

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)

Prüfer: Prof. Dr. Stefan Helber

Hannover, den 11. September 2015

Inhaltsverzeichnis

Abkürzungsverzeichnis	iv
Symbolverzeichnis	1
Abbildungsverzeichnis	3
Tabellenverzeichnis	6
Quellcodeverzeichnis	8
1 Einleitung	9
1.1 Problemstellung	9
1.2 Zielsetzung	12
1.3 Aufbau der Arbeit	13
2 Grundlagen zu auftragsbezogenen Instandhaltungsprozessen	16
2.1 Einordnung in die Produktionswirtschaft und Begriffsbestimmung .	16
2.2 Charakteristika	19
2.3 Entscheidungsunterstützungssysteme	22
3 Das Konzept des Revenue Managements	25
3.1 Herkunft des Revenue Managements	25
3.2 Anwendungsvoraussetzungen und Instrumente	27
3.3 Das dynamisch, stochastische Grundmodell	29
4 Bestehende Ansätze zur Annahme von Aufträgen in der Auftragsfertigung und bei Instandhaltungsprozessen	39
5 Eine Modellformulierung für Auftragsannahme- und Lagerhaltungsentscheidungen bei auftragsbezogenen Instandhaltungsprozessen	48
5.1 Modellannahmen zur Berücksichtigung von Lagerhaltungsentscheidungen	48

5.2	Mathematische Formulierung eines Modells zur Entscheidungsunterstützung bei Auftragsannahme- und Lagerhaltungsentscheidungen	51
5.3	Grundlegendes zum Lösen und Implementieren des Auftragsannahmeproblems	61
6	Numerische Untersuchung	66
7	Schlussbemerkung	98
	Anhang	102
	Literaturverzeichnis	120

Abkürzungsverzeichnis

APS	Advanced Planning and Scheduling System
ATO	Assemble-to-Order (engl. Begriff für kundenindividuelle Fertigung mit standardisierten Komponenten)
ATP	Available-to-promise (engl. Begriff für zusicherbarer Bestand)
DP	Dynamische Programmierung
FCFS	First-come, first-served (engl. Begriff für einfache Reihenfolgeannahme bzw. Windhundverfahren)
GE	Geldeinheiten
KEP	Kundenentkopplungspunkt
MTO	Make-to-Order (engl. Begriff für Auftragsfertigung)
MTS	Make-to-Stock (engl. Begriff für Lagerfertigung)
OK	Opportunitätskosten
RM	Revenue Management (engl. Begriff für Ertragsmanagement)
SCM	Supply Chain Management (engl. Begriff für Wertschöpfungslehre)

Symbolverzeichnis

\mathbf{a}_j	Vektor des Ressourcenverbrauchs bzw. der Bestandsveränderung der Produkthanfrage j
a_{hj}	Verbrauch bzw. Bestandsveränderung der Ressource h der Produkthanfrage j
\mathbf{c}	Vektor der Ressourcenkapazität
c_h	Kapazität der Ressource h
$d_0(\mathbf{c}, \mathbf{y}, t)$	Optimale Politik in Abhängigkeit der Kapazitäten \mathbf{c} und des Lagerbestands \mathbf{y} , sofern keine Anfrage zur Periode t eintrifft
d_0	Verkürzte Schreibweise für $d_0(\mathbf{c}, \mathbf{y}, t)$
$\mathbf{d}(\mathbf{c}, \mathbf{y}, t)$	Vektor der optimalen Politik $d_j(\mathbf{c}, \mathbf{y}, t)$
$d_j(\mathbf{c}, \mathbf{y}, t)$	Optimale Politik für Produkthanfrage j in Abhängigkeit der Kapazitäten \mathbf{c} und des Lagerbestands \mathbf{y} zur Periode t
d_j	Verkürzte Schreibweise für $d_j(\mathbf{c}, \mathbf{y}, t)$
h	Ressource aus der Menge \mathcal{H}
\mathcal{H}	Menge an Ressourcen
i	Leistungserstellungszeitpunkt im klassischen Revenue Management
j	Produkthanfrage aus der Menge \mathcal{J}
\mathcal{J}	Menge an möglichen Produkthanfragen
l	Anzahl der gesamten Ressourcen $h \in \mathcal{H}$
n	Anzahl der gesamten Produkthanfragen $j \in \mathcal{J}$
$p_0(t)$	Wahrscheinlichkeit, dass keine Anfrage in Periode t eintrifft
$p_j(t)$	Wahrscheinlichkeit der Nachfrage nach Produkt j in Periode t
r_j	Erlös der Produkthanfrage j
t	Periode des Buchungshorizonts T
\hat{t}	Leistungserstellungszeitpunkt aus der Menge \hat{T}

T	Buchungshorizont
\hat{T}	Menge der Leistungserstellungszeitpunkte
$V(\mathbf{c}, t)$	Ertragsfunktion in Abhängigkeit der Kapazitäten \mathbf{c} in der Periode t
$V(\mathbf{c}, \mathbf{y}, t)$	Ertragsfunktion in Abhängigkeit der Kapazitäten \mathbf{c} und des Lagerbestands \mathbf{y} in der Periode t
\mathbf{y}	Vektor des Lagerbestands an reparierten Produkten
y_h	Lagerbestand eines reparierten Produkts (Produkt ist durch Ressource h substituierbar)
\mathbf{y}^{max}	Vektor des maximal möglichen Lagerbestands an reparierten Produkten
y_h^{max}	Maximal möglicher Lagerbestand eines reparierten Produkts (Produkt ist durch Ressource h substituierbar)
$Y(\mathbf{a}_j, \hat{t})$	Hilfsfunktion zur Ermittlung der Lagerveränderung durch den Parameter der Bestandsveränderung \mathbf{a}_j ab der Leistungserstellung \hat{t}

Abbildungsverzeichnis

1	Grafische Darstellung der möglichen Entscheidungen zur Befriedigung von auftragsbezogenen Instandhaltungsprozessen	11
2	Grafische Darstellung der Gliederung der Arbeit	14
3	Grafische Darstellung eines Wertschöpfungsprozesses sowie der Veranschaulichung von Fertigungskonzepten und Advanced Planning and Scheduling Systemen	18
4	Grafische Darstellung des Buchungshorizonts und der jeweiligen Zeitpunkte der Leistungserstellung in der Auftragsfertigung	29
5	Beispielhafte Darstellung der ersten rekursiven Folge einer Problemstellung im Netzwerk RM	33
6	Darstellung der Systemzustände der beispielhaften Problemstellung im Netzwerk RM	34
7	Darstellung der Systemzustände der beispielhaften Problemstellung im Netzwerk RM (optimaler Auftragseingang)	36
8	Darstellung der Systemzustände der Problemstellung mit konkurrierenden Anfragen im Netzwerk RM	38
9	Zusammenhang von Buchungsperioden und den Leistungserstellungszeitpunkten	50
10	Darstellung der Systemzustände der Problemstellung unter Beachtung der Möglichkeit der Auftragsannahme- und Lagerhaltungsentscheidung	56
11	Rekursive Übergänge der Systemzustände ohne Memofunktion . . .	62

12	Rekursive Übergänge mit Memofunktion	63
13	Wahrscheinlichkeitsverteilung Szenario 1	69
14	Veränderung der optimalen Politik für das Szenario 1 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 10)	70
15	Wahrscheinlichkeitsverteilung Szenario 2	74
16	Veränderung der optimalen Politik für das Szenario 2 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 10)	75
17	Wahrscheinlichkeitsverteilung Szenario 3	76
18	Veränderung der optimalen Politik für das Szenario 3 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 10)	79
19	Wahrscheinlichkeitsverteilung Szenario 4	81
20	Veränderung der optimalen Politik für das Szenario 4 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 10)	82
21	Wahrscheinlichkeitsverteilung Szenario 5	84
22	Veränderung der optimalen Politik für das Szenario 5 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 5)	85
23	Wahrscheinlichkeitsverteilung Szenario 6	87
24	Veränderung der optimalen Politik für das Szenario 6 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 5)	88
25	Wahrscheinlichkeitsverteilung Szenario 7	90

26	Veränderung der optimalen Politik für das Szenario 7 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 5)	91
27	Wahrscheinlichkeitsverteilung Szenario 8	93
28	Veränderung der optimalen Politik für das Szenario 8 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 5)	94
29	Wahrscheinlichkeitsverteilung Szenario 9	95
30	Veränderung der optimalen Politik für das Szenario 9 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 5)	96

Tabellenverzeichnis

1	Ergebnistabelle für die beispielhafte Problemstellung im Netzwerk RM	36
2	Ergebnistabelle für die Problemstellung mit konkurrierenden Anfragen im Netzwerk RM	37
3	Überblick über Publikationen des traditionellen Konzepts des Revenue Managements in der Fertigungsindustrie	40
4	Optimale Politik für die beispielhafte Problemstellung unter Beachtung von Auftragsannahme- und Lagerhaltungsentscheidungen . . .	58
5	Optimale Politik für die zweite beispielhafte Problemstellung unter Beachtung von Auftragsannahme- und Lagerhaltungsentscheidungen	60
6	Überblick der verwendeten Hardware des Clustersystems an der Leibniz Universität Hannover	66
7	Szenario 1	68
8	Auswertung des Szenarios 1	69
9	Szenario 2	73
10	Auswertung des Szenarios 2	74
11	Szenario 3	77
12	Auswertung des Szenarios 3	78
13	Szenario 4	80
14	Auswertung des Szenarios 4	81

15	Szenario 5	83
16	Auswertung des Szenarios 5	84
17	Szenario 6	86
18	Auswertung des Szenarios 6	87
19	Szenario 7	89
20	Auswertung des Szenarios 7	90
21	Szenario 8	92
22	Auswertung des Szenarios 8	93
23	Szenario 9	95
24	Auswertung des Szenarios 9	96

Quellcodeverzeichnis

1	Definition der Parameter für die Dynamische Programmierung im Netzwerk RM	102
2	Python-Algorithmus für die Dynamische Programmierung im Netzwerk RM	108

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 Produkts wirtschaftlich ist. Die Anfragen treffen in beliebiger Reihenfolge ein und haben eine unterschiedliche Wertigkeit. Im weiteren Verlauf dieser Arbeit werden Unternehmen betrachtet, die als auftragsbezogene Dienstleistung das Instandsetzen von Produkten anbieten. Abhängig von dem eingehenden Kundenauftrag, indem der vom Kunden gewünschte Zustand des Produkts beschrieben ist und anhand dem die notwendigen Prozessschritte zur Instandsetzung des Produkts abgeleitet werden, generiert das Unternehmen auftragsbezogene Erträge. Die Prozessschritte zur Instandsetzung des Produkts geben zusätzlich den notwendigen Ressourcenbedarf für die auszuführende Tätigkeit an, die notwendig sind um das Produkt in seinen ursprünglichen bzw. geforderten Zustand zu versetzen. Ressourcen zur Instandsetzung von Produkten können z. B. Material oder Personalstunden sein. Abhängig von dem möglichen Ertrag und dem 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 Eingangsreihenfolge erfolgt, ist die Entscheidungsfindung für das Unternehmen einfach. Der Kundenauftrag wird angenommen, sofern die Kosten des Ressourceneinsatzes niedriger als der erzielte Ertrag ist. Sofern das Unternehmen eine begrenzte Ressourcenkapazität zur Instandsetzung der Güter besitzt, muss zusätzlich der absolute Ressourcenverbrauch des Auftrags für die 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. 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 Produkten bedarf es einer umfassenderen Betrachtung, als nur die kurzfristige Entscheidung über die Annahme einzelner Aufträge. Angenommen ein Unternehmen der Instandhaltung 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 den Betrachtungszeitraum kann es für das Unternehmen sinnvoll sein, Aufträge mit niedrigem Ertrag abzulehnen, sofern im weiteren Verlauf des betrachteten Zeitraums Aufträge mit höherem Ertrag eintreffen. Eine solche Entscheidung erfolgt in Abhängigkeit der noch vorhanden knappen Ressourcenkapazität, die für die unterschiedlichen Kundenaufträge aufgewendet werden müssen.

Eine weitere Alternative für das instandhaltende Unternehmen wäre es, das zu reparierende Produkt mit einem neuwertigen Produkt auszutauschen. Es handelt sich damit um eine Entscheidung der Lagerhaltung bei auftragsbezogenen Instandhaltungsprozessen. Die Entscheidung zur Reduzierung des vorhandenen Lagerbestands zur Befriedung von aktuellen Kundenaufträgen erweist sich als sinnvoll, wenn die zur Produktion notwendigen Ressourcen für spätere Aufträge notwendig sind. Für eine solche Entscheidung zur Befriedung von aktuellen Aufträgen muss ein Lagerbestand der geforderten Produkte aufgebaut werden. Dabei steht im Mittelpunkt einer solchen Politik die optimale Kapazitätsauslastung und eine Ertragsmaximierung.

Auch unter solcher Entscheidungsalternativen kann zwischen dem einfachen Verfahren des FCFS und einem differenzierten Verfahren unter Berücksichtigung von zukünftigen Fertigungsaufträgen unterschieden werden. Die kurzfristige Sichtweise untersucht nur den notwendigen Kapazitätsaufwand und den möglichen Ertrag des aktuellen Auftrags. Bei der langfristigen Sichtweise unter Berücksichtigung von zukünftigen Anfragen werden die vorhandenen Ressourcenkapazitäten und 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 zur Erhöhung des Lagerbestands erfolgen soll. Wie die Bezeichnung der Entscheidung zur Lagerproduktion bereits im Wortlaut deutlich macht, werden die durch den

Ressourcenverbrauch geformten Leistungsbündel (Produkte) auf Lager produziert. Damit ist es dem Unternehmen möglich zu einem späteren Zeitpunkt die Anfragen mittels des Lagerbestands zu befriedigen.

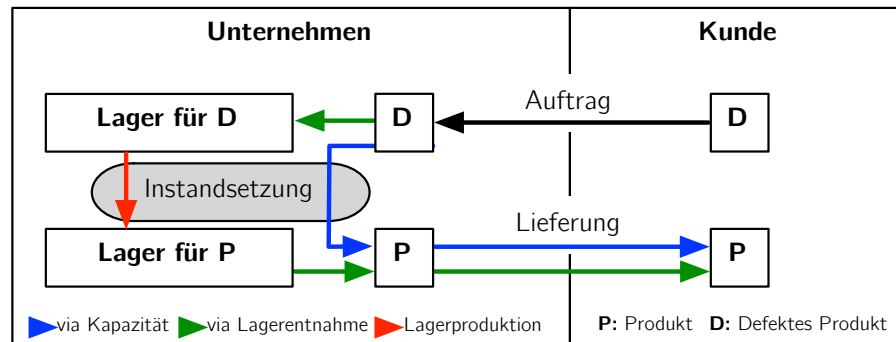


Abbildung 1 Grafische Darstellung der möglichen Entscheidungen zur Befriedigung von auftragsbezogenen Instandhaltungsprozessen

In dieser Arbeit werden die in der Abbildung 1 möglichen Entscheidungen zur Befriedigung von auftragsbezogenen Instandhaltungsprozessen betrachtet. Der Kunde kann innerhalb eines vordefinierten Buchungszeitraums einen Auftrag zur Instandsetzung eines defekten Produkts stellen. Sofern das Unternehmen einen solchen Auftragseingang erhält und sich für die „Auftragsannahme via Kapazität“ entscheidet, wird der Auftrag über die vorhandene Produktions- bzw. Ressourcenkapazität abgewickelt. Alternativ kann das Unternehmen einen solchen Kundenauftrag auch durch einen Austausch des defekten mit einem neuwertigen Produkt befriedigen. D. h. das vom Kunden bereitgestellte defekte Produkt wird in ein Lager für defekte Produkt überführt und das Unternehmen liefert ein neuwertiges Produkt an den Kunden. Das neuwertige Produkt wird dabei aus einem Lager entnommen, dass für solche Produkte bereitgestellt ist. Diese Entscheidung wird als „Auftragsannahme via Lagerentnahme“ bezeichnet. Weiter gibt es die Entscheidung, dass die Ressourcenkapazität für die Instandsetzung von defekten Produkten aus dem unternehmerischen Lager eingesetzt wird. D. h. ein defektes Produkt wird aufgearbeitet und in dem Lager für neuwertige Produkte überführt. Damit ist ein neuer Lagerbestand an einem neuwertigen Produkt vorhanden und kann für nachfolgende Produkthanfragen Verwendung finden. Die Entscheidung wird als „Lagerproduktion“ bezeichnet und kann unabhängig von einem Auftragseingang erfolgen. Sofern jedoch diese Entscheidung aufgrund eines Auftragseingangs einer unrentablen Produkthanfrage getroffen wird, handelt es sich dementsprechend um die Ablehnung der Produkthanfrage. In dieser Arbeit wird angenommen, dass ein ausreichender Lagerbestand an defekten Produkten vorhanden ist, damit sämtliche Ressourcen-

kapazitäten für eine jeweilige Lagerproduktion eingesetzt werden kann. Daher wird der Bestand an defekten Produkten im weiteren Verlauf der Arbeit nicht betrachtet. Damit stehen dem Unternehmen in Summe die Entscheidungen „Auftragsannahme via Kapazität“, „Auftragsannahme via Lagerentnahme“ und „Lagerproduktion“ sowie die generelle „Ablehnung der Anfrage“ zur Verfügung.

1.2 Zielsetzung

In dieser Arbeit wird ein mathematisches Modell für Annahmeentscheidungen von Aufträgen zur Instandsetzung von Produkten dargestellt. Dabei entspricht das durch den Instandhaltungsprozess reparierte Produkt den vom jeweiligen Auftrag geforderten Zustand. Das Modell berücksichtigt dabei die Entscheidungsmöglichkeit über die Auftragsannahme durch die Instandsetzung des Produkts mittels der verfügbaren Ressourcenkapazitäten oder die Befriedigung des Kundenauftrags durch Bereitstellung eines bereits reparierten Produkts. Dabei wird die Entscheidung der Bereitstellung des reparierten Produkts in Abhängigkeit des verfügbaren Lagerbestandes an neuwertigen Produkten getroffen.

Bei der in dieser Arbeit beschriebenen Modellformulierung des Auftragsannahmeproblems bei auftragsbezogenen Instandhaltungsprozessen handelt es sich um ein dynamisches, stochastisches Optimierungsmodell auf Basis des Konzepts des Netzwerk Revenue Managements. Das Optimierungsmodell trifft die Entscheidung, welche Entscheidung die optimale Politik des Auftragsannahmeproblems ist. Dabei ist die optimale Politik die Entscheidung die den möglichen Erwartungswert maximiert. Die Entscheidung erfolgt in Abhängigkeit der Veränderung der verfügbaren Ressourcenkapazitäten nach einer möglichen Instandsetzung des Produkts, des aktuell-vorhandenen Lagerbestandes der bereits reparierten Produkte und der noch potentiell eintreffenden Anfragen im Betrachtungszeitraum.

Das Revenue Management (RM) hat seinen Ursprung in der betrieblichen Problemstellung der Annahme von Aufträgen von freien Sitzplätzen in Passagierflugzeugen.¹ Das Grundmodell wird zur Annahme von Kundenaufträgen bzw. -anfragen von Dienstleistungsunternehmen mit beschränkten Ressourcenkapazitäten verwendet.² Beim RM wird das Auftragsannahmeproblem der zeitlich nacheinander eintreffenden Produkthanfragen als *dynamisch, stochastisches Optimierungsmodell* for-

¹Vgl. Talluri und van Ryzin (2004b), S. 6-10.

²Vgl. Petrick (2009), S. 182-187.

muliert und gelöst. Die Aufgabe des Optimierungsmodells ist es, laut Talluri und van Ryzin (2004b), die Entscheidungsfindung zu unterstützen, damit der Gesamtertrag des Dienstleisters maximiert wird. Mit der Entscheidungsunterstützung ist die optimale Politik der Auftragsannahme für das betrachtete Netzwerk gemeint. Der Gesamtertrag wird in Geldeinheiten (GE) bewertet.

Die Zielsetzung der Arbeit ist demnach das Grundmodell des Netzwerk RM zur Annahme von Kundenaufträgen mit der Möglichkeit der Berücksichtigung von Lagerhaltungsentscheidungen für auftragsbezogene Instandhaltungsprozesse zu erweitern. Zusätzlich zur konzeptionellen Darstellung des hier betrachteten Auftragsannahmeproblems wird für die Modellerweiterung ein Algorithmus entwickelt und implementiert, welcher vorformulierte Beispielszenarien exakt löst. Anschließend wird eine numerische Untersuchung für mehrere Szenarien durchgeführt und die daraus resultierende optimale Politik untersucht. So wird dargestellt, welche Möglichkeiten sich für Unternehmen ergeben, sofern Lagerhaltungsentscheidungen bei auftragsbezogenen Instandhaltungsprozessen berücksichtigt werden.

1.3 Aufbau der Arbeit

Der Aufbau der Arbeit ist in der Grafik 2 dargestellt. In Kapitel 2 wird der Begriff „auftragsbezogener Instandhaltungsprozess“ definiert und die theoretische Grundlagen zu diesem Begriff beschrieben. Der Begriff wird in die Produktionswirtschaft eingeordnet, sowie die Beschreibung der Charakteristika und der Relevanz für betriebliche Entscheidungen von auftragsbezogenen Instandhaltungsprozessen aufgeführt. In Kapitel 3 werden die theoretischen Grundlagen der Arbeit vervollständigt, indem das Konzept des RM zur Annahme von Aufträgen vorgestellt wird. In diesem Kapitel wird auf die Herkunft des Konzepts eingegangen, sowie auf die für das Konzept erforderlichen Anwendungsvoraussetzungen und auf die Instrumente. Des Weiteren ist in dem Kapitel die mathematische Modellformulierung des Grundmodells des Netzwerk RM dargestellt.

Im Anschluss wird in Kapitel 4 ein Literaturüberblick über bestehende Ansätze zur Annahme von Aufträgen der Auftragsfertigung und für Instandhaltungsprozesse aufgeführt. Kapitel 5 zeigt die Modellerweiterung des Netzwerk RM mit der Möglichkeit von Lagerhaltungsentscheidungen. Es werden im ersten Schritt die notwendigen Modellannahmen zur Berücksichtigung von Lagerhaltungsentschei-

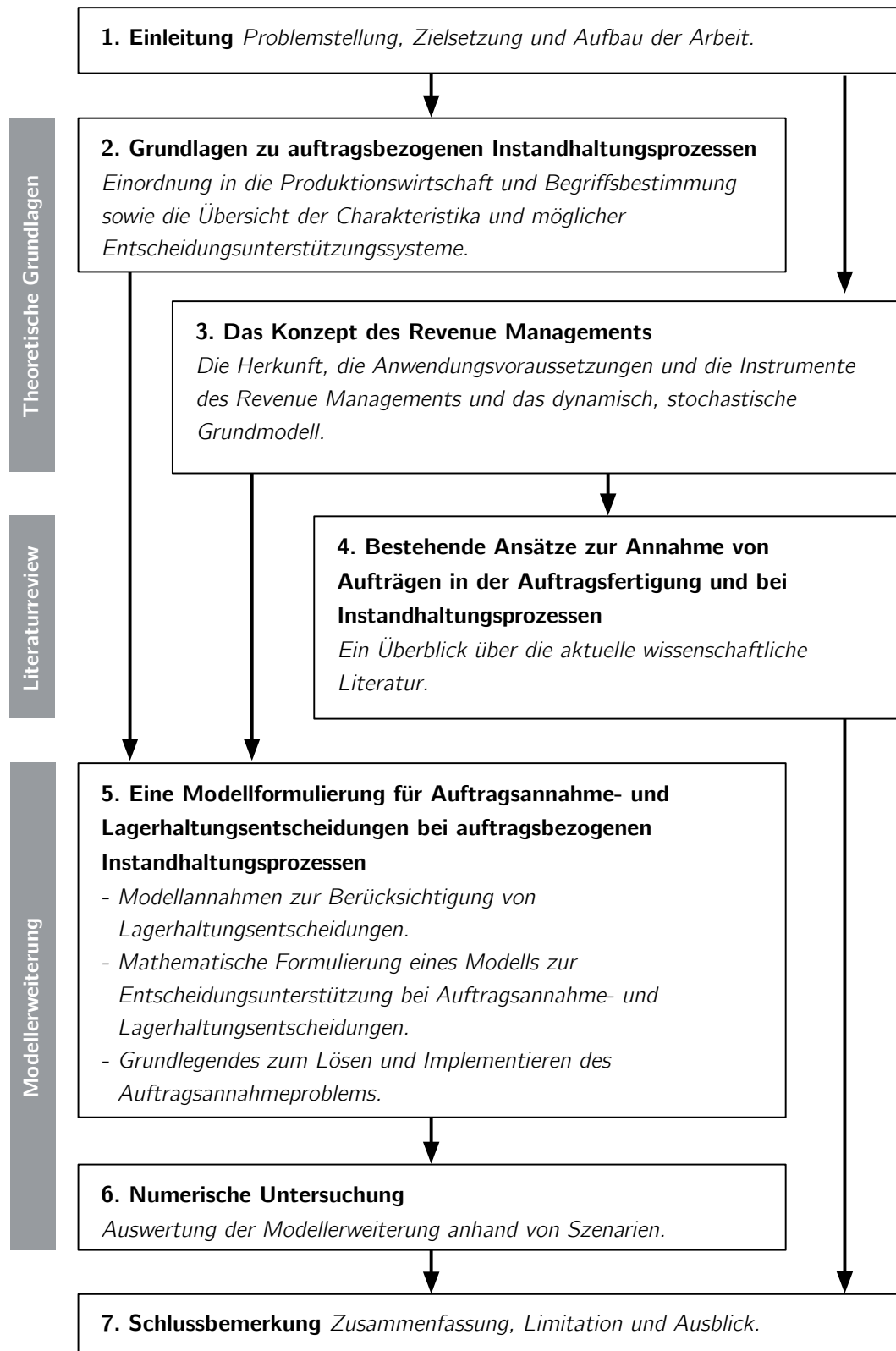


Abbildung 2 Grafische Darstellung der Gliederung der Arbeit

dungen beschrieben und im Anschluss die mathematische Modellformulierung des Modells für Auftragsannahme- und Lagerhaltungsentscheidungen formuliert. Abschließend wird auf das Verfahren und auf den verwendeten Algorithmus zum exakten Lösen des Auftragsannahmeproblems eingegangen. Kapitel 6 zeigt die umfangreiche numerische Untersuchung der vordefinierten Szenarien und die Auswertung der optimalen Politik. Im letzten Kapitel sind die Schlussbemerkungen dieser Arbeit dargestellt. Es handelt sich dabei um eine Zusammenfassung der Ergebnisse der vorliegenden Arbeit, eine Beschreibung der Limitation des Modells und um einen Ausblick für nachfolgende Forschung.

