptimierung in der logistischen Kette. Fortis-Verlag.

- Volling, T.; Akyol, D.E.; Wittek, K. und Spengler, T.S. (2012): A two-stage bid-price control for make-to-order revenue management. In: Computers and Operations Research. Bd. 39, Nr. 5, S. 1021 – 1032.
- Weatherford, L.R. (1998): A tutorial on optimization in the context of perishable-asset revenue management problems for the airline industry. In: Operations Research in the airline industry. Springer, S. 68–100.
- Weatherford, L.R. und Bodily, S.E. (1992): A taxonomy and research overview of perishable-asset revenue management: yield management, overbooking, and pricing. In: Operations Research. Bd. 40, Nr. 5, S. 831–844.
- Zehle, K.O. (1991): Yield-Management–Eine Methode zur Umsatzsteigerung für Unternehmen der Tourismusindustrie. In: Tourismusmanagement und-marketing. Landsberg/Lech: Moderne Industrie, S. S. 483–504.

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Hannover, 11. September 2015