2 Grundlagen zu auftragsbezogenen Instandhaltungsprozessen

2.1 Einordnung in die Produktionswirtschaft und Begriffsbestimmung

Unter Instandhaltungsprozesse (engl. Maintenance-Repair-and-Overhaul) wird in dieser Arbeit 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 sich als eine Leistung eines Unternehmens. Im weiteren wird daher unter Instandhaltung eine (zusätzliche) Dienstleistung eines Unternehmens auf Basis einer Auftragsfertigung definiert. D. h. das Unternehmen nimmt Kundenaufträge zur Instandsetzung von Produkten an und erzielt durch diese Leistung einen auftragsabhängigen Ertrag. Bei der auftragsbezogenen Instandhaltungsprozessen handelt es sich folglich um eine Produktion einer Leistung und kann somit der betriebswirtschaftlichen Betrachtung der Produktionswirtschaft zugeordnet werden.

Bei dem Begriff der Produktionswirtschaft handelt sich um ein Teilgebiet der Betriebswirtschaftslehre.³ In der Produktionswirtschaft wird der Fokus auf die Produktion von Leistung gelegt.⁴ 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

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

⁴Vgl. Dyckhoff und Spengler (2010), S. 3.

sowie um die dispositiven Faktoren Betriebsführung, Organisation und Planung.⁵ Bei Outputgütern handelt es sich um Produkte in Form von Sach- oder Dienstleistungen, die dem Markt und somit der potentiellen Nachfrage der Marktteilnehmer zur Verfügung gestellt werden.⁶ Die Transformation erfolgt durch bestimmte von Menschen veranlasste unternehmerischen Verfahrensweisen.⁷ Beispielsweise kann hier die industrielle Fertigung von Verbrauchs- oder Gebrauchsgütern genannt werden.

Bei der Transformation der Inputgüter erfolgt eine qualitative, quantitative, räumliche oder zeitliche Veränderung der Objekte.⁸ 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)).⁹ Danach sollte das Ziel eines jeden Unternehmens das Betreiben von Wertschöpfung sein.¹⁰ In der klassischen Auffassung der Wertschöpfungslehre durchläuft die Leistungserstellung alle (Teil-)Systeme des Unternehmens.¹¹ Abbildung 3 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: 1. Forschung/Entwicklung, 2. Beschaffung, 3. Produktion (Operations), 4. Distribution sowie 5. Verkauf. Damit ist die um die Wertschöpfung erhöhte Leistung auf dem Markt erstmalig angekommen und das Unternehmen erzielt mit dem Absatz einen Ertrag.

Sofern das Unternehmen eine **Instandhaltung** als Leistung anbietet, wird die Abfolge des Wertschöpfungsprozesses um dieses Unternehmenssystem erweitert. Dabei handelt es sich bei dem System der Instandhaltung i. d. R. um eine wiederholende Wertschöpfung, indem das Unternehmen das zu reparierende Gut durch die angebotene Leistung erneuert. 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.¹² Andererseits kann ein Unternehmen auch die Instand-

⁵Vgl. Schubert und Fricke (2005), S. 324; Weber (1999), S. 284.

⁶Vgl. Schmidt (2012), S. 1.

⁷Vgl. Tempelmeier und Günther (1994), S. 6.

⁸Vgl. Dyckhoff und Spengler (2010), S. 3.

⁹Vgl. Stadtler und Kilger (2005), S. 9-11; Christopher (1998), S. 15; Oliver und Webber (1982), S. 42-47.

¹⁰Vgl. Bach et al. (2012), S. 1.

¹¹Vgl. Werner (2013), S. 5.

¹²Vgl. die Richtlinie 1999/44/EG des Europäischen Parlaments und des Rates vom 25. Mai 1999.

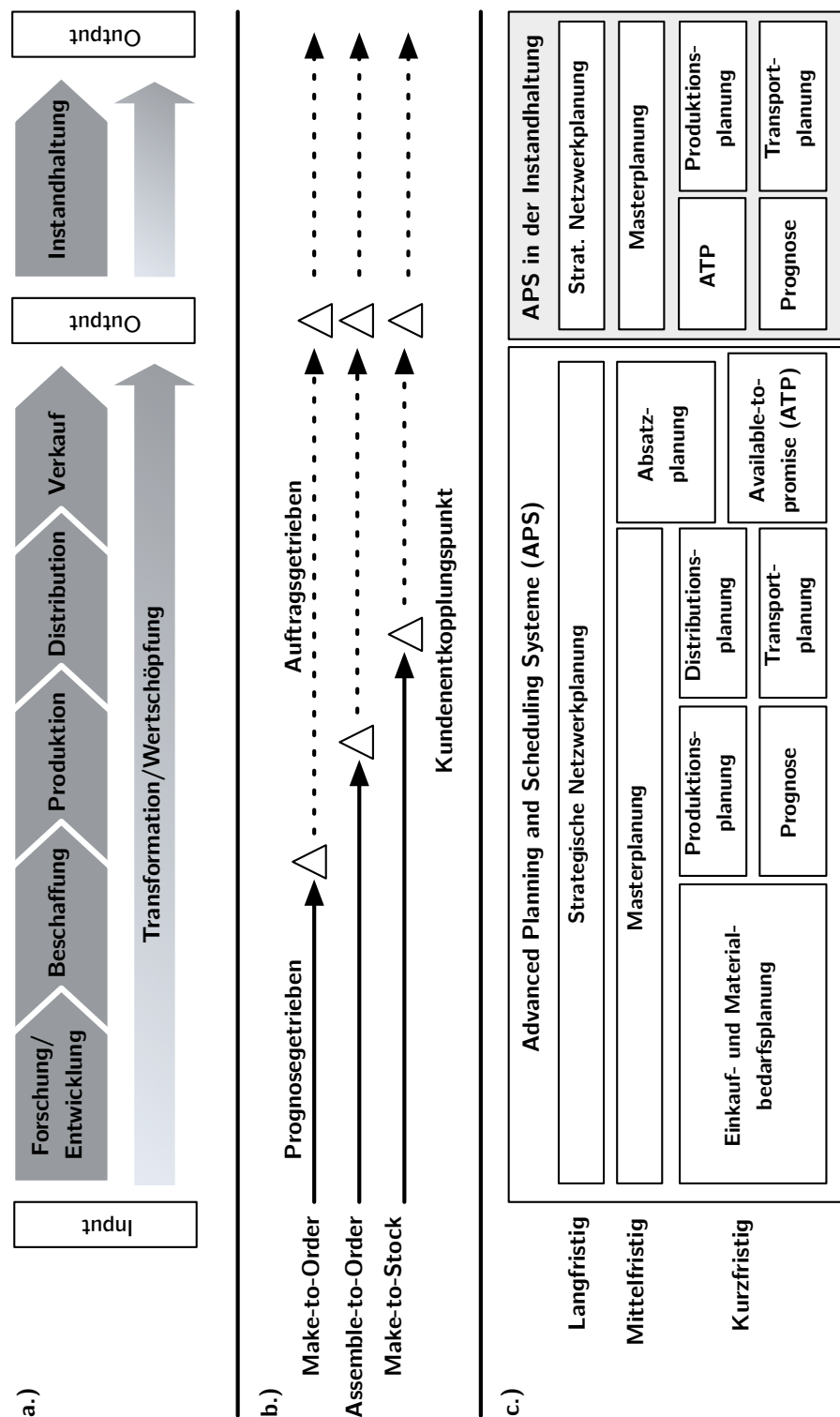


Abbildung 3 Grafische Darstellung eines Wertschöpfungsprozesses sowie der Veranschaulichung von Fertigungskonzepten und Advanced Planning and Scheduling Systemen

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

haltung seiner eigenen Güter als eigenständig angebotene Leistung bereitstellen und damit den Gesamtertrag des Unternehmens erhöhen (Auftragsfertigung). Diese Leistung macht vor allem bei Unternehmen Sinn, die ihren Kunden komplexe Produkte anbieten. Komplexe Produkte zeichnen sich durch ihre hohe Anzahl an hochtechnologischen Komponenten aus, die erst durch den vollständigen und meist kostenintensiven Zusammenschluss eine individuelle Bedürfnisbefriedigung des Nachfragers ermöglichen.¹³ 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 Instandhalten 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.“¹⁴ In dieser Arbeit ist unter dem Begriff der Instandhaltungsprozesse gemeint, dass für den Reparaturauftrag des betrachteten komplexen Produkts ein spezifischer Ausführungsmodus verwendet werden kann, der eine erneute Integration von produkt-spezifischen Ressourcen vorsieht, damit die Funktionsfähigkeit des Produkts wiederhergestellt ist.

Damit lässt sich abschließend das Verständnis von Instandhaltung als Unternehmenssystem ableiten, welches die Verlängerung und zugleich die Erweiterung des Wertschöpfungsprozesses versteht. Abbildung 3 zeigt im Teil a den erweiterten Wertschöpfungsprozess inkl. des Teilsystems der Instandhaltung für ein produzierendes Unternehmen.

2.2 Charakteristika

Nach der DIN 310511 wird Instandhaltung ausgeführt, wenn die Funktionsfähigkeit einer Betrachtungseinheit sichergestellt werden muss, damit der ursprüngliche Wert erhalten bleibt.¹⁵ Betrachtungseinheit können ganze Anlagen und Maschinen sein oder nur einzelne Komponenten.¹⁶

¹³Vgl. Schmidt (2009), S. 97.

¹⁴Vgl. Helbing, S. 984.

¹⁵Vgl. Strunz (2012), S. 1.

¹⁶Vgl. Ryll und Freund (2010), S. 23.

Erläuterung nach DIN 31051:¹⁷

- Instandhaltung ist die Kombination aller technischen und administrativen Maßnahmen des Managements während des Lebenszyklus einer Betrachtungseinheit zur Erhaltung des funktionsfähigen Zustandes oder der Rückführung in diesen, so dass sie die geforderte Funktion erfüllen kann.
- Als Betrachtungseinheit 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.¹⁸ 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. In dieser Arbeit wird bei auftragsbezogenen Instandhaltungsprozessen nur die Instandsetzung von Produkten betrachtet.

Instandhaltungsprozesse lassen sich weiter nach Ausführungszeitpunkt und Ausführungsort der Maßnahmen unterscheiden.¹⁹ Diese Unterscheidung spielt bei der Betrachtung von auftragsbezogenen Instandhaltungsprozessen eine untergeordnete Rolle und wird daher im Verlauf der Arbeit nicht mehr betrachtet.

Da der Absatz zeitlich vor der Produktion der Leistung stattfindet, handelt es sich bei den hier betrachteten auftragsbezogenen Instandhaltungsprozessen um eine Auftragsfertigung.²⁰ In der wissenschaftlichen Literatur wird für die Auftragsfertigung oft der englischen Begriff *Make-to-Order* verwendet (Abk. MTO). 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). Eine Abgrenzung kann anhand des Kundenentkopplungspunkt (KEP) erfolgen, der die Planungsgrundlage der Leistungserstellung bestimmt.²¹ Der KEP zeigt den erstmaligen Kundenkontakt bzw. Auftragseingang

¹⁷Vgl. DIN 31051 (2012).

¹⁸Vgl. Ryll und Freund (2010), S. 23-24.

¹⁹Vgl. Ryll und Freund (2010), S. 24-26; Hinsch (2010), S. 190-191.

²⁰Vgl. Hax (1956), S. 247; Gutenberg (1965), S. 164-165.

²¹Vgl. Quante (2009), S. 21-22; Sharman (1984), S. 71-79.

auf.²² Mit Eingang des Kundenauftrags wechselt die Planungsgrundlage der Leistungserstellung von prognosegetriebener hin zu auftragsgetriebener Planung. Die prognosegetriebene Planungsgrundlage für die Fertigung ist mit einem Prognosefehler für die Ausführung der unternehmerischen Tätigkeit behaftet, was der Betrachtung des KEP weiteres Gewicht verleiht.²³

Bei einem MTO-Auftrag trifft der Kundenauftrag vor der eigentlichen Produktion der Leistung ein. D. h. das Unternehmen startet erst nach Annahme des Auftrags mit der Leistungserstellung. Damit sind die Forschungs/Entwicklung der Leistung und die Beschaffung der Komponenten bzw. der Ressourcen hauptsächlich prognosegetrieben. Alle weiteren Teilsysteme des Unternehmens können sich speziell an den Forderungen des Auftrags richten. Ähnliche Rahmenbedingung besitzt eine ATO, die den KEP innerhalb des Produktionsablaufs hat. Da nur ein gewisser Umfang der Komponenten an auftragsspezifischen Eigenschaften angepasst wird, sind die Grundkomponenten der Produkte abhängig von den Prognosen des Unternehmens über die möglichen Absatznachfrage. Bei MTS erfolgt der Auftragseingang erst nach der Produktion, womit diese Fertigungsart den höchsten Anteil des Einsatzes von betrieblicher Prognosemethoden aufweist, damit ein ausreichendes Absatzvolumen produziert wird.²⁴ Abbildung 3 zeigt im Teil b die Unterschiede der Fertigungsarten im Verlauf des Wertschöpfungsprozesses.

Da bei auftragsbezogenen Instandhaltungsprozessen erst mit Auftragseingang die erforderlichen Produktionsschritte und der Ressourcenbedarf bekannt sind, können diese mit MTO-Prozessen gleichgesetzt werden. 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 damit ein zu geringer Bestand an Ressourcen vorhanden ist, konkurrieren die unterschiedlich eintreffenden Anfragen nach der Instandsetzungsdienstleistung 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. Unrentable Anfragen sollten abgelehnt werden, damit die nicht verbrauchten Ressourcen für spätere rentablere Anfragen Verwendung finden. Im nächsten Abschnitt wird dieser Frage in Bezug der betrieblichen Entscheidungsfindung weiter nachgegangen.

²²Vgl. Olhager (2010), S. 863.

²³Vgl. Quante (2009), S. 21.

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

²⁵Vgl. Thaler (2001), S. 68.

2.3 Entscheidungsunterstützungssysteme

Die Entscheidung über die mögliche Ausgestaltung der Politik bzgl. der Annahme oder Ablehnung von eingehenden Kundenaufträgen besitzt einen großen Einfluss auf den Gesamtertrag und auf die Effizienz der gesamten Wertschöpfungskette eines Unternehmens.²⁶ Ein effizienter Prozess der Wertschöpfung kann der nachhaltigen Konsumentenzufriedenheit dienen.²⁷ Wichtiger Erfolgsfaktor für ein erfolgreiches SCM und somit der Konsumentenzufriedenheit ist der Einsatz von Advanced Planning and Scheduling Systemen (APS).²⁸

Bei APS handelt es sich um computergestützte Systeme, mit denen die Planung der ganzheitlichen Wertschöpfung der Leistung mit Hilfe von mathematischen Modellen des Operations Research unterstützt wird.²⁹ Neben der reinen Planung der Produktion, helfen moderne APS mit ihrem modularen Aufbau auch bei der gesamten Auftragsabwicklung und unterstützen damit Unternehmen bei betrieblichen Entscheidungen innerhalb des SCM-Prozesses, wie z. B. bei dem Einkauf von Ressourcen, der Produktionsplanung und dem Absatz der Leistung.³⁰

Meyr et al. (2015) sortieren einige wichtige Module eines APS in den Dimensionen des SCM-Prozesses und des Planungshorizonts. Abbildung 3 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 erheblichen Einfluss auf die langfristige Rentabilität sowie auf die Wettbewerbsposition des Unternehmens und haben räumlichen sowie zeitlichen Charakter.³¹ Typische Entscheidungen, die aufgrund des Moduls der strategischen Netzwerkplanung für ein Unternehmen getroffen werden, sind die Festlegungen des Kunden- und Produktprofils über einen längeren Betrachtungszeitraum. Aber auch grundlegende Entscheidungen bzgl. des Herstellungsprozesses des Unternehmens werden aufgrund dessen abgeleitet. Datengrundlage für diese Planung bilden u. a. Prognosen, Marktforschungen und Wirtschaftstrends. Die Absatzplanung dient einerseits der Absatzprognose und andererseits der Analyse der notwendigen Sicherheitsbestände.²⁹ Das Modul Masterplanung synchronisiert den

²⁶Vgl. Talluri und van Ryzin (2004b), S. 2.

²⁷Vgl. Heikkilä (2002), S. 752.

²⁸Vgl. Fleischmann und Meyr (2004), S. 298.

²⁹Vgl. Fleischmann und Meyr (2004), S. 298.

³⁰Vgl. Meyr et al. (2015), S. 99-100; Fleischmann und Meyr (2004), S. 298.

³¹Vgl. Goetschalckx und Fleischmann (2005), S. 117-118.

Materialfluss entlang des gesamten SCM-Prozesses und unterstützt dadurch mittel-fristige 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 für die Fertigung kurzfristig terminierte Pläne für Komponenten, Teile und Ressourcen 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.²⁹ 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 Pläne für die Verkaufsstellen und für die Logistik generiert.²⁹ Zum Abschluss ist in der Übersicht von Meyr et al. (2015) das Modul Available-to-promise (ATP) aufgeführt. Bei ATP handelt es sich um ein System, mit dem Kundenaufträge auf Realisierbarkeit in Bezug der aktuellen und zukünftigen Verfügbarkeit von Angebot sowie Kapazität analysiert werden.³⁵ Das Modul dient hauptsächlich der Kundenzufriedenheit, da es die eintreffenden Kundenaufträge prüft und ggf. die geforderte Lieferung der Leistung zusagt.²⁹

Die vorgestellten Module sind von Meyr et al. (2015) jedoch ins Verhältnis mit einem typischen SCM-Prozess gesetzt. Es stellt sich die Frage, welche Systeme für betriebliche Entscheidungen durch Einbeziehung eines Instandhaltungsprozesses in den SCM-Prozess relevant sind. Die Instandhaltung verlängert den Wertschöpfungsprozess des Unternehmens und benötigt ebenfalls den Einsatz von speziellen APS-Systemen. Es kann eine Teilmenge der Systeme nach Meyr et al. (2015) auch auf Instandhaltungsprozesse übertragen werden, da diese der Auftragsfertigung gleichgesetzt werden. Es handelt sich um die Systeme die sich aufgrund der Produktion einer Leistung ergeben, aber auch um ein System in Anlehnung an das System ATP. Für auftragsbezogene Instandhaltungsprozesse bedarf es einer langfristigen Entscheidungsunterstützung in Bezug der strategischen Netzwerkplanung. Mit dem Modell lässt sich ableiten, welche Ausgestaltung die Leistung der Instandhaltung erhält. Die Masterplanung, die für mittelfristige Entscheidungen notwendig ist, unterstützt den Ressourceneinsatz über einen gewissen vordefinierten Planungshorizont und ist damit ebenfalls für die mögliche Ressourcenverwen-

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

³³Vgl. Stadtler (2008), S. 217-218.

³⁴Vgl. Dickersbach (2004), S. 131.

³⁵Vgl. Stadtler et al. (2008), S. 183.

dung bei Instandhaltungsaufträgen von Nöten. Für systemgestützte Entscheidungen zur Instandhaltung bedarf es für die kurzfristige Betrachtungsweise an Input der Einkaufs- und Materialplanung. Bei diesen Informationen kann es sich z. B. um den aktuellen Lagerbestand oder den Zeitpunkt von Lieferung neuer Produktionsrohstoffe handeln. Bei auftragsbezogenen Instandhaltungsprozessen wird unterstellt, dass eine Entscheidung bzgl. der Auftragsannahme auf Basis des Systems ATP im Vorfeld getätigt werden muss. D. h. die Entscheidungsgrundlage über die Annahme oder Ablehnung eines Auftrags muss im Vorfeld der Produktionsplanung vorliegen. Ebenso sind Prognosen über mögliche weitere Auftragseingänge zur diesem Zeitpunkt und für die nachfolgenden Entscheidungen zu beachten. Erst mit Informationen über den aktuellen Bestand an Ressourcen und deren Verbrauch sowie einer Prognose über die weitere Auftragsentwicklung ist die Annahme von Aufträgen möglich. Damit wäre eine Beachtung des Auftrags in der Produktions- und Transportplanung möglich.

Für auftragsbezogene Instandhaltungsprozesse bedarf es daher einer speziellen Planung in Form einer optimalen Politik der effizienten Nutzung der Ressourcen. Ein Unternehmen benötigt für die kurzfristige Entscheidung bzgl. der Annahme oder Ablehnung von Instandhaltungsaufträgen Informationen einer umfangreichen Planung auf operativer Ebene. Dabei kann die Erstellung einer solchen Planung für MTO- bzw. auftragsbezogene Instandhaltungsprozesse anhand des Konzepts des Netzwerk RM erfolgen. Dabei liefert diese für jede noch verfügbare Ressourcenkapazität und für jeden Zeitpunkt des Planungshorizonts eine optimale Politik der Akzeptanz von Aufträgen. Ziel ist es, nach Petrick (2009), die Glättung der Nachfrage herbeizuführen, damit alle Ressourcen gleichmäßig und möglichst vollständig ausgelastet sind sowie eine Maximierung des Gesamterlöses herbeizuführen. Eine solche Planung erfolgt unter Beachtung des angebotenen Leistungsprogramms sowie der Prognose über den möglichen weiteren Auftragsverlauf und kann dem Modul ATP zusortiert werden. Im weiteren Verlauf dieser Arbeit wird das Grundmodell des Netzwerk RM vorgestellt (Kapitel 3) und ein Modell speziell für auftragsbezogenen Instandhaltungsprozesse unter Beachtung von Lagerhaltungsentscheidungen aufgestellt (Kapitel 5). Ein solches Modell des Operations Research kann einem modernen APS-System bzw. dem ATP-Modul hinterlegt werden. Mit einer Planung aufgrund der Anwendung des Konzepts des RM ist eine verbesserte Kapazitätsauslastung für das Unternehmen möglich.³⁶

³⁶Vgl. Petrick (2009), S. 58-61.

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.³⁷ 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.³⁸

Der Begriff *Revenue Management* wird im deutschsprachigen Raum meist mit *Ertragsmanagement* oder *Erlösmanagement* übersetzt.³⁹ Yield Management wird als Synonym benutzt.⁴⁰ Dabei greift der Begriff *Yield* zu kurz, da damit in der Luftverkehrsbranche der Erlös je Passagier und geflogener Meile bezeichnet wird.⁴¹ Die Bezeichnung *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.⁴² Erste Ansätze des Konzepts 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 Revenue Management (RM).⁴³

In der Literatur wird der Begriff des RM unterschiedlich definiert. Friege (1996) bezeichnet das RM als *Preis-Mengen-Steuerung*, Daudel und Vialle (1992) als

³⁷Vgl. Klein (2001), S. 246.

³⁸Vgl. Spengler und Rehkopf (2005), S. 124.

³⁹Vgl. z. B. Zehle (1991), S. 486.

⁴⁰Vgl. z. B. Kolisch und Zatta (2006), S. 319.

⁴¹Vgl. z. B. Weatherford (1998), S. 69.

⁴²Vgl. Klein und Steinhardt (2008), S. 6; Spengler und Rehkopf (2005), S. 124-125.

⁴³Vgl. Petrick (2009), S. 1-3.

Preis-Kapazitäts-Steuerung und Talluri und van Ryzin (2004b) verstehen es als das gesamte *Management der Nachfrage*. Die beiden ersteren Definitionen können als Synonym für eines der Instrumente des RM stehen und daher finden diese für das gesamte Konzept keine weitere Verwendung.⁴⁴ 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 eines Unternehmens die Gesamterlöse zu maximieren, die sich aufgrund der speziellen Anwendungsgebiete ergeben. Weiter beschreibt 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 RM 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 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**, damit verdeutlicht dargestellt wird, in welchen Branchen das RM Potentiale liefert. Dabei sollten branchenspezifische Besonderheiten, neben den zahlreichen Ähnlichkeiten Berücksichtigung finden, sowie das begrenzte Kapazitätskontingent, damit die Potentiale

⁴⁴Vgl. Petrick (2009), S. 9-13.

des RM zur Maximierung der Gesamterlöse in den Dienstleistungsbranchen erfolgen kann.⁴⁵ In dem nachfolgenden Kapitel werden die Anwendungsvoraussetzungen sowie Instrumente des RM vorgestellt.

3.2 Anwendungsvoraussetzungen und Instrumente

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:⁴⁶

- „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 und
- Möglichkeit zur Marktsegmentierung und im Ergebnis dessen zur segmentorientierten Preisdifferenzierung.

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 ihre 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,

⁴⁵Vgl. z. B. von Martens (2009), S. 11-24.

⁴⁶Vgl. u. a. Friege (1996), S. 616-622, und Weatherford und Bodily (1992), 831-832.

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.⁴⁷ Zu den Instrumenten zählen die **segmentorientierte Preisdifferenzierung**, die **Kapazitätensteuerung** und die **Überbuchungssteuerung**. Die Instrumente weisen dabei unterschiedliche Abhängigkeiten untereinander auf, da z. B. die Kapazitätensteuerung auf den Ergebnissen der Preisdifferenzierung aufbaut und die Überbuchungssteuerung selten ohne Kapazitätensteuerung gelöst werden kann. Eine genauere Betrachtung der Instrumente liefern Talluri und van Ryzin (2004b) oder Petrick (2009) in ihren jeweiligen Arbeiten.

Aufgrund der Anwendungsvoraussetzungen des RM kann das Konzept auch auf die Auftragsfertigung bzw. MTO-Prozesse übertragen werden.⁴⁸ 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 erneuerbare Kapazitäten handelt (Arbeitskraft, Maschinenkapazität, usw.). Anders formuliert, die Ressourcen werden zur nachfolgenden Leistungserstellung erneuert und sofern sie in dem Betrachtungszeitraum keine Verwendung fanden, verfallen die Kapazitäten der Ressourcen zum nächsten Betrachtungszeitraum. Ein Betrachtungszeitraum versteht sich im Konzept des RM einen sogenannten Buchungshorizont T , indem die Anfragen bis zum Zeitpunkt der Erstellung der Leistung eintreffen können. Dabei beschreibt t einen Zeitpunkt des Buchungshorizonts. Die Gesamtlänge des Buchungshorizonts entspricht dem Parameter T und verläuft rückwärts in der Zählweise bis zur Leistungserstellung. Der Buchungshorizont im klassischen RM ist mit genau einer Leistungserstellung i gekoppelt. Mit der Leistungserstellung i werden die durch die Auftragsannahme reservierten Kapazitäten des Buchungshorizonts durch die Produktion beansprucht. Mit dem Start der Leistungserstellung erfolgt ebenfalls die Vorausbuchungszeit T für die nachfolgende Leistungserstellung $i + 1$. Mit der Leistungserstellung $i + 1$ sind die Kapazitäten der Ressourcen regeneriert. Abbildung 5 zeigt den Zusammenhang von Buchungshorizont, Leistungserstellung und abnehmender Ressourcenkapazität.

⁴⁷Die Prognose zählt laut Petrick (2009) nicht als eigenständiges Instrument des RM.

⁴⁸Vgl. Hintsches et al. (2010), S.176-178; Kimes (1989b), S. 349-351.

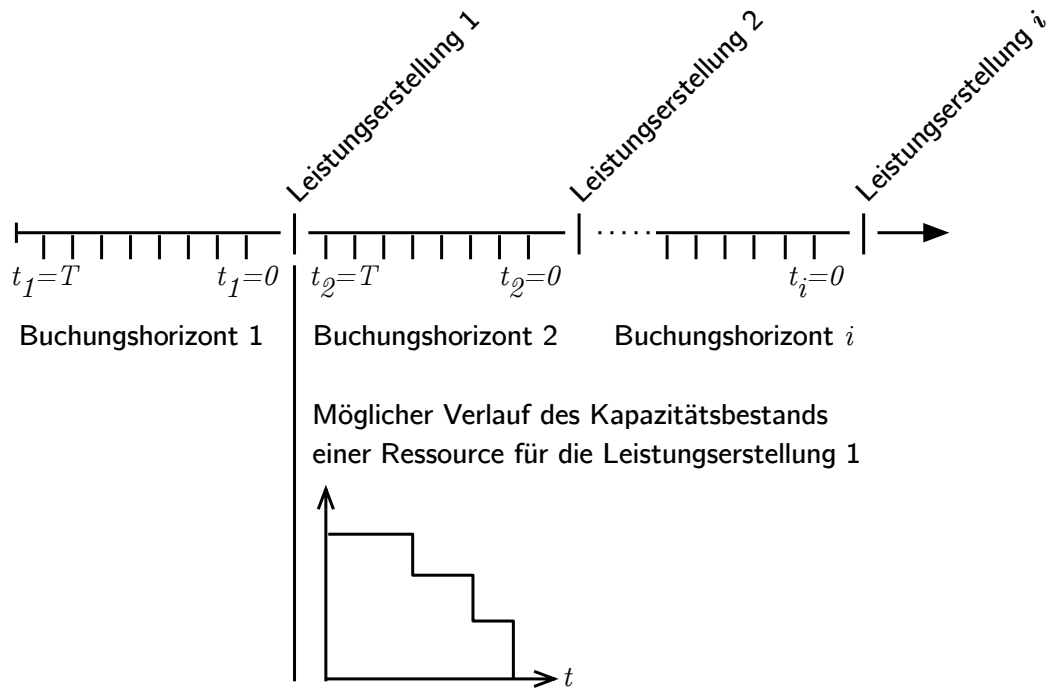


Abbildung 4 Grafische Darstellung des Buchungshorizonts und der jeweiligen Zeitpunkte der Leistungserstellung in der Auftragsfertigung
In Anlehnung an: Helber und Heinze (2014).

3.3 Das dynamisch, stochastische Grundmodell

Im Folgenden wird das dynamisch, stochastische Grundmodell des RM nach Talluri und van Ryzin (2004) 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 als Vektor $\mathbf{c} = (c_1, \dots, c_h, \dots, c_l)$ formuliert. Eine Anfrage nach einem Produkt (Dienstleistung) im 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 Vektor-

schreibweise kann der Ressourcenverbrauch für eine Produkthanfrage j als Vektor $\mathbf{a}_j = (a_{1j}, \dots, a_{hj}, \dots, a_{lj})$ formuliert werden. Der Buchungshorizont entspricht T Perioden und die Aufteilung kann jeweils in einzelne Perioden $t = 1, \dots, T$ erfolgen. 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.⁴⁹

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:⁵⁰

$$V(\mathbf{c}, t) = \sum_{j \in \mathcal{J}} p_j(t) \max \left[\underbrace{V(\mathbf{c}, t-1)}_{\text{Ablehnung}}, \underbrace{r_j + V(\mathbf{c} - \mathbf{a}_j, t-1)}_{\text{Auftragsannahme}} \right] + p_0(t) \underbrace{V(\mathbf{c}, t-1)}_{\text{Kein Auftrag}} \quad (1)$$

Es handelt sich hier um das stochastisch, dynamische Grundmodell im Netzwerk RM. Das Konzept dieser Modellformulierung ist von Bellman entwickelt und dient der Ermittlung der optimalen Politik in Bezug auf den aktuellen Zustand eines Systems.⁵¹ 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 als sogenannte *Bellman'sche Funktionsgleichung* oder *Bellman-Gleichung* bezeichnet.

Der Erwartungswert $V(\mathbf{c}, t)$ mit einer Kapazität \mathbf{c} zur Buchungsperiode t ist die Summe aller Entscheidung über die Produkthanfragen \mathcal{J} , ob eine Produkthanfrage j abgelehnt ($V(\mathbf{c}, t-1)$) oder akzeptiert wird ($r_j + V(\mathbf{c} - \mathbf{a}_j, t-1)$). Sofern keine Akzeptanz der Produkthanfrage j erfolgt, verstreicht eine Buchungsperiode ohne Inanspruchnahme der Kapazität. Wird jedoch die Produkthanfrage j akzeptiert, ist die Kapazität \mathbf{c} durch den Ressourcenverbrauch \mathbf{a}_j zur nächsten Buchungsperiode t reduziert und der jeweilige Ertrag r_j aus der Produkthanfrage j erzielt. Die Entscheidung über Ablehnung oder Annahme einer Produkthanfrage j erfolgt anhand des Erwartungswerts für nachfolgende Perioden $t \in T$. Siehe in Gleichung (1) die Maximierungsfunktion. Dabei ist die Entscheidung mit der jeweiligen Eintrittswahrscheinlichkeit $p_j(t)$ der Produkthanfrage j zur Periode t gewichtet. Zusätzlich

⁴⁹Vgl. Talluri und van Ryzin, S. 18.

⁵⁰Vgl. Petrick (2009), S. 185-187; Bertsimas und Popescu (2003), S. 261.

⁵¹Vgl. Bellman (1954), S. 4-5.

ist der Erwartungswert $V(\mathbf{c}, t)$ um die Wahrscheinlichkeit des Nichteintreffens einer Produkthanfrage j erhöht ($p_0(t)V(\mathbf{c}, t-1)$). Der Erwartungswert $V(\mathbf{c}, t)$ weist dabei die Grenzbedingungen

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

$$V(\mathbf{c}, t) = -\infty, \text{ wenn } c_h < 0 \forall h \in \mathcal{H}, \quad (3)$$

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 vereinfacht werden zur nachfolgenden Gleichung:⁵²

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

Eine eintreffende Produkthanfrage j ist demnach dann akzeptiert, wenn der Ertrag r_j größer gleich der Differenz des Erwartungswertes unter der Prämisse der Ablehnung und dem Erwartungswert unter der Prämisse der Annahme einer Produkthanfrage j ist:

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

Sofern der Ertrag r_j diese Bedingung erfüllt, handelt es sich bei der Annahme der Produkthanfrage j um die optimale Politik. D. h. sofern diese Anfrage eintrifft, sollte diese unter Berücksichtigung der aktuellen Kapazität \mathbf{c} zur Periode t akzeptiert werden. Dabei kann der rechte Term der Bedingung (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 Produkthanfrage $j \in \mathcal{J}$ ausschließlich nur dann,

⁵²Vgl. Spengler et al. (2007), S. 161.

sofern die OK des Ressourcenverbrauchs niedriger als der Ertrag r_j ist. Der maximal mögliche Erwartungswert unter Beachtung der Kapazität \mathbf{c} zum Zeitpunkt t ist damit die Addition des Erwartungswerts unter der Prämisse der Ablehnung der Anfragen und die Summe der um die jeweiligen OK reduzierten Erträge r_j der akzeptierten Produkthanfragen $j \in \mathcal{J}$. Mathematisch lässt sich dies 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 möglichen Produkthanfragen $j \in \mathcal{J}$ 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 der Produkthanfrage $j = 1$ wird die Ressource $h = 1$ mit einer Einheit und zur Ausführung der Produkthanfrage $j = 2$ wird wiederum eine Einheit der Ressource $h = 2$ benötigt. Damit gilt $a_{11} = 1$ und $a_{22} = 1$. Durch Annahme einer Anfrage $j = 1$ wird der Ertrag $r_1 = 100$ und durch Annahme von Anfrage $j = 2$ der Ertrag $r_2 = 200$ generiert. Der Buchungshorizont entspricht $T = 4$. Die Wahrscheinlichkeiten des Eintreffens einer Produkthanfrage $j = 1$ 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, lässt sich mit $p_0(t) = 1 - \sum_{j \in \mathcal{J}} p_j(t)$ berechnen und bilden den Vektor $p_0(t) = (0.4, 0.4, 0.4, 0.4)$.

Die Parameter lassen sich damit abschließend wie folgt definieren:

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

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

Das stochastisch, dynamische Optimierungsproblem aus Gleichung (4) lässt sich als Graph darstellen, indem die einzelnen Erwartungswerte als Systemzustände des Netzwerks aufgeführt sind.⁵³ 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

⁵³Vgl. 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 5 zeigt die erste rekursive Folge für die Gleichung (4) als gerichteten Graphen und im nachfolgenden wird auf die Notation eingegangen.

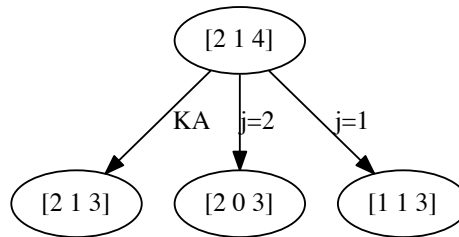


Abbildung 5 Beispielhafte Darstellung der ersten rekursiven Folge einer Problemstellung im Netzwerk RM

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 der *Bellman'schen Funktionsgleichung*. Bei der Benennung eines solchen Knotens wird eine Zahlenfolge verwendet, bei dem 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 Anfragen j und durch Betrachtung 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. Sofern keine Produkthanfrage j eintrifft, wird der nachfolgende Systemzustand $[2 \ 1 \ 3]$ erreicht. Alternativ können Produkthanfragen $j = 1$ oder $j = 2$ eintreffen und dementsprechend werden die vorhandenen Kapazitäten \mathbf{c} um den Ressourcenverbrauch $a_{11} = 1$ bzw. $a_{22} = 1$ reduziert. Daraus folgt, dass der Systemzustand $[1 \ 1 \ 3]$ bzw. $[2 \ 0 \ 3]$ im Netzwerk erreichbar ist. Die optimale Politik eines solchen Graphen wird abgebildet durch die Kanten. Die optimale Politik einer Produkthanfrage j wird im Graphen als durchgezogener Pfeil dargestellt, sofern die Entscheidung der Annahme der Produkthanfrage den Erwartungswert maximiert. Erfolgt keine Maximierung des Erwartungswerts durch die Auftragsannahme der

Produktanfrage j , wird der Übergang in Form der Kante als gestrichelter Pfeil dargestellt. Ein Übergang über die Kante zum nächsten Systemzustand wäre theoretisch möglich, aber entspricht nicht der optimalen Politik. Sofern eine solche unrentable Produkthanfrage j eintrifft, sollte unter der Zielsetzung der Maximierung des Gesamtertrags die Ablehnung der Anfrage erfolgen.

Aufbauend 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 6 zeigt die möglichen Systemzustände für das eingeführte Beispiel.

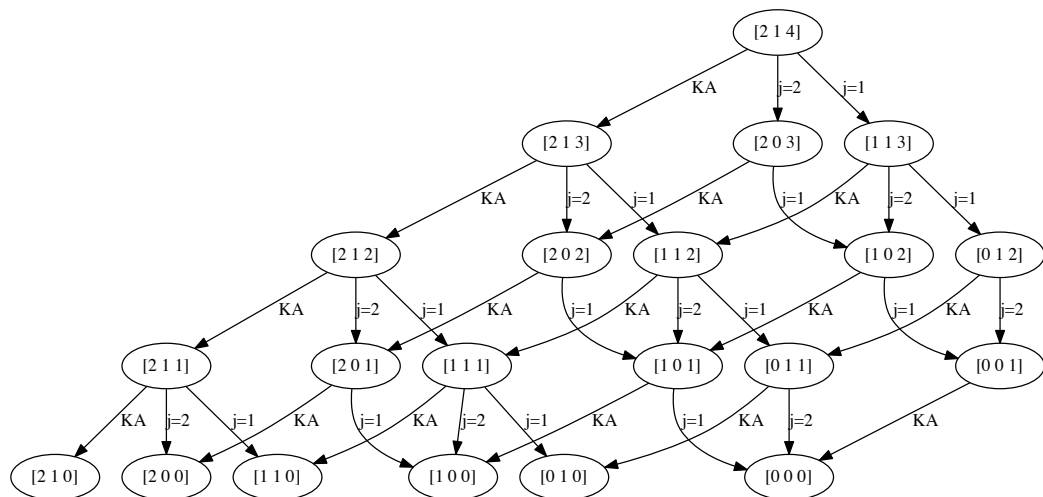


Abbildung 6 Darstellung der Systemzustände der beispielhaften Problemstellung im Netzwerk RM

Anhand dieses Graphen mit den möglichen Systemzuständen aus der Abbildung 6 wird ersichtlich, welche Erwartungswerte (Teilprobleme) für das Entscheidungsproblem der Auftragsannahme durch Rückwärtsinduktion gelöst werden müssen.⁵⁴ 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

⁵⁴Vgl. Puterman (2009), S. 92-93.

muss, dass nicht für alle Systemzustände mit $t = 1$ alle Produktanfragen möglich sind. Dies folgt aus der Grenzbedingung aus Gleichung (3). Zur Veranschaulichung wird der Erwartungswert des Systemzustands $[1 \ 0 \ 1]$ mittels der Gleichung (4) 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 Erwartungswert (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 der Berücksichtigung der vorhandenen Kapazitäten c und den 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 (7) für jeden Systemzustand ermitteln. Für jede Kante des Graphen in Abbildung 6 kann damit der Ertrag abzgl. der OK hinterlegt werden ($r_j - OC_j$). Die optimale Politik für eine Produktanfrage j im betrachteten Graphen der Abbildung 6 ist demnach die Kante bei dem der Ertrag abzgl. der OK den Erwartungswert maximiert. Abbildung 7 zeigt alle Systemzustände und Übergänge als Graphen mit den jeweiligen Erwartungswerten als Knoten und den Wert $r_j - OC_j$ als Kante.

Bei einem Fallbeispiel mit solchen Parameterwerten ist damit nur eine Auswertung des Erwartungswertes über den gesamten Verlauf des Buchungshorizonts T möglich. Dies folgt aus der Tatsache, dass keine Anfragen zu einem beliebigen Systemzustand abgelehnt werden. Bei einem derartigen Ergebnis können jedoch strategische Handlungsempfehlungen abgeleitet werden. Z. B. kann die Anpassung der Instrumente der Marktbearbeitung vom betrachteten Unternehmen erfolgen, damit der beste Pfad der möglichen Auftragseingänge abgearbeitet wird. Der bes-

Tabelle 1 Ergebnistabelle für die beispielhafte Problemstellung im 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

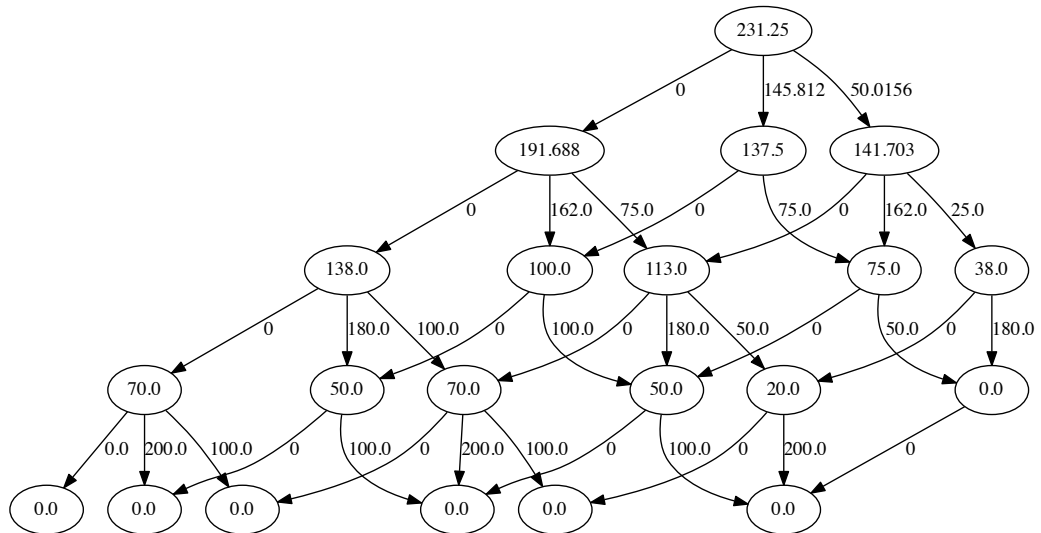


Abbildung 7 Darstellung der Systemzustände der beispielhaften Problemstellung im Netzwerk RM (optimaler Auftragseingang)

te 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 Instrumen-

Tabelle 2 Ergebnistabelle für die Problemstellung mit konkurrierenden Anfragen im Netzwerk RM

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

te der Marktbearbeitung wird die prognostizierte Verteilung der Auftragseingänge zum Irrtum und der maximal mögliche Gesamtertrag ist generiert.

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 . Die nachfolgende Abbildung 8 zeigt den zugehörigen Graphen mit den Systemzuständen, den möglichen Entscheidungen sowie der grafischen Darstellung der optimalen Politik und die Tabelle 2 zeigt die berechneten Erwartungswerte sowie die optimale Politik für jeden Systemzustand $[c_1 t]$.

Mit dem Beispiel wird klar, dass aufgrund des höheren Ertrags r_j und der höheren Wahrscheinlichkeit $p_j(t)$ über den gesamten Verlauf des Buchungshorizonts T die

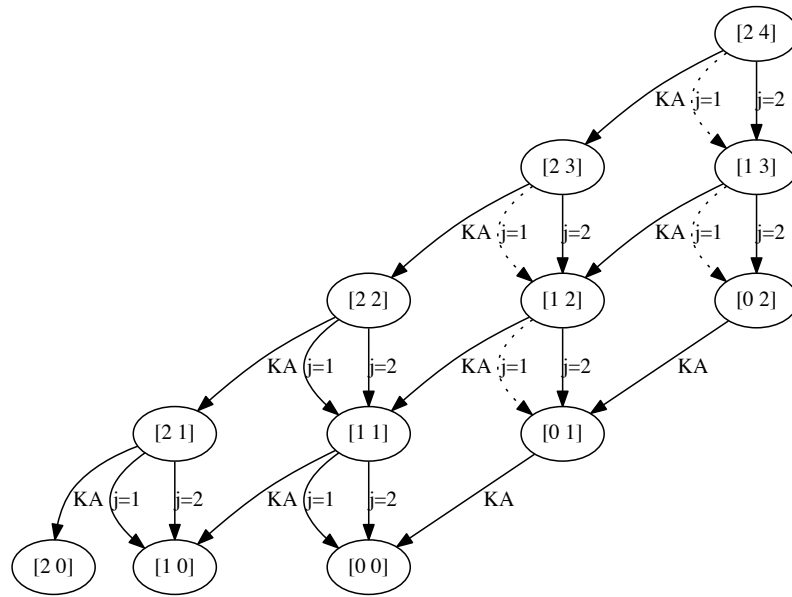


Abbildung 8 Darstellung der Systemzustände der Problemstellung mit konkurrierenden Anfragen im Netzwerk RM

Annahme des Produktauftrags $j = 2$ im Systemzustand $[3 \ 4]$ die optimale Politik ist. Sofern eine Produkthanfrage $j = 1$ zu dieser Periode t eintrifft, sollte die Ablehnung erfolgen. 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 des Netzwerks (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 8 ebenfalls zu erkennen, dass sofern am Anfang des Buchungshorizonts keine Anfrage eintrifft, die optimale Politik sich verändert. Die OC_j einer Produkthanfrage $j = 1$ nehmen im Zeitverlauf ab und die Entscheidung über die Annahme einer Anfrage $j = 1$ wird der optimalen Politik ergänzt. 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 RM zur Annahme von Aufträgen bei Dienstleistungen findet bereits über mehrere Dekanen in der wissenschaftlichen Literatur Anwendung.⁵⁵ Neue Veröffentlichungen versuchen das Konzept auf die Problemstellung der Annahme von Aufträgen der Kundeneinzelfertigung zu übertragen. 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.⁵⁶ 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, erweitert um neue Veröffentlichungen in dem Forschungsbereich. Die Tabelle zeigt jeweils zur Publikation den betrachteten KEP, die Anzahl der berücksichtigten Konsumerklassen des Modells und die Methodik der Arbeit. Die Konsumerklassen resultieren aus der für das Konzepts des RM 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 es zu Lagerfehlbeständen bei Niedrig-Preis-Segmenten führt, sofern stochastische Nachfrage auf fixe und kurzfristige Kapazität trifft. Jedoch rechtfertigen

⁵⁵Vgl. Klein (2001), S. 246.

⁵⁶Vgl. Kimms und Klein (2007), S. 1.

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

Autoren	KEP	#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: KEP: Kundenentkopplungspunkt, F: Fallstudie, K: Konzeption, L: Literaturüberblick, M: Simulations-/Analysemodell.

tigen diese Lagerfehlbestände eine Premiumpreisstrategie bei den Kundengruppen mit höherer Preisbereitschaft, was letztendlich zu Umsatzsteigerungen führt.⁵⁷ 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 RM als sogenannter 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 Er-

⁵⁷Vgl. H. deB. Harris und Pinder (1995), S. 307-308.

mittlung des Bid-Preises erfolgt bei den traditionellen Modellformulierungen des RM anhand eines *deterministischen linearen Programms* unter deterministisch eintreffender Nachfrage.⁵⁸ 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.⁵⁹ 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.⁶⁰ Viele Konzepte der neueren Veröffentlichungen im Bereich des Netzwerk RM sehen die Verwendung des Bid-Preises vor.⁶¹

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 ein Anwendungsfall aus der Branche der Eisen und Stahl erzeugenden Industrie untersucht. Es werden zwei Fallstudien betrachtet. Zum einen eine taktisch-operative Allokation der Kapazität und zum anderen eine operative Annahmeentscheidung. 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.⁶² 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.⁶³ Für die operative Entscheidungsunterstützung formuliert der Autor ein Modell des Netzwerk RM, welches durch ein spezielles Verfahren die Restkapazität bestimmt. Das Verfahren berechnet die Restkapazität nach einem

⁵⁸Vgl. Talluri und van Ryzin (2004), S. 107-108.

⁵⁹Vgl. Talluri und van Ryzin (2004b), S. 89.

⁶⁰Vgl. Talluri und van Ryzin (2004b), S. 86-87.

⁶¹Vgl. Petrick et al. (2010), S. 2028; Gönsch und Steinhardt (2013), S. 98-100.

⁶²Vgl. Rehkopf (2007), S. 113.

⁶³Vgl. Rehkopf (2007), S. 127-134.

real-time Ansatz für den zusicherbaren Bestand⁶⁴ aus der kapazitierten Hauptproduktionsprogrammplanung und den bereits eingeplanten Aufträgen des Netzwerks.⁶⁵ Ebenso wird hier der Ansatz des Bid-Preises eingesetzt, welcher durch ein duales lineares Optimierungsmodell ermittelt wird.⁶⁶ Dabei konnte er in der Fallstudie das Potential der Anwendung eines Bid-Preises und des angewandten Ansatzes zur Ermittlung der Restkapazität feststellen.⁶⁷ 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.⁶⁸ Dabei werden in der Publikation die Bid-Preise jeweils mit einer linearen Modellformulierung sowie einer multi-dimensionalen Knapsack-Modellformulierung berechnet und verglichen. Dabei wird deutlich, dass beide Verfahren den Deckungsbeitrag ähnlich verbessern, wobei sich das Lösen anhand der Knapsack-Modellformulierung als robuster darstellt.⁶⁹ In der Arbeit zeigten die Autoren, dass durch Anwenden der Heuristik sich der Gesamterlös im Vergleich zu einer einfachen Reihenfolgeannahme (FCFS) um 5,3% erhöhen lässt.⁷⁰

Volling et al. (2012) beschäftigen sich ebenfalls mit dem Auftragsannahmeproblem von MTO-Prozessen. Dabei legt die Arbeit u. a. den Fokus auf die Ermittlung und den Zeitpunkt der Neuberechnung des Bid-Preises für das 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

⁶⁴Vgl. den Ansatz „Available-to-promise (ATP)“.

⁶⁵Vgl. Rehkopf (2007), S. 140; Spengler et al. (2007), S. 160.

⁶⁶Vgl. Rehkopf (2007), S. 144.

⁶⁷Vgl. Rehkopf (2007), S. 177.

⁶⁸Vgl. Spengler et al. (2007), S. 157–171.

⁶⁹Vgl. Spengler et al. (2007), S. 168–169.

⁷⁰Vgl. Spengler et al. (2007), S. 170.

Bid-Preis und die erwartete Nachfrage in eine nicht lineare Beziehung gesetzt, bis beide Betrachtungsobjekte eine gleiche Nachfrageverteilung aufweisen.⁷¹ 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 exemplarisch 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. Dabei definieren die Autoren das RM im weitesten Sinne auch als mögliches Instrument der strategisch-taktische Entscheidungsunterstützung.⁷²

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 Systemkomponenten kann ein verbessertes Kapazitäts- und Preispolitik 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 über Systemlösungen erfolgt und dass das Preismanagement gegenüber des reinen Kapazitätsmanagements in den letzten Jahren an Bedeutung gewonnen hat.⁷³

⁷¹Vgl. Volling et al. (2012), S. 1026.

⁷²Vgl. Kimms und Klein (2005), S. 24.

⁷³Vgl. Kolisch und Zatta (2006), S. 40-41.

Anders als die vorher aufgeführten Autoren befassen sich Barut und Sridharan (2005) nicht nur mit dem Auftragsannahmeproblem und der Kapazitätsallokation, welches sich aus dem 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.⁷⁴ 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 verschiedenen Produktklassen mit unterschiedlichen Deckungsbeiträge erfolgt.⁷⁵

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.⁷⁶ 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, wobei im Vorfeld die Ermittlung der möglichen Durchführbarkeit erfolgt. Für die Durchführungsprüfung wird ein lineare Modell gelöst. Dies ist notwendig, damit der Zeitpunkt der angebotenen Fertigstellungszeitpunkt und Planungsgrundlage geprüft wird.⁷⁷ Auch hier werden die Opportunitätskosten durch Ermittlung eines Bid-Preises approximiert.⁷⁸ 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.⁷⁹

⁷⁴Vgl. Barut und Sridharan (2005), S. 291.

⁷⁵Vgl. Barut und Sridharan (2005), S. 310.

⁷⁶Vgl. Gühlich et al. (2015), S. 10.

⁷⁷Vgl. Gühlich et al. (2015), S. 12-14.

⁷⁸Vgl. Gühlich et al. (2015), S. 18-19.

⁷⁹Vgl. Gühlich et al. (2015), S. 22-24.

Kuhn und Defregger (2004) beschreiben die Anwendung des RM anhand eines 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 sie 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 mit der linearen Programmierung der optimalen 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 RM 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 interne 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.⁸⁰ 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 der 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 abhängig des Anwendungsfeld ergeben. Danach lässt sich für den Anwendungsfall eine Charakterisierung der relevanten Merkmale für das RM ermitteln. Die in dieser Arbeit betrachteten Auftragsfertigung bzw. Instandhaltung kann

⁸⁰Vgl. Specht und Gruß (2008), S. 66.

die in der Veröffentlichung von Quante et al. (2009) beschriebenen Fertigung von Maschinen gleichgesetzt werden. Bei der Maschinenfertigung ist der KEP vor der Produktion, daher handelt es sich um ein klassischen 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 eine Preisflexibilität ab dem Zeitpunkt des Auftragseingangs möglich. Die Autoren bestimmen des Weiteren die Profitheterogenität. Bei einem MTO-Prozess der Auftragsfertigung ist der Profit lt. den Autoren abhängig vom Zeitpunkt und des eintreffenden Kundenauftrags. Zusammengesetzt wird er durch den erzielten Ertrag, der ebenfalls vom Zeitpunkt und Auftrag abhängig ist. Die Ertrag wird um die Kosten reduziert und um einen Wert der strategische Bedeutung erhöht. Diese beiden Werte sind wiederum nur abhängig vom Kundenauftrag.⁸¹

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. Es werden verschiedene Modelle in ihren Ausprägungen der eingesetzten Parametern verglichen, wie z. B. bei den Modellen der stochastischen Lagerbestandskontrolle, des „vielversprechsten Auftrags“ und des traditionellen RM.⁸² 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 und Zuverlässigkeit sowie die kurze Reaktionszeiten der Auftragsannahme.⁸³

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. Damit bilden die Autoren neben der Kapazitätssteuerung auch Verfahren der Segmentierung und der Bepreisung ab, da aus ihrer Sicht die richtige Ansprache der unterschiedlichen Kundengruppen eines Fertigungsun-

⁸¹Vgl. Quante et al. (2009), S. 37.

⁸²Vgl. Quante et al. (2009), S. 43-44.

⁸³Vgl. Quante et al. (2009), S. 56-57.

ternehmens hohe Relevanz aufweist.⁸⁴ Durch die Marktsegmentierung ist es einem Unternehmen 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 die Kapazität nicht aufgrund eines ablaufenden Planungshorizonts. Die verbleibenden Kapazitäten könnten 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.⁸⁴ 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 zeitdiskreter Markov-Entscheidungsprozess und einem heuristischen Verfahren 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.⁸⁵

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.⁸⁶

⁸⁴Vgl. Cheraghi et al. (2010), S. 64-65.

⁸⁵Vgl. Defregger und Kuhn (2007), S. 150-154.

⁸⁶Vgl. Sucky (2009), S. 89.

5 Eine Modellformulierung für Auftragsannahme- und Lagerhaltungsentscheidungen bei auftragsbezogenen Instandhaltungsprozessen

5.1 Modellannahmen zur Berücksichtigung von Lagerhaltungsentscheidungen

Zur Berücksichtigung von Lagerhaltungsentscheidungen im Netzwerk RM wird für die Erweiterung der Gleichung (1) der Parameter für den Lagerbestands y_h benötigt. In Bezug des Ausgangsproblems der Lagerhaltungsentscheidung ist dieser Schritt der Modellerweiterung nötig, damit ein Parameter für bereits reparierte Produkte im Modell vorhanden ist. Das Modell berücksichtigt Produkte die im gleichen Verhältnis zu einer jeweiligen Ressource $h \in \mathcal{H}$ substituierbar sind. Es existiert damit für jede Ressource $h \in \mathcal{H}$ ein möglicher Lagerbestand y_h . Die Obergrenze eines jeden Lagerbestands wird als Parameter y_h^{max} beschrieben. Die einzelnen Lagerbestände können als Vektor \mathbf{y} bzw. \mathbf{y}^{max} zusammengefasst werden.

Das Modell für Auftragsannahme- und Lagerhaltungsentscheidungen soll es ermöglichen ertragsarme Anfragen j nach auftragsbezogenen Instandhaltungsprozessen abzulehnen und eine Erhöhung des Lagerbestands y_h über den Parameter a_{hj} zu ermöglichen, sofern die Annahme dieser Anfragen relevante Ressourcenkapazitäten c_h für im Buchungshorizonts T spätere eintreffende ertragreiche Anfragen j reduzieren würde. Weiter ist eine Annahme von Anfragen j nach Instandhaltungsprozessen über den Lagerparameter y_h möglich, indem anstelle der Kapazitäts- 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.

Für die Modellerweiterung werden Produkthanfragen betrachtet, die für bestimmte Leistungserstellungszeitpunkte vorgesehen sind. Ein Leistungserstellungszeitpunkt wird durch den Parameter \hat{t} definiert. Damit existiert für eine jede Ressourcen $h \in \mathcal{H}$ eine Kapazität $c_h^{\hat{t}}$ und ein Lagerbestand $y_h^{\hat{t}}$ für einen jeden Leistungserstellungszeitpunkt \hat{t} . Damit wird den Parametern für den Kapazitäts- und Lagerbestand der hochgestellte Index zum zugehörigen Buchungsabschnitt \hat{t} ergänzt. Die Menge aller Leistungserstellungszeitpunkte wird mit dem Parameter \hat{T} abgebildet. Zur Vereinfachung des Modells wird nur eine einzelne Ressource h über den gesamten Buchungshorizont betrachtet. Der jeweilige Eintrag im Vektor $\mathbf{c}^{\hat{t}}$ entspricht somit der Kapazität $c_h^{\hat{t}}$ der Ressource $h = 1$ zum Leistungserstellungszeitpunkt \hat{t} . Der aktuell betrachtete Leistungserstellungszeitpunkt \hat{t} wird ermittelt durch nachfolgende Gleichung:⁸⁷

$$\hat{t} = \left\lfloor \frac{\max(T) - t}{\frac{\max(T)}{\max(\hat{T})}} \right\rfloor + 1 \quad (8)$$

Es gilt die Annahme, dass die Leistungserstellungszeitpunkte $\hat{t} \in \hat{T}$ gleichmäßig über den gesamten Buchungshorizont T aufteilen sind. Die bis zur Leistungserstellung benötigten Perioden werden durch Division des Buchungshorizonts T und der Leistungszeitpunkte \hat{T} ermittelt. Dabei gilt die Bedingung, dass der Buchungshorizont T durch die Anzahl der Leistungszeitpunkte \hat{t} ganzzahlig teilbar sein muss.

Der Kapazitätsverbrauch $a_{hj}^{\hat{t}}$ einer Produkthanfrage j beschreibt mit dem jeweiligen Eintrag den Zeitpunkt der Erfüllung der Anfrage und dem daraus resultierenden Verbrauch der Ressource $h = 1$ zur Leistungserstellung \hat{t} . Sei \tilde{j} eine Anfrage nach einem Produkt für diese Modellerweiterung, dann beschreibt der Parameter $\mathbf{a}_{\tilde{j}}^{\hat{t}} = (0, 1, 0)$, dass die Erfüllung der Anfrage für den Leistungserstellungszeitpunkt $\hat{t} = 2$ vorgesehen ist. Bspw. erfolgt für eine Anfrage \tilde{j} mit einem Kapazitätsverbrauch $\mathbf{a}_{\tilde{j}}^{\hat{t}} = (0, 1, 0)$ für einen Buchungshorizont in Länge von $T = 15$ eine Leistungserstellung der Anfrage j mit Ablauf der sechsten Periode t . Damit wird ab der Periode $t = 5$ eine neue Leistungserstellung betrachtet.

⁸⁷Vgl. Helber und Heinze (2014).

Dem modifizierten Modell soll es erlaubt sein, durch Betrachten der Leistungserstellungszeitpunkte $\hat{t} \in \hat{T}$ die vorzeitige Inanspruchnahme von Kapazitäten $c_h^{\hat{t}}$ für Produktanfragen j für nachfolgende Leistungserstellungszeitpunkte \hat{t} zu ermöglichen, was der Annahme der Instandhaltungsauftrags entspricht. Diese Veränderung resultiert aus der Tatsache, dass die Bestandsveränderung $a_j^{\hat{t}}$ einer Anfrage j bereits vor des eigentlichen Leistungserstellungszeitpunkts \hat{t} den Zugriff auf die Kapazität $c^{\hat{t}}$ oder den Lagerbestand $y^{\hat{t}}$ erlaubt. Jedoch darf kein Zugriff des Parameters $a_j^{\hat{t}}$ nach Überschreitung des Leistungserstellungszeitpunkts \hat{t} auf die jeweiligen Bestände $c^{\hat{t}}$ bzw. $y^{\hat{t}}$ erfolgen. Anders formuliert bedeutet dies, dass nicht beanspruchte Kapazitäten $c_h^{\hat{t}}$ 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 $j \in \mathcal{J}$ gesteuert. Dabei ist der Zeitpunkt an dem die erste Produktanfrage j eintrifft unerheblich. Mit dieser Modellerweiterung ist eine umfassendere Betrachtung der Auftragsannahme in der Auftragsfertigung als im Vergleich zum Grundmodell 1 aus Kapitel 3 möglich.⁸⁸ Abbildung 9 zeigt die Rahmenbedingungen der Modellerweiterung als grafische Darstellung. Dabei soll der Zusammenhang von Buchungsperioden und der Leistungserstellungszeitpunkte verdeutlicht werden. Ebenfalls zeigt die Abbildung beispielhaft welche Produktanfragen in welchen Zeiträumen eintreffen könnten.

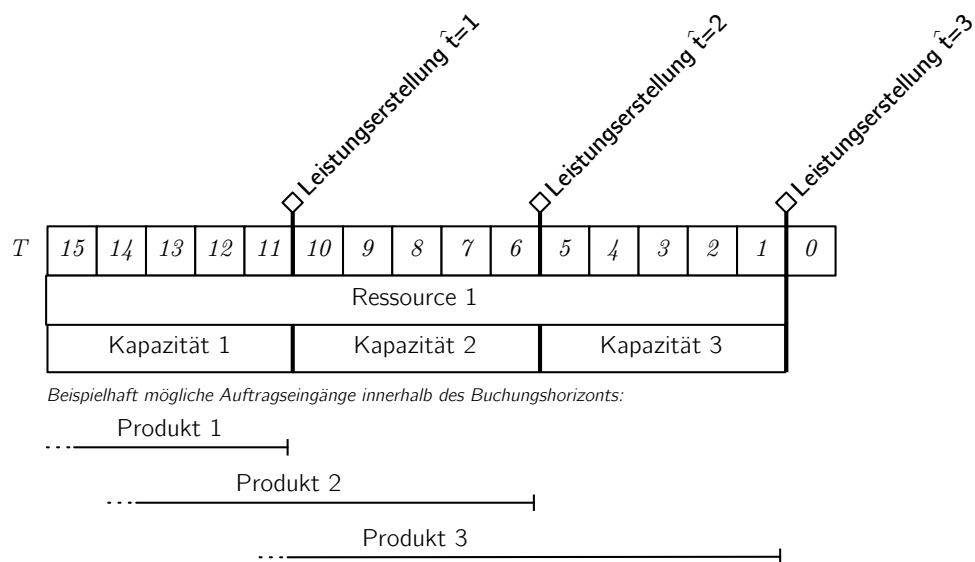


Abbildung 9 Zusammenhang von Buchungsperioden und den Leistungserstellungszeitpunkten

In Anlehnung an: Helber und Heinze (2014).

⁸⁸Vgl. Helber und Heinze (2014).

5.2 Mathematische Formulierung eines Modells zur Entscheidungsunterstützung bei Auftragsannahme- und Lagerhaltungsentscheidungen

Für die Modellerweiterung erfolgt die Anpassung der *Bellman'schen Funktionsgleichung* (1) um die Entscheidung der Lagerentnahme in Form des Terms $r_j + V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}} - Y(\mathbf{a}_j^{\hat{t}}, \hat{t}), t-1)$. 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 Entscheidung der gewollten Ablehnung einer Anfrage j zur Produktion eines Lagerbestand $y_h^{\hat{t}}$ wird mit dem Term $V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j^{\hat{t}}, \mathbf{y}^{\hat{t}} + Y(\mathbf{a}_j^{\hat{t}}, \hat{t}+1), t-1)$ dargestellt. Für das Modellformulierung gilt weiterhin, dass eine Ressource und eine Einheit des Lagerbestands im gleichen Verhältnis substituierbar sind.

Sofern eine Reduzierung des Lagerbestand $\mathbf{y}^{\hat{t}}$ als Entscheidung gewählt ist, wird der Lagerparameter bzw. die einzelnen Eintragungen des Vektors des Lagerparameters für alle betreffenden Leistungserstellungszeitpunkte $\hat{t} \in \hat{T}$ reduziert. Weiter ist dem Modell gestattet, eine Entscheidung über die Ablehnung einer Anfrage j zu berücksichtigen, bei der aufgrund einer Reduktion der Ressourcenkapazität $\mathbf{c}^{\hat{t}}$ eine Erhöhung des Lagerbestand $\mathbf{y}^{\hat{t}}$ um jeweils des Parameters $\mathbf{a}_j^{\hat{t}}$ möglich ist. Eine Erhöhung des Lagerbestands $\mathbf{y}^{\hat{t}}$ einer jeden Ressource h erfolgt über den gesamten Buchungshorizont und ist ab dem nachfolgenden Leistungserstellungszeitpunkt \hat{t} verfügbar.

Zur Ermittlung der Veränderung des Lagerbestands aufgrund der Lagerentnahme oder der Lagerproduktion ist die Hilfsfunktion $Y(\mathbf{a}_j^{\hat{t}}, \hat{t})$ erforderlich, die den Parameter für den Kapazitätsverbrauch $\mathbf{a}_j^{\hat{t}}$ über alle nachfolgenden Leistungserstellungszeitpunkten kumuliert.⁸⁹ Als Beispiel wird der Kapazitätsverbrauch $\mathbf{a}_j^{\hat{t}} = (0, 1, 1, 0)$ betrachtet. Ein Auftrag j mit einem Verbrauch des vorher aufgeführten Parameters $\mathbf{a}_j^{\hat{t}}$ zeigt eine Leistungserstellung über mehrere Perioden auf. Die Hilfsfunktion zur Ermittlung der Lagerbestandsveränderung berechnet damit eine Veränderung in Höhe des Vektors $Y(\mathbf{a}_j^{\hat{t}}, \hat{t}) = (0, 1, 2, 2)$. Sofern es sich um eine Veränderung in Form einer Lagerproduktion handelt, ist der Lagerbestand erst zur nachfolgenden Leistungserstellungsperiode $\hat{t} + 1$ verfügbar. D. h. die Hilfsfunktion erstellt

⁸⁹Vgl. Helber und Heinze (2014).

dem Vektor $Y(\mathbf{a}_j^{\hat{t}}, \hat{t} + 1) = (0, 0, 1, 2)$. Es muss beachtet werden, dass ein Auftrag j mit einem Kapazitätsverbrauch von $\mathbf{a}_j = (0, 0, 1, 2)$ eine Lagererhöhung $Y(\mathbf{a}_j, \hat{t} + 1) = (0, 0, 0, 1)$ verursacht. Diese Erhöhung liegt außerhalb des Betrachtungszeitraums. 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:

$$\begin{aligned}
 V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t) = & \sum_{j \in \mathcal{J}} p_j(t) \max \left[\underbrace{V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1)}_{\text{Ablehnung}}, \underbrace{r_j + V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1)}_{\text{Annahme via Kapazität}}, \right. \\
 & \underbrace{r_j + V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}} - Y(\mathbf{a}_j^{\hat{t}}, \hat{t}), t-1)}_{\text{Annahme via Lagerentnahme}}, \underbrace{V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j^{\hat{t}}, \mathbf{y}^{\hat{t}} + Y(\mathbf{a}_j^{\hat{t}}, \hat{t} + 1), t-1)}_{\text{Ablehnung und Lagerproduktion}} \left. \right] \\
 & + p_0(t) \max_{\substack{j \in \mathcal{J}, \\ p_j(t) > 0}} \left[\underbrace{V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t-1)}_{\text{Keine Lagerproduktion}}, \underbrace{V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j^{\hat{t}}, \mathbf{y}^{\hat{t}} + Y(\mathbf{a}_j^{\hat{t}}, \hat{t} + 1), t-1)}_{\text{Lagerproduktion}} \right]
 \end{aligned} \tag{9}$$

Die Gleichung (9) 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 Kapazität“, „Annahme via Lagerentnahme“ sowie „Lagerproduktion“ gewählt. Damit geht aus der Gleichung (9) hervor, dass es zu jedem Erwartungswert, somit für jedes Teilproblem des Netzwerks, für eine jede Anfrage j die vier Optionen bzw. Entscheidungen existieren. Es gibt damit für eine jede Produkthanfrage j im gesamten Netzwerk einen Wert, der die optimale Politik beim Eintreffen einer Anfrage beschreibt. Dieser Wert der optimalen Politik beschreibt die Variable $d_j(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t) \forall j \in \mathcal{J}$. Dieser Variable existiert für jedes Produkt und ist abhängig vom Systemzustand. Ein Systemzustand wird durch die verbleibende Kapazität $\mathbf{c}^{\hat{t}}$ und des Lagerbestands $\mathbf{y}^{\hat{t}}$ zur Periode t beschrieben. Die Variable kann ebenfalls als Vektor $\mathbf{d}(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t)$ 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 Politik. Die Variable nimmt den Wert $d_j(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t) = 1$ an, sofern die Option der Auftragsannahme via Kapazität den höchsten Erwartungswert liefert und den Wert $d_j(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t) = 2$,

wenn die Auftragsannahme via Lagerbestand der optimalen Politik entspricht. Sofern die Ablehnung des Auftrags erfolgen soll, aber inkl. der Entscheidung der Lagerproduktion, dann nimmt die Variable $d_j(\mathbf{c}^t, \mathbf{y}^t, t)$ den Wert 3 an. Sofern die optimale Politik die generelle Ablehnung der Anfrage ist, nimmt die Variable den Wert $d_j(\mathbf{c}^t, \mathbf{y}^t, t) = 0$ an.

Für die verschiedenen Entscheidungsalternativen existieren unterschiedliche OC_j , wie aus der Gleichung (9) zu erkennen ist. Damit trägt der Parameter OC_j den Superskript d für die jeweilig möglichen Entscheidungsalternative aufgrund der Modellformulierung. In dieser Modellerweiterung existieren vier unterschiedliche Entscheidungen, die den möglichen Ausprägungen der optimalen Politik der Variable $d_j(\mathbf{c}^t, \mathbf{y}^t, t)$ entsprechen. Sofern die verschiedenen möglichen Entscheidungen im betrachteten Systemzustand gleichwertig sind, was 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 Entscheidungen 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:

$$d_j(\mathbf{c}^t, \mathbf{y}^t, t) := \begin{cases} 1, & \text{für } r_j - OC_j^1 \geq r_j - OC_j^2 \\ 2, & \text{für } r_j - OC_j^2 \geq -OC_j^3 \\ 3, & -OC_j^3 > 0 \\ 0, & \text{sonst} \end{cases}. \quad (10)$$

Die *Bellman'schen Funktionsgleichung* (9) zeigt ebenfalls den Term an, sofern keine Anfrage eintrifft. Bei der Auftragsannahme von Kundenaufträgen unter Berücksichtigung von Lagerhaltungsentscheidung muss beachtet werden, dass auch Entscheidungsalternativen beim Einreffen keiner Produkthanfragen $j \in \mathcal{J}$ bestehen. Sofern keine Anfrage eintrifft, gibt es die Entscheidung über alle möglichen Produkthanfragen $j \in \mathcal{J}$ die Lagerproduktion durch Inanspruchnahme der Kapazitäten durchzuführen. Damit existiert auch eine optimale Politik, sofern keine Anfrage $j \in \mathcal{J}$ zur Periode t eintrifft. Für diese optimale Politik wird die Variable $d_0(\mathbf{c}^t, \mathbf{y}^t, t)$ verwendet. Es bestehen die gleichen Abhängigkeiten wie bei der Variable für die optimale Politik bei Auftragseingang ($d_j(\mathbf{c}^t, \mathbf{y}^t, t)$). Die Variable zeigt jedoch nicht einen Index über die optimale Entscheidung, sonder sie zeigt

den Index der Produkthanfrage j die für die Lagerproduktion durch Verwendung des zugehörigen Ressourceneinsatzes bzw. Ausführungsmodus \mathbf{a}_j verwendet werden soll. Sofern keine Lagerproduktion die optimale Politik ist, nimmt die Variable den Wert 0 an. Dabei wird für der optimalen Politik $d_0(\mathbf{c}^t, \mathbf{y}^t, t)$ zur Vereinfachung des Modells angenommen, dass nur dann eine Lagererhöhung erfolgen kann, sofern Anfragen nach Produkten $j \in \mathcal{J}$ möglich sind ($p_j(t) > 0$). Damit wird die Variable wie folgt definiert:

$$d_0(\mathbf{c}^t, \mathbf{y}^t, t) := \begin{cases} j, & \text{für } \{j \mid \max_{\substack{j \in \mathcal{J}, \\ p_j(t) > 0}} -OC_j^3\} \\ 0, & \text{sonst} \end{cases}. \quad (11)$$

Im nachfolgenden Schritt kann die Gleichung (9) analog der Gleichung (1) umgeformt werden:

$$\begin{aligned} V(\mathbf{c}^t, \mathbf{y}^t, t) &= \sum_{j \in \mathcal{J}} p_j(t) V(\mathbf{c}^t, \mathbf{y}^t, t-1) + \sum_{j \in \mathcal{J}} p_j(t) \max[0, \\ &\quad r_j - V(\mathbf{c}^t, \mathbf{y}^t, t-1) + V(\mathbf{c}^t - \mathbf{a}_j^t, \mathbf{y}^t, t-1), \\ &\quad r_j - V(\mathbf{c}^t, \mathbf{y}^t, t-1) + V(\mathbf{c}^t, \mathbf{y}^t - Y(\mathbf{a}_j^t, \hat{t}), t-1), \\ &\quad V(\mathbf{c}^t - \mathbf{a}_j^t, \mathbf{y}^t + Y(\mathbf{a}_j^t, \hat{t} + 1), t-1) - V(\mathbf{c}^t, \mathbf{y}^t, t-1)] \\ &\quad + p_0(t) V(\mathbf{c}^t, \mathbf{y}^t, t-1) \\ &\quad + p_0(t) \max_{\substack{j \in \mathcal{J}, \\ p_j(t) > 0}} [0, V(\mathbf{c}^t - \mathbf{a}_j^t, \mathbf{y}^t + Y(\mathbf{a}_j^t, \hat{t} + 1), t-1) \\ &\quad - V(\mathbf{c}^t, \mathbf{y}^t, t-1)] \\ &= V(\mathbf{c}^t, \mathbf{y}^t, t-1) + \sum_{j \in \mathcal{J}} p_j(t) \max[0, \\ &\quad r_j - V(\mathbf{c}^t, \mathbf{y}^t, t-1) + V(\mathbf{c}^t - \mathbf{a}_j^t, \mathbf{y}^t, t-1), \\ &\quad r_j - V(\mathbf{c}^t, \mathbf{y}^t, t-1) + V(\mathbf{c}^t, \mathbf{y}^t - Y(\mathbf{a}_j^t, \hat{t}), t-1), \\ &\quad V(\mathbf{c}^t - \mathbf{a}_j^t, \mathbf{y}^t + Y(\mathbf{a}_j^t, \hat{t} + 1), t-1) - V(\mathbf{c}^t, \mathbf{y}^t, t-1)] \\ &\quad + p_0(t) \max_{\substack{j \in \mathcal{J}, \\ p_j(t) > 0}} [0, V(\mathbf{c}^t - \mathbf{a}_j^t, \mathbf{y}^t + Y(\mathbf{a}_j^t, \hat{t} + 1), t-1) \\ &\quad - V(\mathbf{c}^t, \mathbf{y}^t, t-1)] \end{aligned} \quad (12)$$

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

$j = \{1, 2, 3\}$, $h = \{1\}$, $r_1 = 100$, $r_2 = 200$, $r_3 = 5000$,
 Startperiode $t = 3$, Anzahl Leistungserstellungen $\hat{T} = 3$,

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

$$\mathbf{y}^{\hat{t}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \mathbf{y}^{\hat{t},max} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$$

Bei dem Beispiel wird ein Netzwerk mit drei Produkthanfragen $j \in \mathcal{J}$ betrachtet. Es existieren drei Leistungserstellungszeitpunkte \hat{t} für eine Ressource $h \in \mathcal{H}$. Die Annahmen der Produkthanfragen $j \in \mathcal{J}$ generieren einen Ertrag in Höhe von $r_1 = 100, r_2 = 200$ und $r_3 = 5000$. Die Kapazitäten für die Ressourcen betragen $\mathbf{c}^{\hat{t}} = (1, 1, 1)$ und die Bestandsveränderungen für die Anfragen betragen $\mathbf{a}_1^{\hat{t}} = (1, 0, 0)$, $\mathbf{a}_2^{\hat{t}} = (0, 1, 0)$ sowie $\mathbf{a}_3^{\hat{t}} = (0, 0, 2)$. Der Buchungshorizont beträgt $T = 3$. Es existiert kein Lagerbestand $\mathbf{y}^{\hat{t}}$ und es kann nur ein maximaler Lagerbestand $\mathbf{y}^{\hat{t},max} = (2, 2, 2)$ 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. Abbildung 10 zeigt für das Beispiel die möglichen Systemzustände mit allen Übergängen. Dabei wird ein Systemzustand im Netzwerk als Zahlenfolge $[c_1^{\hat{t}=1} \ c_1^{\hat{t}=2} \ c_1^{\hat{t}=3} \ y_1^{\hat{t}=1} \ y_1^{\hat{t}=2} \ y_1^{\hat{t}=3} \ t]$ beschrieben.

Wie in der Abbildung 10 zu erkennen ist, wäre die optimale Politik $d_j(\mathbf{c}, \mathbf{y}, t)$ im Systemzustand $[1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 3]$ eine Produkthanfrage $j = 1$ abzulehnen und eine Bestandsveränderung des Lagers zu bewilligen (Lagerproduktion) sowie die Annahme der Produkthanfrage j anhand der vorhandenen Kapazität. 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

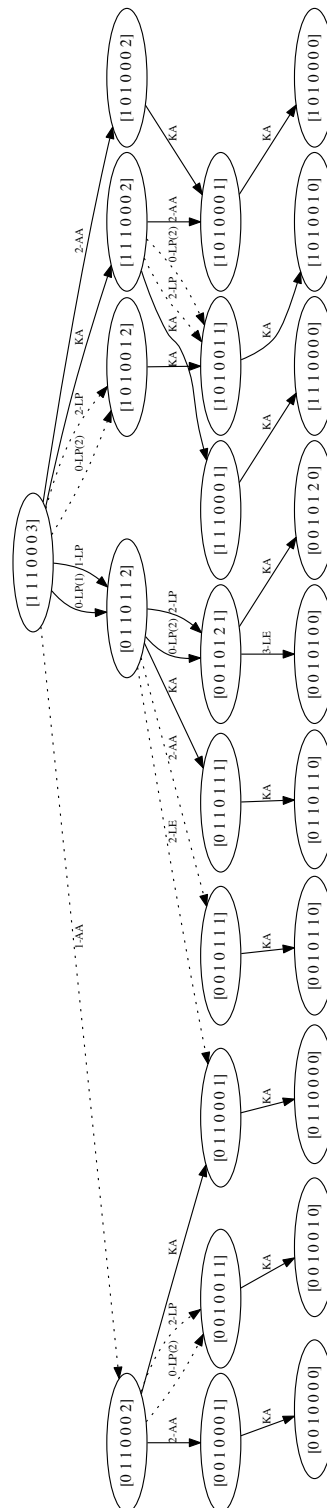


Abbildung 10 Darstellung der Systemzustände der Problemstellung unter Beachtung der Möglichkeit der Auftragsannahme- und Lagerhaltungsentscheidung

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

Systemzustand $[1\ 1\ 1\ 0\ 0\ 0\ 3]$ eintrifft, ist die optimale Politik die Lagerproduktion unter Verwendung der Kapazitäten. Damit erreicht das Netzwerk den Systemzustand $[0\ 1\ 1\ 0\ 1\ 1\ 2]$. Erkennbar ist, dass die Reduktion der Kapazität $c_1^{\hat{t}=1}$ auf 0 erfolgt und nach dem Leistungserstellungszeitpunkt $\hat{t} = 1$ ein Lagerbestand $\mathbf{y}^{\hat{t}}$ zur Verfügung steht. Eine Auftragsannahme der Anfrage $j = 1$ gehört in diesem Netzwerk nicht zur optimalen Politik.

Im Systemzustand $[0\ 1\ 1\ 0\ 1\ 1\ 2]$ ist aufgrund der Wahrscheinlichkeitsverteilung $p_j(t)$ keine Anfrage nach Produkt $j = 1$ 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 inkl. einer Lagerproduktion der Produkthanfrage $j = 2$ alleiniger 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 $\mathbf{y}^{\hat{t}} = (0, 1, 2)$ 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, dann würde ein Unternehmen unter Einhaltung der optimalen Politik ein Gesamtertrag 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 $c^{\hat{t}}$. Aufgrund dieser Entscheidung ist ein Gesamtertrag in Höhe von 200 GE erzielt. Eine Entscheidung über die Lagerproduktion unter diesen Parametergegebenheiten hätte in Bezug der Zielsetzung der Maximierung des Gesamtdeckungsbeitrags keinen weiteren Nutzen. Wie in der Abbildung 10 zu erkennen ist, würde eine Lagerproduktion zwar den Bestand an Ressourcen auf Lager erhöhen, jedoch wäre im weiteren Verlauf keine Annahme einer anderen Produkthanfrage $j \in \mathcal{J}$ möglich.

Trifft im Ausgangssystemzustand $[1\ 1\ 1\ 0\ 0\ 0\ 3]$ keine Anfrage j ein, dann ist die optimale Politik $d_0(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t) = 1$. D. h. sofern keine Anfrage $j \in \mathcal{J}$ zur Periode $t = 3$ eintrifft, wäre es die optimale Politik des Unternehmens den Parameter der Bestandsveränderung \mathbf{a}_j der Produkthanfrage $j = 1$ zu verwenden, damit sich der Lagerbestand $\mathbf{y}^{\hat{t}}$ ab der Leistungsperiode $\hat{t} = 2$ um eine Einheit erhöht. Anders formuliert, das Unternehmen sollte die vorhandene Kapazität nicht verfallen lassen und

Tabelle 4 Optimale Politik für die beispielhafte Problemstellung unter Beachtung von Auftragsannahme- und Lagerhaltungsentscheidungen

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	1	2	0.0	0	0	0	0
0	1	1	0	0	0	2	60.0	0	0	1	0
1	0	1	0	0	0	2	0.0	0	0	0	0
1	1	1	0	0	0	2	60.0	0	0	1	0
0	1	1	0	1	1	2	1050.0	2	0	3	0
0	1	1	0	0	0	1	0.0	0	0	0	0
0	0	1	0	0	1	1	0.0	0	0	0	0
1	0	1	0	0	1	1	0.0	0	0	0	0
1	1	1	0	0	0	1	0.0	0	0	0	0
1	0	1	0	0	0	1	0.0	0	0	0	0
0	1	1	0	1	1	1	0.0	0	0	0	0
0	0	1	0	1	2	1	1500.0	0	0	0	2
0	0	1	0	1	1	1	0.0	0	0	0	0
0	0	1	0	0	0	1	0.0	0	0	0	0

eine Lagerproduktion veranlassen. Die weitere optimale Politik für dieses Netzwerk ist, sofern keine Produkthanfragen eintreffen, das Verfolgen einer optimalen Politik, die eine Annahme der Produkthanfrage $j = 3$ zur letzten Periode t 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 $\hat{t} \in \hat{T}$ entspricht. Zur besseren Veranschaulichung wird ein umfangreicheres Beispiel berechnet:

$$j = \{1, 2, 3, 4\}, \quad h = \{1\}, \quad r_1 = 100, \quad r_2 = 5000, \quad r_3 = 100, \quad r_4 = 5000, \\ \text{Startperiode } t = 10, \quad \text{Anzahl Leistungserstellungen } \hat{T} = 3,$$

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

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

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

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 führt die jeweiligen Erwartungswerte und die optimalen Politik für jeden Systemzustand auf. Dabei wird in diesem Beispiel ein Systemzustand als Zahlenfolge $[c_1^{\hat{t}-1} \ c_1^{\hat{t}-2} \ y_1^{\hat{t}-1} \ y_1^{\hat{t}-2} \ t]$ 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 $c_h^{\hat{t}}$ der Produkthanfrage $j = 1$ genutzt werden, damit ein Lagerbestand $y_h^{\hat{t}}$ generiert wird. Der Lagerbestand 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 $\mathbf{a}_j^{\hat{t}}$ der Anfrage j wird nicht aufgewendet um den Lagerbestand $\mathbf{y}^{\hat{t}}$ um $Y(\mathbf{a}_j^{\hat{t}}, \hat{t}+1)$ 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 $c_h^{\hat{t}}$ aus $\hat{t} = 1$ wäre aufgrund des ablaufenden Buchungshorizonts weiterhin verfügbar.

Tabelle 5 Optimale Politik für die zweite beispielhafte Problemstellung unter Beachtung von Auftragsannahme- und Lagerhaltungsentscheidungen

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	1	0	0	8	4165.375	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	0	0	0	8	0.0	0	0	0	0	0
1	0	0	0	8	3949.65014648	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
1	0	0	0	7	3417.60009766	1	3	1	0	0
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	0	6	0.0	0	0	0	0	0
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	1	0	0	6	3696.0	0	0	0	0	1
0	0	0	1	6	3696.0	0	0	0	0	2
0	0	0	0	5	0.0	0	0	0	0	0
0	1	0	1	5	4701.0	0	0	0	0	1
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	1	0	0	5	3370.0	0	0	0	0	1
1	0	0	0	5	0.0	0	0	0	0	0
1	0	0	0	4	0.0	0	0	0	0	0
0	1	0	1	4	3886.0	0	0	0	0	1
0	0	0	1	4	2962.5	0	0	0	0	2
0	0	0	0	4	0.0	0	0	0	0	0
0	1	0	0	4	2962.5	0	0	0	0	1
1	1	0	0	4	2962.5	0	0	0	0	1
1	0	0	0	3	0.0	0	0	0	0	0
0	1	0	0	3	2453.0	0	0	0	0	1
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	0	0	0	3	0.0	0	0	0	0	0
0	0	0	1	3	2453.0	0	0	0	0	2
0	0	0	1	2	1816.0	0	0	0	0	2
1	0	0	0	2	0.0	0	0	0	0	0
1	1	0	0	2	1816.0	0	0	0	0	1
0	1	0	0	2	1816.0	0	0	0	0	1
0	1	0	1	2	2040.0	0	0	0	1	1
0	0	0	0	2	0.0	0	0	0	0	0
0	1	0	0	1	1020.0	0	0	0	1	1
0	0	0	1	1	1020.0	0	0	0	2	2
0	0	0	0	1	0.0	0	0	0	0	0
0	1	0	1	1	1020.0	0	0	0	1	1
1	1	0	0	1	1020.0	0	0	0	1	1
1	0	0	0	1	0.0	0	0	0	0	0

Sofern keine Anfragen in dem Beispiel eintreffen, ergibt sich ein differenziertes Bild bzgl. der optimalen Politik. Sofern Kapazitäten die $c_h^{\hat{t}}$ 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 Produkthanfrage $j = 2$ oder $j = 4$ eingetroffen und dementsprechend die Kapazität reduziert ist, wird das Modell die Kapazität $c_h^{\hat{t}=1} = 1$ zum Lagerbestand $y_h^{\hat{t}=2}$ umschichten. Dies erfolgt mit dem Bestandveränderungsparameter $a_j^{\hat{t}}$ für $j = 1$ und nur dann, wenn keine Anfrage $j \in \mathcal{J}$ in den Zeitpunkten $t > 5$ eintreffen. Damit folgt die optimale Politik $d_0(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t)$ der optimalen Politik $d_j(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t)$.

5.3 Grundlegendes zum Lösen und Implementieren des Auftragsannahmeproblems

Gleichung (9) zeigt die mathematische Modellformulierung für das Auftragsannahmeproblem im Netzwerk RM mit Lagerhaltungsentscheidungen. In diesem Abschnitt wird ein mögliches Lösungsverfahren für das Optimierungsmodell gezeigt, damit eine Implementierung in ein Computersystem möglich ist. Mittels einer solchen Implementierung kann das exakte Lösen von vordefinierten Szenarien erfolgen. Bei der *Bellman'sche Funktionsgleichung* (9) aus Abschnitt 5.2 handelt es um ein stochastisch, dynamisches Optimierungsproblem in rekursiver Form.⁹⁰ Das Optimierungsmodell besteht aus verschiedenen gleichartigen Teilproblemen und zur Lösung des gesamten Optimierungsmodells müssen alle Teilprobleme gelöst werden.⁹¹ Diese Art der rekursiven Modellformulierung geht auf das von dem amerikanischen Mathematiker Richard Bellman entwickelte Konzept der **Dynamischen Programmierung (DP)** zurück.⁹² Bei diesem Konzept werden die berechneten Teilergebnisse gespeichert und bei der weiteren Berechnung des Optimierungsmodells verwendet.⁹³ Durch eine solche Implementierung wird die rekursive Berechnung des Modells verbessert, da der Zugriff auf bereits berechnete Teilergebnisse erfolgt, sofern die einzelnen Teilergebnisse sich überlappen.

⁹⁰Vgl. Petrick (2009), S. 185.

⁹¹Vgl. Powell (2007), S. 26, 31-33.

⁹²Vgl. Bellman (1954), S. 4; Bellman (1952), S. 716-717.

⁹³Vgl. Owsnicki-Klewe und Wissner (1999), S. 197.

Eine Form der Implementierung des Konzepts der DP ist das Anwenden einer *Memofunktion*. Mit der Memofunktion werden die bereits berechneten Lösungen des Optimierungsmodells gespeichert. Sofern die rekursive Folge auf ein bereits berechnetes Teilproblem stößt, wird auf die abgespeicherte Lösung zurückgegriffen.⁹⁴ Das stochastisch, dynamisches Optimierungsmodell lässt sich als Graph interpretieren, bei dem die Knoten die Teilprobleme und die Kanten die einzelnen Übergängen in die nachfolgenden Teilprobleme sind. Zur Verdeutlichung einer Memofunktion im Netzwerk RM in der Auftragsannahme wird ein einfaches Beispiel eingeführt: $j = \{1, 2\}$, $h = \{1\}$, $c_1 = 2$, $a_{11} = 1$, $a_{12} = 1$, Startperiode $t = 2$.

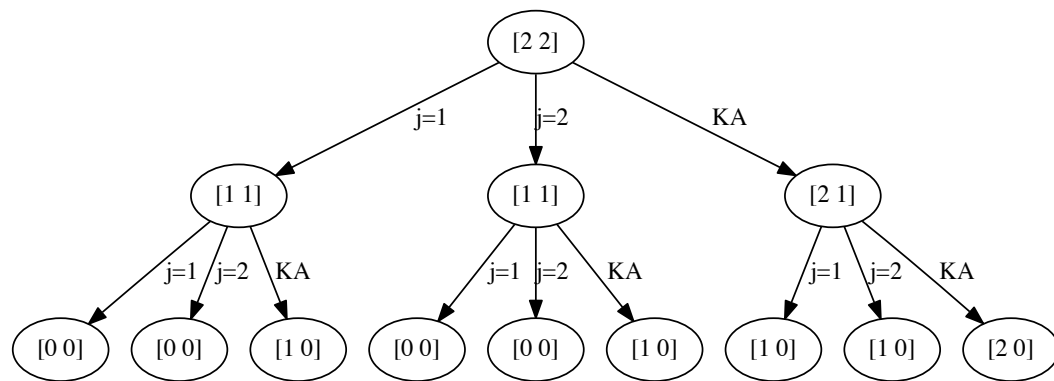


Abbildung 11 Rekursive Übergänge der Systemzustände ohne Memofunktion

In Anlehnung an: Hetland (2010), S. 179. **Legende:** Annahme eines Produktauftrags entspricht 'j', KA='Kein Auftrag'

Abbildung 11 zeigt die möglichen Systemzustände in Form der Knoten und die 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 Teilprobleme jeweils mehrfach ermittelt, da sie jeweils für das vorherige Teilproblem notwendig sind.

Durch Anwenden einer Memofunktion können die möglichen Übergänge der Systemzustände des Beispiels reduziert werden, wie Abbildung 12 zeigt. Jedes Systemzustand ist nur einmal im Zustandsraum 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 $j = 1$

⁹⁴Vgl. Hetland (2010), S. 176-181.

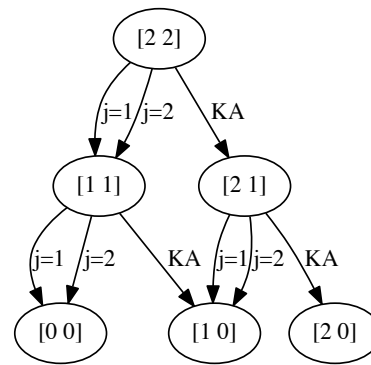


Abbildung 12 Rekursive Übergänge mit Memofunktion

In Anlehnung an: Hetland (2010), S. 179. **Legende:** Annahme einer Produktauftrag entspricht 'j', KA='Kein Auftrag'

gelöst ist, erfolgt keine Berechnung des gleichen Systemzustands $[1 \ 1]$ aufgrund der Anfrage nach Produkt $j = 2$. 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 (12). 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. Das Lösen des Optimierungsproblems erfolgt damit 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 $V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t)$

Eingabe: *Memofunktion*

Eingabe: $\mathcal{H}, \mathcal{J}, T, \hat{T}, \mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, \mathbf{a}_j^{\hat{t}}, r_j, p_j(t)$

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

wenn $t \neq 0$ dann

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

für jedes $j \in \mathcal{J}$ tue

wenn $\mathbf{c}^{\hat{t}} - \mathbf{a}_j^{\hat{t}} \geq \mathbf{0}$ dann

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

sonst

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

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

wenn $\mathbf{y}^{\hat{t}} - Y(\mathbf{a}_j^{\hat{t}}, \hat{t}) \geq \mathbf{0}$ dann

$value_j^{storage} =$

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

sonst

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

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

wenn $\mathbf{c}^{\hat{t}} - \mathbf{a}_j^{\hat{t}} \geq \mathbf{0}$ and $\mathbf{y}^{\hat{t}} + Y(\mathbf{a}_j^{\hat{t}}, \hat{t} + 1) \leq \mathbf{y}^{\hat{t}, max}$ dann

$value_j^{workup} =$

$p_j(t) \cdot \max[0, V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j^{\hat{t}}, \mathbf{y}^{\hat{t}} + Y(\mathbf{a}_j^{\hat{t}}, \hat{t} + 1), t - 1) - value^{reject}]$

sonst

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

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

Ende

$list = [0]$

für jedes $j \in \mathcal{J}$ tue

wenn $p_j(t) > 0$ dann

$list = list + [V(\mathbf{c}^{\hat{t}} - \mathbf{a}_j^{\hat{t}}, \mathbf{y}^{\hat{t}} + Y(\mathbf{a}_j^{\hat{t}}, \hat{t} + 1), t - 1) - value^{reject}]$

sonst

$list = list + [0]$

Ende

$V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t) = value^{reject} + \sum_{j \in \mathcal{J}} (value_j^{accept} + value_j^{storage} + value_j^{workup}) + p_0(t) \max[list]$

sonst

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

$Memofunktion = Memofunktion + [V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t)]$

sonst $V(\mathbf{c}^{\hat{t}}, \mathbf{y}^{\hat{t}}, t)$ aus *Memofunktion*;

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

Bei der Implementierung des Algorithmus zum Lösen der Modellerweiterung des Auftragsannahmeproblems im Netzwerk RM mit Lagerhaltungsentscheidungen wurde die Programmiersprache Python/2.7.1 verwendet. Es handelt sich um eine höhere Programmiersprache mit einem Interpreter.⁹⁵ Als Kommandozeileninterpreter wird in dieser Arbeit 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.⁹⁶ Zusätzlich wird die Python-Bibliothek SciPy/0.15.1 genutzt, die weitere Werkzeuge für die NumPy-Datenstrukturen liefert.⁹⁷ Die Laufzeitverbesserung kommt zu Stande, da die Funktionen der Programmbibliothek auf homogene Datenstrukturen zurückgreifen.⁹⁸ Damit lassen sich annähernd ähnliche Ergebnisse wie mit Fortran, C und C++ erzielen, wobei weiterhin die einfache Syntax von Python/2.7.1 besteht. 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 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.

⁹⁵Vgl. Hetland (2005), S. 8.

⁹⁶Vgl. Lindblad und Kinser (2013), S. 35-49.

⁹⁷Vgl. Lindblad und Kinser (2013), S. 49-52.

⁹⁸Vgl. Langtangen (2006), S. 131.

6 Numerische Untersuchung

Für die numerische Untersuchung wurden mehrere Berechnungen der optimalen Politik für bestimmte Szenarien mit unterschiedlichen Eigenschaften und verschiedener Eintrittswahrscheinlichkeiten der Produkthanfragen 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 Form 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 der verwendeten Hardware des Clustersystems an der Leibniz Universität Hannover

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 des Netzwerk RM mit Lagerhaltungsentscheidung sieht die Berechnung der möglichen Systemzustände und des maximal möglichen Erwartungswerts vor. Die Memofunk-

tion wird gespeist durch die vorher berechneten Systemzustände und ist wiederum bei der Berechnung der Erwartungswerte für die tatsächlich benötigten Systemzustände notwendig (vgl. Kapitel 5.3). Bei der Berechnung der Erwartungswerte wird ein angepasstes DP des Netzwerk RM verwendet. Im Anhang ist der verwendete Quellcode hinterlegt. Zusätzlich erfolgt mithilfe der Implementierung, nachdem die Berechnung der Erwartungswerte vollzogen ist, 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 vorgestellten Rechner-System 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 Szenarien dargestellt. Je Szenario gibt es eine Anzahl an Buchungsperioden T und eine Anzahl an Leistungserstellungszeitpunkten \hat{T} . Die Leistungserstellungszeitpunkte $\hat{t} \in \hat{T}$ teilen sich dabei gleichmäßig über den Buchungshorizont T auf. 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_{hj} 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 Auftrag j angegeben und über alle Produktaufträge $j \in \mathcal{J}$ des jeweiligen Szenarios als grafischer Kurvenverlauf abgebildet.

Szenario 1

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 $\hat{t} \in \hat{T}$ vorgesehen und die Anfragen wechseln sich dabei jeweils zwischen dem möglichen Erträgen $r_j = 100$ und $r_j = 1000$ ab. Die Wahrscheinlichkeitsverteilung einer jeden Produkthanfrage verläuft dabei li-

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):				0:00:00.098259	
Anzahl möglicher Systemzustände:				785376	
Anzahl benötigter Systemzustände:				35324	
Rechenzeit DP (h):				6:39:09.675572	
Max. Erwartungswert (GE):				4998.52	

near über den Buchungshorizont T mit dem konstanten Wert $p_j(t) = 0,09$, wobei mit Überschreitung des Leistungserstellungszeitpunkts \hat{t} die Eintrittswahrscheinlichkeit der zugehörigen Produkthanfrage j für alle nachfolgenden Perioden t auf $p_j(t) = 0$ sinkt (vgl. Abbildung 13). Tabelle 7 zeigt die verwendeten Parameter des Szenarios und das Ergebnis der Berechnung.

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 und der Relation der gewählten optimalen Politik des Szenarios, jeweils sofern keine Anfrage (d_0) und sofern eine Anfrage eintrifft ($d_j \forall j \in \mathcal{J}$). Wie in der Tabelle 8 zu erkennen ist, wird für die

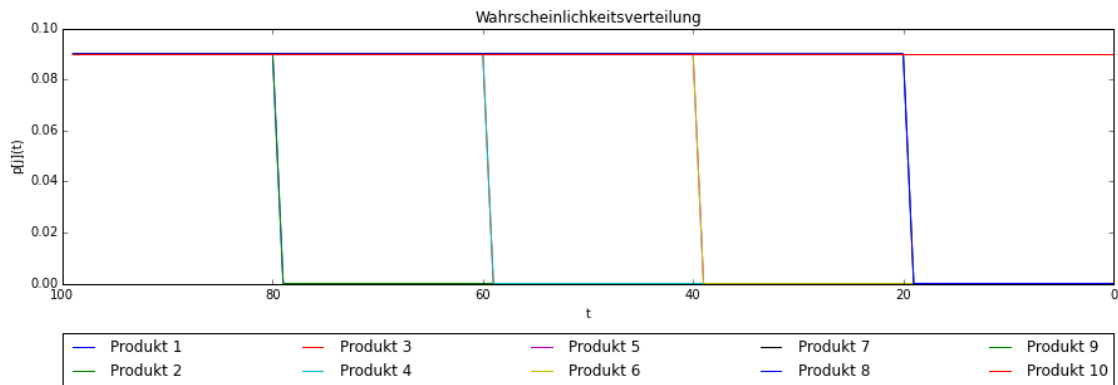


Abbildung 13 Wahrscheinlichkeitsverteilung Szenario 1

Tabelle 8 Auswertung des Szenarios 1

j	0	1	2	3	4	5	6	7	8	9	10
d_0	24151	1168	0	2787	0	3226	0	3628	0	0	0
d_0 (%)	69.08	3.34	0	7.97	0	9.23	0	10.38	0	0	0

d_j	Ablehnung (0)		Annahme (1)		Lagerentnahme (2)		Lagerproduktion (3)	
	Anz.	%	Anz.	%	Anz.	%	Anz.	%
d_1	33752	96.54	40	0.11	0	0	1168	3.33
d_2	33464	95.72	1175	3.35	0	0	321	0.92
d_3	32019	91.59	44	0.13	0	0	2897	8.26
d_4	29915	85.57	3066	8.75	1092	3.12	887	2.53
d_5	31401	89.82	44	0.13	0	0	3515	10.03
d_6	24944	71.35	6015	17.17	3272	9.34	729	2.08
d_7	30819	88.16	30	0.09	0	0	4111	11.73
d_8	17557	50.22	10109	28.87	6966	19.88	328	0.94
d_9	33780	96.62	880	2.51	300	0.86	0	0
d_{10}	8720	24.94	16871	48.22	9369	26.74	0	0

Produktanfragen $j = 1$ eher die optimale Politik der Lagerproduktion gewählt, anstelle der Auftragsannahme. Für Produkthanfragen $j = 2$ gilt dies nicht, da der

Ertrag dieser Anfragen höher ist und damit empfiehlt das Modell eher die Auftragsannahme. Der weitere Verlauf der Anzahl der gewählten optimalen Politik für ertragsarme Anfragen j steigt über den Buchungshorizont dabei nicht merklich an. Sofern keine Anfragen eintreffen, ist die vom Modell gewählte optimale Politik d_0 die Verwendung von Anfragen mit niedrigem Ertrag für eine mögliche Lagerhaltung.

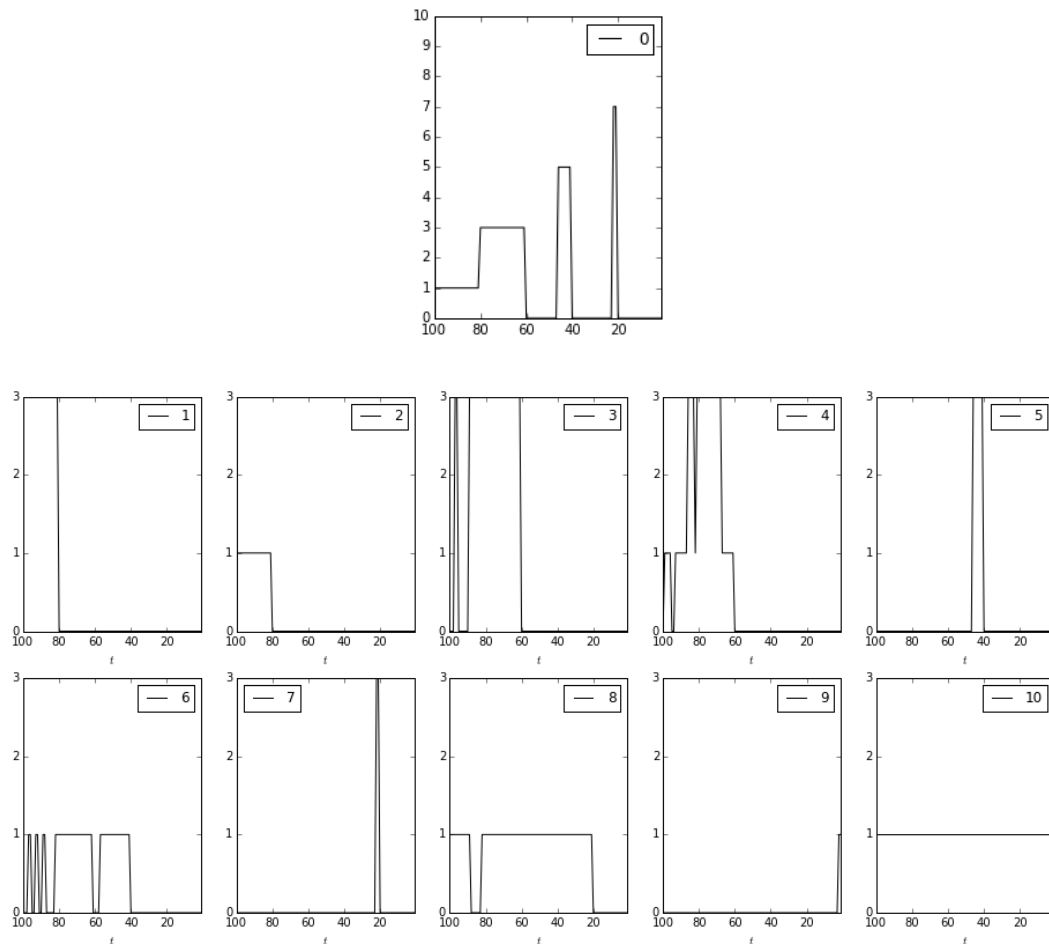


Abbildung 14 Veränderung der optimalen Politik für das Szenario 1 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produktanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 10)

Damit die Veränderung der optimalen Politik d_0 und $d_j \forall j \in \mathcal{J}$ im Zeitverlauf untersucht werden kann, wird eine Auswertung für einen bestimmten Verlauf im System der jeweiligen Szenarien betrachtet. D. h. es werden bestimmte Auftrags-eingänge angenommen und es wird die optimale Politik, die aufgrund dieses Verlaufs resultiert, erfasst. Für dieses und die nachfolgenden Szenarien wird angenommen, dass über den gesamten Buchungshorizont T keine Anfrage angenommen

wird. Damit bleiben die Ressourcenkapazitäten $\mathbf{c}^{\hat{t}}$ sowie der Lagerbestände $\mathbf{y}^{\hat{t}}$ der Ressource $h = 1$ für die Leistungserstellungszeitpunkte $\hat{t} \in \hat{T}$ über alle Perioden $t \in T$ konstant. Da es sich hier um ein dynamisches Optimierungsmodell handelt, ist eine solche Auswertung innerhalb des Netzwerks nur ein möglicher Verlauf der optimalen Politik. Sofern zu einer beliebigen Periode t eine andere Entscheidung getroffen bzw. eine andere Kante des Graphen gewählt wird, könnte sich die optimale Politik des Netzwerks für alle Perioden t ab dem Zeitpunkt der Entscheidung ändern. Aufgrund des vordefinierten Verlaufs ist auch für alle $t \in T$ die optimale Politik der Lagerentnahme nicht möglich ($d_j = 2$). Dies resultiert aus der Tatsache, dass für eine solche optimale Politik ein Lagerbestand $\mathbf{y}^{\hat{t}}$ notwendig ist.

Abbildung 14 zeigt den Verlauf der Veränderung der optimalen Politik für das Szenario 1 sofern keine Anfragen eintreffen und je Produkthanfrage bei konstanter Ressourcenkapazität. Der erste Verlauf der Abbildung zeigt die optimale Politik bzgl. der Entscheidung, ob eine Lagerproduktion durchgeführt werden soll wenn keine Anfrage eintrifft und zeigt damit die optimale Politik d_0 . Damit lässt sich anhand dieses Kurvenverlaufs ablesen, durch welche Anfragen j eine Lagerproduktion zur Ertragsmaximierung erfolgen soll. Wie zu erkennen ist, besteht die optimale Politik d_0 für die Perioden $100 \geq t \geq 81$ (gesamte Anzahl der Perioden t bis zur Leistungserstellung $\hat{t} = 1$) darin, anfangs den Modus der Produkthanfrage $j = 1$ für eine Lagerproduktion zu verwenden. D. h. der Lagerbestand wäre ab der Leistungsperiode $\hat{t} = 2$ verfügbar. Sofern diese Anfrage $j = 1$ nicht mehr möglich ist, wechselt das Modell für die Perioden $80 \geq t \geq 61$ zur optimalen Politik $d_0 = 3$. Sofern auch diese Anfrage $j = 3$ nicht mehr möglich ist, werden anfangs keine möglichen Produkthanfragen $j \in \mathcal{J}$ für eine mögliche Lagerproduktion als optimale Politik ermittelt. Erst durch Annäherung an den Leistungserstellungszeitpunkt $\hat{t} = 3$ wird erneut die Lagerproduktion als optimale Politik vom Modell ermittelt. Es wird eine optimale Politik der Lagerproduktion anhand des Ausführungsmodus für die ertragsarme Produkthanfrage $j = 5$ verwendet. Zum Ausführungszeitpunkt $\hat{t} = 4$ ist ein ähnliches Bild im Bezug zur Produkthanfrage $j = 7$ zu erkennen, wobei der Zeitraum der optimale Politik zur Lagerproduktion erneut kürzer vor dem Leistungserstellungszeitpunkt ist.

Die weiteren Verläufe der Abbildung 14 zeigen jeweils für die Produkthanfrage $j \in \mathcal{J}$ die optimale Politik d_j . D. h. die möglichen Entscheidungen „Auftragsablehnung ($d_j = 0$)“, „Auftragsannahme via Kapazität ($d_j = 1$)“, „Auftragsannahme via Lagerentnahme ($d_j = 2$)“ und „Auftragsablehnung und Lagerproduktion ($d_j = 3$)“ für den vordefinierten Verlauf des Netzwerks über alle Perioden $t \in T$. Für Pro-

duktanfrage $j = 1$ zeigt die optimale Politik für den produktspezifisch möglichen Zeitraum ausschließlich $d_1 = 3 \forall t \in T$. Die Produkthanfrage $j = 2$ werden ausschließlich angenommen ($d_2 = 1 \forall t \in T$). Bei Produkthanfragen $j = 3$ erfolgt nur die optimale Politik der Lagerproduktion, aber nicht über die gesamten möglichen Perioden $t \in T$. Bei Produkthanfragen $j = 4$ wechselt sich die optimale Politik der Auftragsannahme und der Lagerproduktion ab. Die Lagerproduktion ist ausschließlich optimale Politik für Anfragen $j = 5$. Diese optimale Politik stellt sich jedoch kurz vor dem Verderben der Kapazität ein. Produkthanfragen $j = 6$ werden ausschließlich angenommen, aber nicht über den gesamten möglichen Zeitraum $t \in T$. Es gibt Intervalle an denen die Anfragen abgelehnt werden. Anfragen $j = 7$ werden kurz vor dem Leistungserstellungszeitpunkt $\hat{t} = 4$ auf Lager produziert. Die optimale Politik für Produkthanfragen $j = 8$ lautet $d_8 = 1$ über alle möglichen Perioden $t \in T$, wobei eine Unterbrechung in dem zum Leistungserstellungszeitpunkt $\hat{t} = 1$ zugehörigen Buchungszeitraum $100 \geq t \geq 81$ stattfindet. Für Produkthanfragen $j = 9$ kommt nur die Lagerentnahme kurz vor Verfall der Kapazität in Betracht. Die Auftragsannahme von Produkthanfragen $j = 10$ ist über den gesamten Buchungshorizont T die optimale Politik.

Szenario 2

Das zweite Szenario zeichnet sich durch linear ansteigende Eintrittswahrscheinlichkeiten $p_j(t)$ der Produkthanfragen $j \in \mathcal{J}$ aus. Dabei steigt die Eintrittswahrscheinlichkeit für eine jeweilige Anfrage j bis zum Leistungserstellungszeitpunkt \hat{t} auf $p_j(t) = 0,3$ an, wobei die Steigung an jeweils unterschiedlichen Perioden startet. Die weiteren Eigenschaften der Produkthanfragen basieren auf den des Szenarios 1. Tabelle 9 und Abbildung 15 zeigen die Parameter für das Szenario 2.

Tabelle 10 zeigt die optimale Politik für das Szenario 2. Aufgrund der Wahrscheinlichkeitsverteilung $p_j(t)$ und aufgrund des hohen Ertrags r_j von Produkthanfrage $j = 8$ wird diese tendenziell gegenüber der anderen Produkthanfragen $j < 8$ öfters gewählt. In diesem Szenario steigt die Anzahl der Entscheidungen über eine Lagerproduktion mit ablaufenden Buchungshorizonts für Anfragen mit niedrigem Ertrag an. Die Anzahl an Systemzustände ist unter einer solchen Verteilung geringer, da nicht alle Anfragen zu jeder Periode verfügbar sind. Dies reduziert auch die Berechnungszeit des Algorithmus. Tendenzuell erfolgt die Entscheidungen über eine Lagerproduktion nicht so häufig wie bei einer konstant linearen Verteilung. Jedoch wird in diesem Szenario keine Anfrage $j = 1$ angenommen, sondern ausschließlich für die Lagerproduktion verwendet. Für die weiteren ertragsarmen Produkthanfragen

Tabelle 9 Szenario 2

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 ansteigend auf $p_j(t) = 0,3$ von 100 bis 81	
2	1000	1	1	Linear ansteigend auf $p_j(t) = 0,3$ von 100 bis 81	
3	100	1	2	Linear ansteigend auf $p_j(t) = 0,3$ von 100 bis 61	
4	1000	1	2	Linear ansteigend auf $p_j(t) = 0,3$ von 100 bis 61	
5	100	1	3	Linear ansteigend auf $p_j(t) = 0,3$ von 60 bis 41	
6	1000	1	3	Linear ansteigend auf $p_j(t) = 0,3$ von 60 bis 41	
7	100	1	4	Linear ansteigend auf $p_j(t) = 0,3$ von 80 bis 21	
8	1000	1	4	Linear ansteigend auf $p_j(t) = 0,3$ von 80 bis 21	
9	100	1	5	Linear ansteigend auf $p_j(t) = 0,3$ von 40 bis 1	
10	1000	1	5	Linear ansteigend auf $p_j(t) = 0,3$ von 40 bis 1	
Rechenzeit Systemzustände (h):				0:00:00.094716	
Anzahl möglicher Systemzustände:				785376	
Anzahl benötigter Systemzustände:				17667	
Rechenzeit DP (h):				1:36:44.968302	
Max. Erwartungswert (GE):				4998.84	

ist die Auftragsannahme eher vereinzelt die optimale Politik. Sofern keine Anfragen eintreffen wird eine Lagerproduktion als optimale Politik im Vergleich zu einem linearen Verlauf der Eintrittswahrscheinlichkeiten tendenziell weniger gewählt. Nur mit ablaufenden Buchungshorizont wird die Politik der Lagerproduktion vom Modell häufiger als optimal bestimmt.

Die Auswertung des Verlaufs der optimalen Politik, sofern keine Anfragen über den Buchungshorizont T akzeptiert werden, zeigt die Abbildung 16 für das Szenario 2. Die optimale Politik d_0 ist jeweils die Lagerproduktion der ertragsarmen Produktanfragen des Szenarios. Wobei diese optimale Politik ungefähr im mittleren Bereich des zeitlich vorherigen Buchungsabschnitts vom produktspezifischen

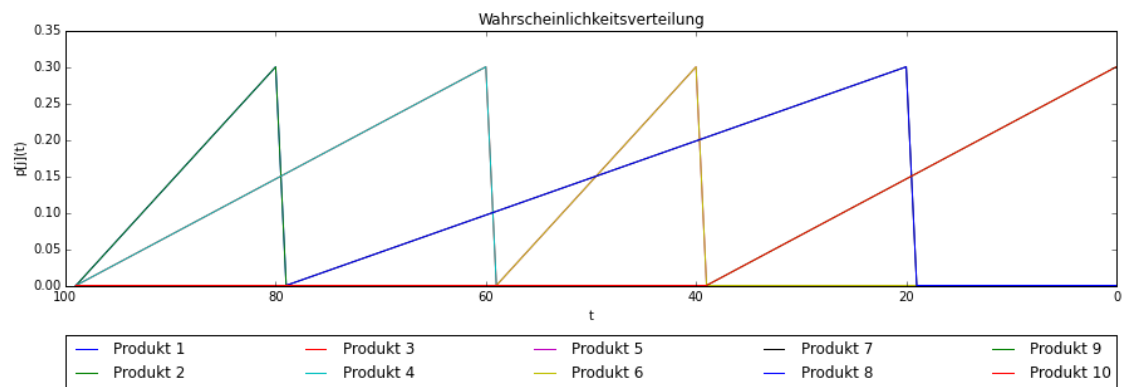


Abbildung 15 Wahrscheinlichkeitsverteilung Szenario 2

Tabelle 10 Auswertung des Szenarios 2

j	0	1	2	3	4	5	6	7	8	9	10
d_0	15297	31	0	97	0	644	0	1234	0	0	0
d_0 (%)	88.41	0.18	0	0.56	0	3.72	0	7.13	0	0	0

d_j	Ablehnung (0)		Annahme (1)		Lagerentnahme (2)		Lagerproduktion (3)	
	Anz.	%	Anz.	%	Anz.	%	Anz.	%
d_1	17272	99.82	0	0	0	0	31	0.18
d_2	17255	99.72	48	0.28	0	0	0	0
d_3	17204	99.43	1	0.01	0	0	98	0.56
d_4	16931	97.85	232	1.33	140	0.8	0	0
d_5	16652	96.24	7	0.04	0	0	644	3.7
d_6	15909	91.94	944	5.43	450	2.59	0	0
d_7	16057	92.8	12	0.07	0	0	1234	7.09
d_8	11026	63.72	3664	21.1	2613	15.03	0	0
d_9	16849	97.38	332	1.91	122	0.7	0	0
d_{10}	6713	38.8	6934	39.98	3656	21.03	0	0

Leistungserstellungszeitpunkt \hat{t} eintritt. In diesem Szenario ist die optimale Politik d_j eindeutig für die ertragsarmen und ertragreichen Produkthanfragen $j \in \mathcal{J}$. Pro-

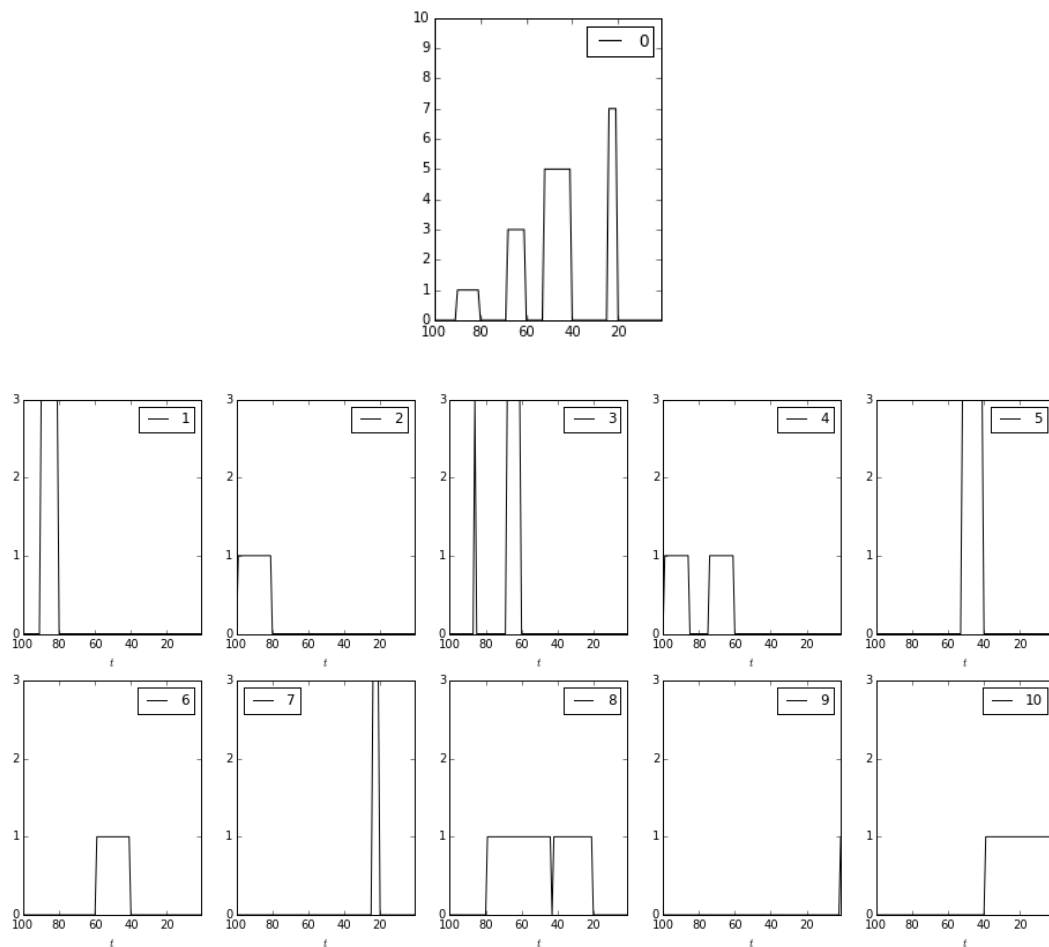


Abbildung 16 Veränderung der optimalen Politik für das Szenario 2 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 10)

dukthanfragen mit hohem Ertrag r_j werden angenommen und Produkthanfragen mit geringem Ertrag r_j werden auf Lager produziert. Jedoch zeigt sich kein stetiger Verlauf für die jeweilige Produkthanfrage j . D. h. wie in Abbildung 16 zu erkennen ist, gibt es des Öfteren Unterbrechungen bei den jeweiligen optimalen Politiken. Bspw. ist bei der Produkthanfrage $j = 3$ eine längere Unterbrechung zwischen den ersten beiden Buchungsabschnitten ermittelt. Bei der Produkthanfrage $j = 8$ ist wiederum eine kurze Unterbrechung vor der Leistungserstellung $\hat{t} = 3$ eingetreten. Die Unterbrechungen resultieren aufgrund der neu eintretenden Konkurrenz Anfragen im Buchungshorizont.

Szenario 3

Bei dem Szenario 3 handelt es sich um normalverteilte Eintrittswahrscheinlichkeiten $p_j(t)$ für die Produkthanfragen $j \in \mathcal{J}$. Dabei ist für jede Anfrage der Erwartungswert μ und die Standardabweichung σ angegeben. Zu beachten ist, dass aufgrund des unter Umständen umfangreichen Zeitraums der Wahrscheinlichkeitsverteilung $p_j(t)$ die zugehörige Normalverteilung $\mathcal{N}(\mu, \sigma)$ einer Produkthanfrage j um einen Skalar normiert ist. Andernfalls wären die Werte der Wahrscheinlichkeitsverteilung für solche Anfragen viel geringer als die für Anfragen mit einem kurzen Betrachtungszeitraum und einer niedrigen Standardabweichung σ . Dies spiegelt z. B. die Verteilung von Produkt 9 im Vergleich zu Produkt 1 in Tabelle 11 wider. Weiter zeigt die Tabelle alle anderen Parameterausprägungen für das Szenario 3. Die weiteren Eigenschaften der Produkthanfragen orientieren sich anhand der bereits beschriebenen Szenarien. Abbildung 17 zeigt die Wahrscheinlichkeitsverteilung des Szenarios.

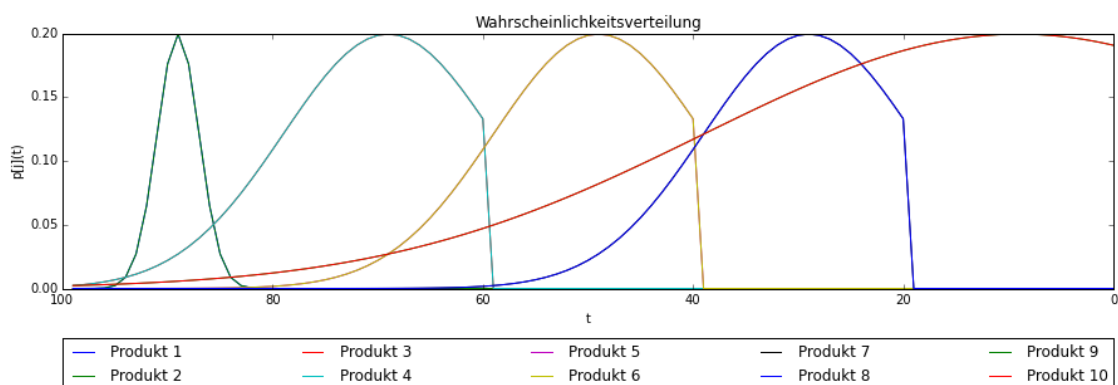


Abbildung 17 Wahrscheinlichkeitsverteilung Szenario 3

Sofern eine solche Wahrscheinlichkeitsverteilung $p_j(j)$ angenommen wird, verhält sich die optimale Politik d_j ähnlich der Verteilung mit einem linearen Verlauf der Eintrittswahrscheinlichkeiten (siehe Auswertung des Szenarios 1). Tabelle 12 zeigt die Auswertung des Szenarios 3. Die optimale Politik bzgl. einer Lagerhaltungsentscheidung wird eher für Produkthanfragen j mit geringem Ertrag r_j ermittelt. Auch bei einer möglichen Entscheidung der Lagerproduktion, sofern keine Anfrage eintrifft bzw. für die optimalen Politik d_0 , sind ähnliche Werte wie bei einem linearen Verlauf der Eintrittswahrscheinlichkeiten $p_j(t)$ festzustellen.

Abbildung 18 zeigt den Verlauf der optimalen Politik für das Szenario 3. Auch hier

Tabelle 11 Szenario 3

	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	Normalverteilt $\mathcal{N}(90, 2)$
2	1000	1	1	Normalverteilt $\mathcal{N}(90, 2)$
3	100	1	2	Normalverteilt $5 \cdot \mathcal{N}(70, 10)$
4	1000	1	2	Normalverteilt $5 \cdot \mathcal{N}(70, 10)$
5	100	1	3	Normalverteilt $5 \cdot \mathcal{N}(50, 10)$
6	1000	1	3	Normalverteilt $5 \cdot \mathcal{N}(50, 10)$
7	100	1	4	Normalverteilt $5 \cdot \mathcal{N}(30, 10)$
8	1000	1	4	Normalverteilt $5 \cdot \mathcal{N}(30, 10)$
9	100	1	5	Normalverteilt $15 \cdot \mathcal{N}(10, 30)$
10	1000	1	5	Normalverteilt $15 \cdot \mathcal{N}(10, 30)$

Rechenzeit Systemzustände (h):	0:00:00.052650
Anzahl möglicher Systemzustände:	785376
Anzahl benötigter Systemzustände:	35324
Rechenzeit DP (h):	3:01:16.875038
Max. Erwartungswert (GE):	4999.91

gilt der vordefinierte Verlauf innerhalb des Netzwerks, sodass keine Anfragen innerhalb des Buchungshorizonts T angenommen werden. Die optimale Politik d_0 ist erneut die Verwendung der Ausführungsmodi der ertragsarmen Produkthanfragen j für die Lagerproduktion. Wobei unter Annahme dieser Wahrscheinlichkeitsverteilung $p_0(t)$ über alle Perioden $t \in T$ im Buchungsabschnitt $60 \geq t \geq 41$ die optimale Politik $d_0 = 5$ einige Unterbrechungen aufweist. Die Produkthanfrage $j = 4$ und $j = 6$ zeigen in diesem Szenario eine interessante optimale Politik d_j . Für Produkthanfragen $j = 4$ ist die optimale Politik d_4 zum Anfang des Buchungshorizonts T die Auftragsannahme und im späteren Verlauf die Lagerproduktion. Die Entscheidung der Lagerproduktion folgt damit aufgrund des anstehenden Verderbens der Ressourcenkapazität und der Wahrscheinlichkeitsverteilung. Bei der

Tabelle 12 Auswertung des Szenarios 3

j	0	1	2	3	4	5	6	7	8	9	10
d_0	28843	561	0	931	0	1679	0	2946	0	0	0
d_0 (%)	82.5	1.6	0	2.66	0	4.8	0	8.43	0	0	0

d_j	Ablehnung (0)		Annahme (1)		Lagerentnahme (2)		Lagerproduktion (3)	
	Anz.	%	Anz.	%	Anz.	%	Anz.	%
d_1	34259	97.99	140	0.4	0	0	561	1.6
d_2	33649	96.25	1198	3.42	0	0	113	0.32
d_3	34006	97.27	22	0.06	0	0	932	2.66
d_4	30576	87.46	2954	8.43	1251	3.57	179	0.51
d_5	33148	94.82	22	0.06	60	0.17	1730	4.94
d_6	25345	72.5	5909	16.87	3206	9.15	500	1.43
d_7	31769	90.87	12	0.03	80	0.23	3099	8.84
d_8	17642	50.46	10097	28.84	7024	20.05	197	0.56
d_9	34316	98.16	454	1.29	190	0.54	0	0
d_{10}	8677	24.82	16791	48	9492	27.09	0	0

optimalen Politik der Produkthanfragen $j = 6$ ist ein ähnliches Bild erkennbar. Im Intervall mit den größten Wahrscheinlichkeitswerten $p_6(t)$ ist die optimale Politik $d_6 = 3$. Sofern die Wahrscheinlichkeitswerte $p_6(t)$ jedoch eher gering sind, scheint die Auftragsannahme ($d_6 = 1$) oftmals die optimale Politik zu sein.

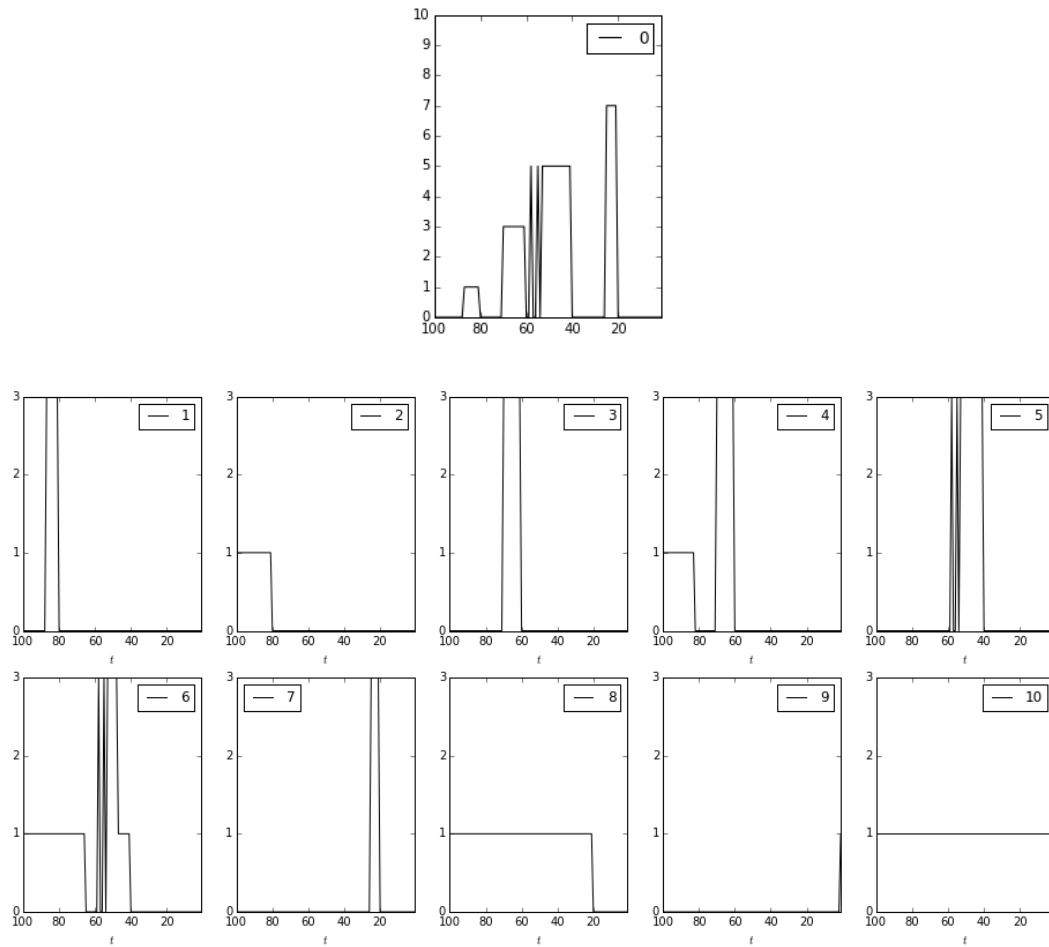


Abbildung 18 Veränderung der optimalen Politik für das Szenario 3 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 10)

Szenario 4

Für das Szenario 4 wird eine Cauchyverteilung für die Eintrittswahrscheinlichkeiten $p_j(t)$ der Produkthanfragen $j \in \mathcal{J}$ angenommen. Damit wird eine Verteilung modelliert, die für eine jede Produkthanfrage j den Höhepunkt der Wahrscheinlichkeit des Eintreffens zur letztmöglichen Buchungsperiode t vorsieht, aber mit geringer Wahrscheinlichkeit über alle Buchungsperioden $t \in T$ möglich ist. D. h. die Wahrscheinlichkeit steigt zum Leistungserstellungszeitpunkt \hat{t} für eine jede Produkthanfrage j stark an. Abbildung 19 zeigt die beschriebene Verteilung anhand eines Kurvendiagramms. Auch in diesem Szenario ist die Verteilung eines Produkts $j \in \mathcal{J}$ durch einen jeweiligen Skalar verstärkt, damit die Eintrittswahrscheinlichkeiten $p_j(t)$ größere Werte aufweisen. Die weiteren Eigenschaften des Szenarios

Tabelle 13 Szenario 4

	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	Cauchyverteilt $5 \cdot \mathcal{C}(81, 5)$
2	1000	1	1	Cauchyverteilt $5 \cdot \mathcal{C}(81, 5)$
3	100	1	2	Cauchyverteilt $5 \cdot \mathcal{C}(61, 5)$
4	1000	1	2	Cauchyverteilt $5 \cdot \mathcal{C}(61, 5)$
5	100	1	3	Cauchyverteilt $5 \cdot \mathcal{C}(41, 5)$
6	1000	1	3	Cauchyverteilt $5 \cdot \mathcal{C}(41, 5)$
7	100	1	4	Cauchyverteilt $5 \cdot \mathcal{C}(21, 5)$
8	1000	1	4	Cauchyverteilt $5 \cdot \mathcal{C}(21, 5)$
9	100	1	5	Cauchyverteilt $5 \cdot \mathcal{C}(1, 5)$
10	1000	1	5	Cauchyverteilt $5 \cdot \mathcal{C}(1, 5)$

Rechenzeit Systemzustände (h):	0:00:00.156592
Anzahl möglicher Systemzustände:	785376
Anzahl benötigter Systemzustände:	35324
Rechenzeit DP (h):	6:04:54.450184
Max. Erwartungswert (GE):	4918.91

orientieren sich an den bisherigen Szenarien. Tabelle 13 zeigt die Parameter zusammenfassend.

Die Auswertung in der Tabelle 14 zeigt für die Cauchyverteilung, dass die Anzahl der optimalen Politik d_j für Produkthanfragen $j \in \mathcal{J}$ zur möglichen Lagerproduktion keine besonderen Ausprägungen hat. Die Verteilung zeichnet sich dadurch aus, dass die Wahrscheinlichkeiten $p_j(t)$ über dem Wert 0 liegen und somit über den gesamten Buchungshorizont T eine Anfrage j möglich ist. Die optimale Politik für ertragreiche Produkthanfragen ist tendenziell eher die Akzeptanz einer Produkthanfrage j , was anhand des Vergleichs der Prozentwerte der optimale Politik der Annahme und der Lagerproduktion deutlich wird.

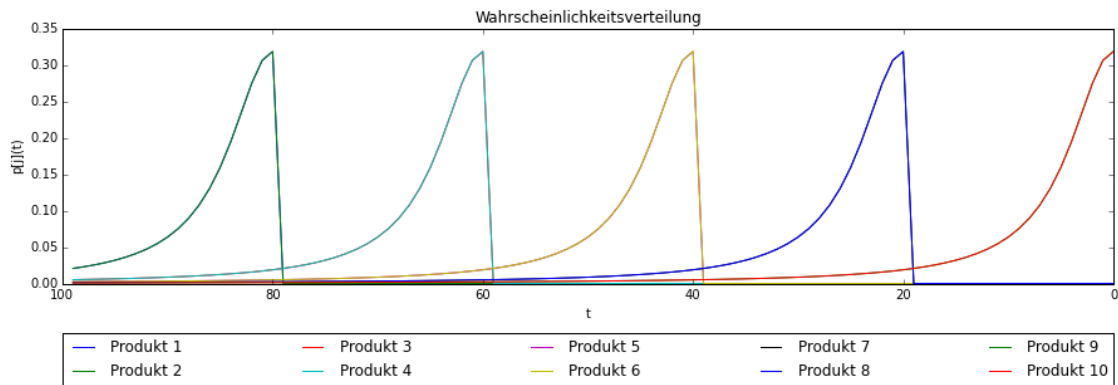


Abbildung 19 Wahrscheinlichkeitsverteilung Szenario 4

Tabelle 14 Auswertung des Szenarios 4

j	0	1	2	3	4	5	6	7	8	9	10
d_0	28073	247	0	1192	0	2151	0	3297	0	0	0
d_0 (%)	80.3	0.71	0	3.41	0	6.15	0	9.43	0	0	0

d_j	Ablehnung (0)		Annahme (1)		Lagerentnahme (2)		Lagerproduktion (3)	
	Anz.	%	Anz.	%	Anz.	%	Anz.	%
d_1	34693	99.24	20	0.06	0	0	247	0.7
d_2	33464	95.72	1463	4.17	0	0	33	0.09
d_3	33741	96.51	22	0.06	0	0	1197	3.41
d_4	29232	83.62	4007	11.43	1681	4.8	40	0.11
d_5	32689	93.5	22	0.06	0	0	2249	6.42
d_6	23830	68.16	7100	20.27	3991	11.39	39	0.11
d_7	31519	90.16	12	0.03	0	0	3429	9.78
d_8	17252	49.35	10408	29.73	7101	20.27	199	0.57
d_9	34506	98.7	332	0.95	122	0.35	0	0
d_{10}	8501	24.32	16803	48.03	9656	27.56	0	0

Die Veränderung des Verlaufs der optimalen Politik für das Szenario 4 zeigt die Abbildung 20. Bzgl. der optimalen Politik d_0 lässt sich entnehmen, dass eine Lager-

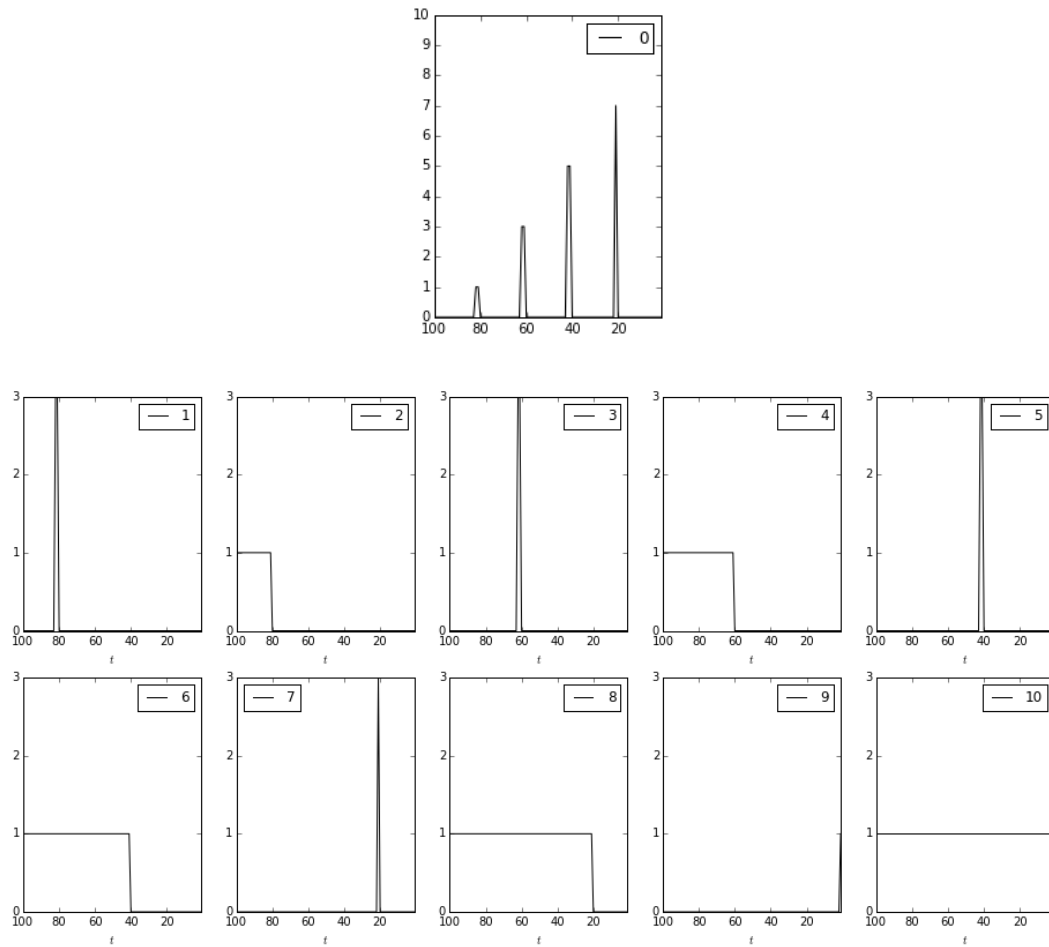


Abbildung 20 Veränderung der optimalen Politik für das Szenario 4 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 10)

produktion für die ertragsarmen Produkte kurz vor Verfall ermittelt ist und somit kurz bevor die Leistung in Produktion gehen kann. Für alle Produkthanfragen $j \in \mathcal{J}$ mit hohem Ertrag r_j ist die optimale Politik $d_j = 1$ und damit die Auftragsannahme dieser Produkthanfragen. Diese optimale Politik gilt bis zum produktspezifischen Leistungserstellungszeitpunkt \hat{t} . Für die ertragsarmen Anfragen $j \in \mathcal{J}$ ist die optimale Politik $d_j = 3$ jeweils kurz vor dem jeweiligen Leistungserstellungszeitpunkt \hat{t} . Damit wird die Kapazität für nachfolgende Anfragen über das Lager gesichert. Diese Politik gilt jedoch nicht für die ertragsarme Produkthanfrage $j = 9$, da für diese eine Lagerproduktion aufgrund des auslaufenden Buchungshorizonts T nicht möglich ist. Kurz vor Anlauf des Leistungserstellungszeitpunkt \hat{t} ist für diese Produkthanfrage die Annahme die optimale Politik d_9 .

Szenario 5

Für die nachfolgenden Szenarien wird die Anzahl der Produkthanfragen $j \in \mathcal{J}$ auf fünf mögliche Anfragen reduziert. Diese sind jeweils für einen bestimmten Leistungserstellungszeitpunkt \hat{t} vorgesehen. Die Leistungserstellungszeitpunkte \hat{t} erstrecken sich weiterhin gleichmäßig über den gesamten Buchungshorizont T . Für das Szenario 5 existieren nur diese fünf Produkthanfragen. Der Ertrag r_j für die Produkthanfragen $j \in \mathcal{J}$ ist dabei gestaffelt. Der Ertrag r_j der Anfrage $j = 1$ beträgt 200 GE und für die weiteren Produkthanfragen steigt der Ertrag jeweils um 200 GE. D. h. für die letzte Produkthanfrage $j = 5$ wird ein Ertrag in Höhe von $r_5 = 1000$ erzielt. Bei dem Szenario 5 wird wieder eine linear aufsteigende Wahrscheinlichkeitsfunktion $p_j(t)$ für eine jede Produkthanfrage j verwendet. Tabelle 15 zeigt die verwendeten Parameter und die Abbildung 21 zeigt die Eintrittswahrscheinlichkeiten der Anfragen für das Szenario 5.

Tabelle 15 Szenario 5

T	\hat{T}	h	$c_h^{\hat{t}} \forall \hat{t} \in \hat{T}$	$y_h^{\hat{t}} \forall \hat{t} \in \hat{T}$	$y_h^{\hat{t}, max} \forall \hat{t} \in \hat{T}$
100	5	1	1	0	2
j	r_j	a_{1j}	\hat{t}	Verteilung $p_j(t)$	
1	200	1	1	Linear ansteigend auf $p_j(t) = 0,3$ von 100 bis 81	
2	400	1	2	Linear ansteigend auf $p_j(t) = 0,3$ von 100 bis 61	
3	600	1	3	Linear ansteigend auf $p_j(t) = 0,3$ von 100 bis 41	
4	800	1	4	Linear ansteigend auf $p_j(t) = 0,3$ von 80 bis 21	
5	1000	1	5	Linear ansteigend auf $p_j(t) = 0,3$ von 40 bis 1	
Rechenzeit Systemzustände (h):				0:00:00.111966	
Anzahl möglicher Systemzustände:				785376	
Anzahl benötigter Systemzustände:				17667	
Rechenzeit DP (h):				1:01:44.564707	
Max. Erwartungswert (GE):				4606.34	

In diesem Szenario zeigt sich, dass die optimale Politik die Annahme der ertragreichen Anfrage $j = 5$ ist, sofern diese eintrifft ($d_5 = 1$). Für die anderen Anfragen

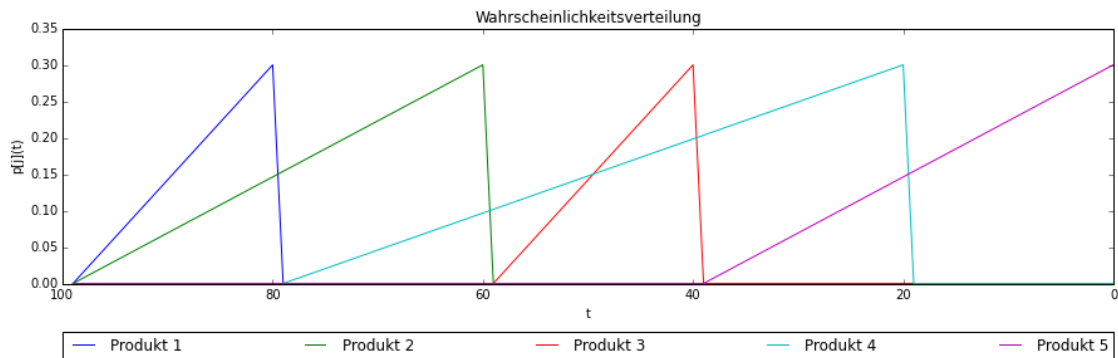


Abbildung 21 Wahrscheinlichkeitsverteilung Szenario 5

Tabelle 16 Auswertung des Szenarios 5

j	0	1	2	3	4	5
d_0	14480	24	109	398	2292	0
d_0 (%)	83.68	0.14	0.63	2.3	13.25	0

d_j	Ablehnung (0)		Annahme (1)		Lagerentnahme (2)		Lagerproduktion (3)	
	Anz.	%	Anz.	%	Anz.	%	Anz.	%
d_1	17279	99.86	0	0	0	0	24	0.14
d_2	17189	99.34	5	0.03	0	0	109	0.63
d_3	16870	97.5	35	0.2	0	0	398	2.29
d_4	14639	84.6	78	0.45	294	1.69	2292	13.18
d_5	6713	38.8	6922	39.92	3668	21.1	0	0

$j < 5$ gilt eher die optimale Politik der Lagerproduktion anstelle der Auftragsannahme. Durch eine Lagerproduktion wäre die erneute Annahme der Produkthanfrage $j = 5$ im Betrachtungszeitraum möglich. Sofern keine Anfragen eintreffen (d_0), wird tendenziell zum Ende des Buchungshorizonts T die optimale Politik der Lagerproduktion vom Modell bestimmt.

Abbildung 22 zeigt die Veränderung der optimalen Politik für das Szenario 5 für den vordefinierten Verlauf der Auftragseingänge. Bei diesem und den nachfolgen-

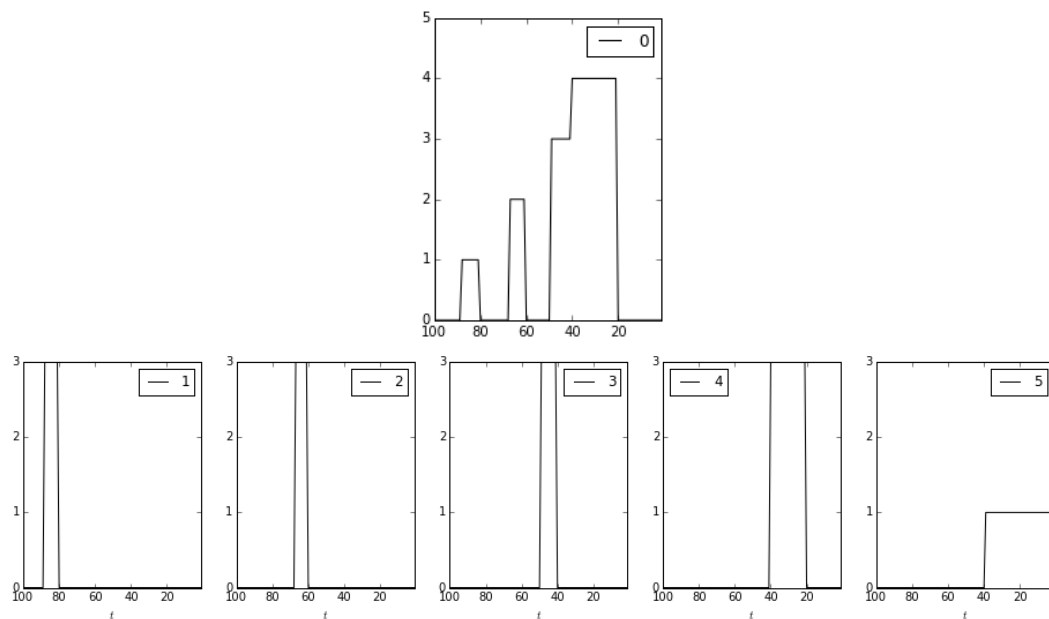


Abbildung 22 Veränderung der optimalen Politik für das Szenario 5 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 5)

den Szenarien muss für eine jeweilige Produkthanfrage j beachtet werden, dass die Anzahl der Produkte reduziert ist und die Anfragen zwar um die gleiche Ressource konkurrieren, aber keine gleich verteilte Produkthanfrage mit anderem Ertragswert existiert. Für die optimale Politik d_0 wird vor dem jeweiligen Leistungserstellungszeitpunkt \hat{t} die Entscheidung der Lagerproduktion ermittelt. Mit Ausnahme der Produkthanfrage $j = 5$, für die weiterhin keine Lagerproduktion möglich ist. Für die Produkthanfrage $j = 4$ wird wiederum für den gesamten Zeitraum $40 \geq \hat{t} \geq 21$ die optimale Politik der Lagerproduktion ermittelt. Bei der optimalen Politik d_j wird für Produkthanfragen $j < 5$ jeweils die Lagerproduktion vom Modell bestimmt. Diese optimale Politik kommt zustande, da die Produkthanfrage $j = 5$ den höchsten Ertrag r_j erzielt. Sofern ein Lagerbestand $y^{\hat{t}}$ verfügbar ist, kann die Akzeptanz dieser ertragreichen Produkthanfrage $j = 5$ für den Buchungsabschnitt $40 \geq \hat{t} \geq 1$ mit einer höheren Wahrscheinlichkeit erfolgen. Damit ist die optimale Politik d_5 die Annahme der Produkthanfrage über die gesamte Wahrscheinlichkeitsverteilung $p_5(t)$.

Szenario 6

Das Szenario 6 zeichnet sich durch eine normalverteilte Wahrscheinlichkeitsverteilung $p_j(t)$ aus. Dabei weist die Anfrage $j = 1$ einen Erwartungswert von $\mu = t = 90$

und eine Standardabweichung von $\sigma = 1$ aus. Für die weiteren Produkthanfragen ist der Erwartungswert μ jeweils um $t = 20$ Perioden verschoben. Die Standardabweichung σ variiert jedoch je Produkthanfrage j und wird zur Normierung über den gesamten Buchungshorizont T mit einem Skalar multipliziert. Damit besitzt eine jeweilige Verteilung $p_j(t)$ höhere Wahrscheinlichkeitswerte über den gesamten Betrachtungszeitraum. Abbildung 23 zeigt diese Verteilung. Die fünf möglichen Produkthanfragen $j \in \mathcal{J}$ haben für die anderen Parameter die Werte des Szenarios 5 hinterlegt. Tabelle 17 zeigt die Parameter für das Szenario 6.

Tabelle 17 Szenario 6

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	200	1	1	Normalverteilt $\mathcal{N}(90, 1)$	
2	400	1	2	Normalverteilt $10 \cdot \mathcal{N}(70, 10)$	
3	600	1	3	Normalverteilt $10 \cdot \mathcal{N}(50, 10)$	
4	800	1	4	Normalverteilt $10 \cdot \mathcal{N}(30, 10)$	
5	1000	1	5	Normalverteilt $30 \cdot \mathcal{N}(10, 30)$	
Rechenzeit Systemzustände (h):				0:00:03.354823	
Anzahl möglicher Systemzustände:				785376	
Anzahl benötigter Systemzustände:				35324	
Rechenzeit DP (h):				3:49:23.317486	
Max. Erwartungswert (GE):				4690.53	

Tabelle 18 zeigt die Auswertung für das Szenario 6. Unter Beachtung einer solchen Verteilung der Eintrittswahrscheinlichkeit ist die optimale Politik d_0 bzgl. einer Lagerproduktion für die Buchungsperioden $t \in T$ relativ ähnlich untereinander verteilt. Die Annahme von Produkthanfragen $j < 5$ scheint für das Modell nicht zur optimalen Politik zu gehören, wobei die Annahme der Produkthanfrage $j = 5$ relativ häufig zur optimalen Politik gewählt ist. Das Modell sieht eher für die Produkthanfragen $j < 5$ die Lagerproduktion vor, als die Annahme der Anfragen mit tendenziell niedrigerem Ertrag.

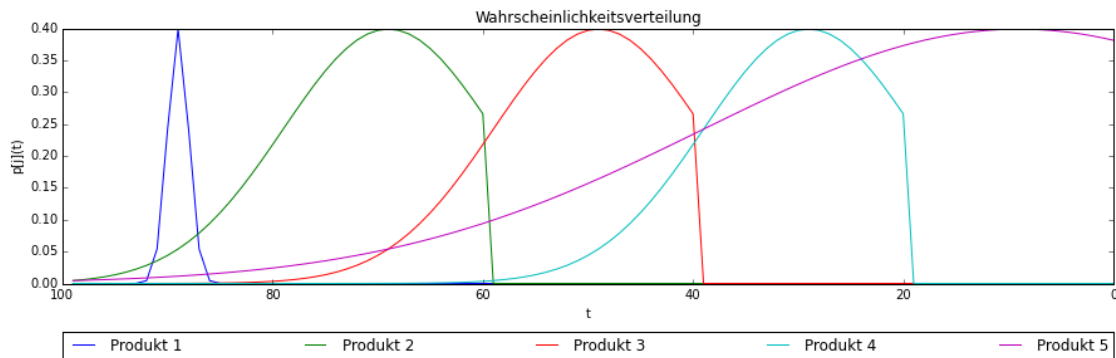


Abbildung 23 Wahrscheinlichkeitsverteilung Szenario 6

Tabelle 18 Auswertung des Szenarios 6

j	0	1	2	3	4	5
d_0	29249	701	1040	1476	2494	0
d_0 (%)	83.66	2.01	2.97	4.22	7.13	0

d_j	Ablehnung (0)		Annahme (1)		Lagerentnahme (2)		Lagerproduktion (3)	
	Anz.	%	Anz.	%	Anz.	%	Anz.	%
d_1	33949	97.11	310	0.88	0	0	701	2
d_2	33749	96.54	159	0.45	0	0	1052	3
d_3	32632	93.34	211	0.6	516	1.47	1601	4.57
d_4	30601	87.53	48	0.14	1619	4.62	2692	7.68
d_5	9223	26.38	15587	44.55	10150	28.97	0	0

Wird die Abbildung 24 betrachtet, dann zeigt sich für das Szenario 6 eine ähnliche Veränderung der optimalen Politik wie im Szenario 3. Beide Szenarien haben eine Normalverteilung unterstellt und weisen bei den Verläufen der optimalen Politik je Anfrage einen punktuellen Wechsel zwischen den Entscheidungsalternativen. Die grundsätzliche Strategie scheint unter Verwendung der ertragsarmen Produkthanfragen die „Lagerproduktion“ zu sein. D. h. die optimale Politik ist es, die Kapazitäten vor dem Leistungserstellungszeitpunkt zum Lagerbestand zu transformieren, sofern keine Produkthanfragen eintreffen und sofern eine jeweilige Produkthanfrage $j \in \mathcal{J}$

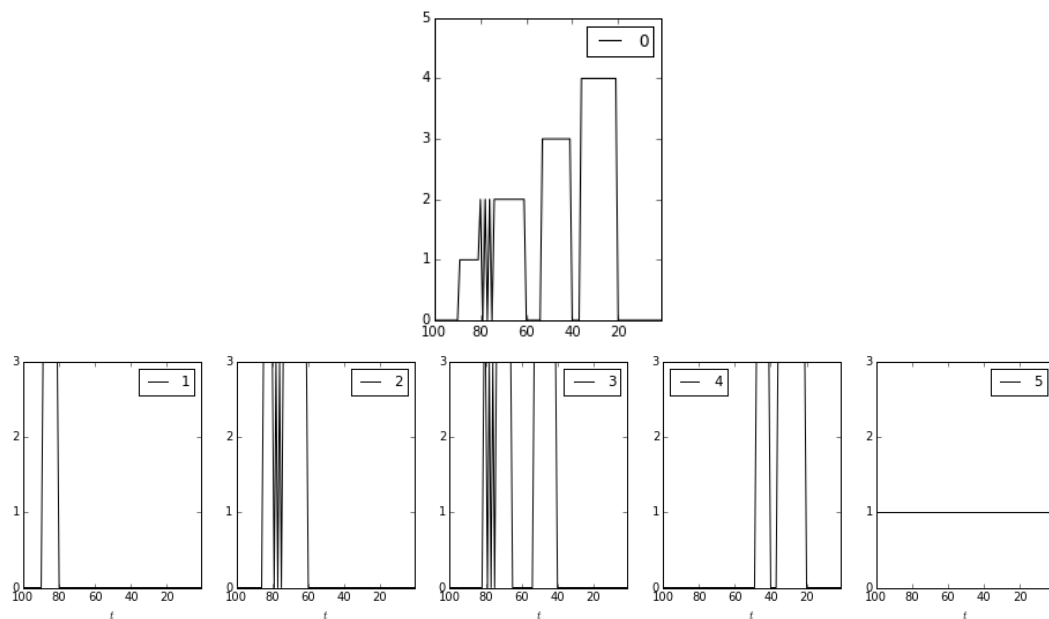


Abbildung 24 Veränderung der optimalen Politik für das Szenario 6 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produktanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 5)

eintrifft. Damit steigt die Wahrscheinlichkeit im weiteren Zeitverlauf die ertragreiche Produktanfrage $j = 5$ akzeptieren zu können. Dies erhöht den möglichen Gesamtertrag des Unternehmens. Die optimale Politik für Produktanfragen $j < 5$ schwankt dabei aufgrund der Normalverteilung zwischen der „Lagerproduktion“ und der „Ablehnung des Auftrags“ in gewissen Buchungsabschnitten. Bspw. zeigt dies der Verlauf der optimalen Politik d_3 für Anfragen $j = 3$. Im Umfeld der Buchungsperiode $t = 80$ wechseln die Entscheidungsalternativen untereinander mehrfach und im Bereich der Buchungsperiode $t = 60$ ist über einen längeren Buchungsabschnitt die „Ablehnung des Auftrags“ die optimale Politik. Wird die Verteilung des Szenarios in Abbildung 23 betrachtet, dann kann dieser Verlauf der optimalen Politik erklärt werden. Im Buchungsabschnitt um die Periode $t = 80$ hat die Produktanfrage $j = 2$ ihren höchsten Erwartungswert und im Buchungsabschnitt um die Periode $t = 60$ tritt erstmalig die Wahrscheinlichkeit einer Produktanfrage $j = 4$ ein. Wird dann zusätzlich noch die ansteigende Wahrscheinlichkeit der ertragreichen Produktanfrage $j = 5$ herangezogen, dann wird der Ertragswert, sofern eine Produktanfrage $j = 3$ zur Lagerproduktion Verwendung finden soll, jeweils in diesen Buchungsabschnitten durch die Entscheidungsalternative überlagert.

Szenario 7

In dem Szenario 7 wird unterstellt, dass am Anfang relativ gleichwertige Anfragen existieren und das Eintreffen dieser Anfragen nochmalverteilt ist. Es gibt aber zusätzlich die Möglichkeit, die auch relativ wahrscheinlich ist, dass eine höherwertige Anfrage $j = 5$ eintrifft und einen Ertrag in Höhe von $r_j = 5000$ erzielt. Es soll mit dem Modell getestet werden, ob eine solche Anfrage die optimale Politik des Netzwerks beeinflusst. Tabelle 19 zeigt die verwendeten Parameter und die Abbildung 25 zeigt die Wahrscheinlichkeitsverteilung für das Szenario 7.

Tabelle 19 Szenario 7

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	200	1	1	Normalverteilt $10 \cdot \mathcal{N}(90, 10)$
2	400	1	2	Normalverteilt $10 \cdot \mathcal{N}(70, 10)$
3	200	1	3	Normalverteilt $10 \cdot \mathcal{N}(50, 10)$
4	300	1	4	Normalverteilt $10 \cdot \mathcal{N}(30, 30)$
5	5000	1	5	Normalverteilt $30 \cdot \mathcal{N}(10, 0.45)$

Rechenzeit Systemzustände (h):	0:00:00.097337
Anzahl möglicher Systemzustände:	785376
Anzahl benötigter Systemzustände:	16057
Rechenzeit DP (h):	1:02:49.057969
Max. Erwartungswert (GE):	6257.86

Die Auswertung des Szenarios 7 zeigt in Tabelle 20, dass unter eine solchen Verteilung die optimale Politik der Lagerproduktion von Produkthanfragen $j \in \mathcal{J}$ selten bestimmt ist, mit Ausnahme der optimalen Politik der Produkthanfrage $j = 3$. Für diese Produkthanfrage wird häufiger die optimale Politik der Lagerproduktion gewählt ($d_3 = 3$). Für die anderen Produkthanfragen sieht das Modell es eher weniger vor, dass ein Lagerbestand für spätere ertragreiche Produkthanfragen aufgebaut wird, sofern die Anzahl der optimalen Politik der Auftragsablehnung keine Beachtung findet. Für die optimale Politik d_0 wird ebenfalls eher die Produkthanfrage

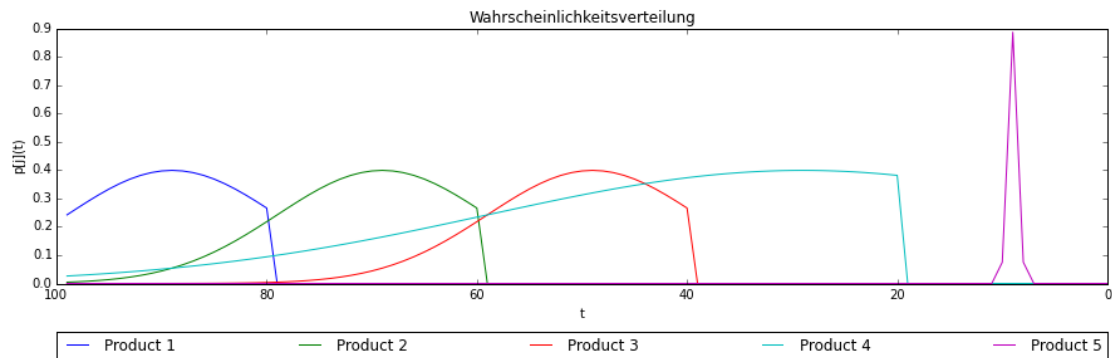


Abbildung 25 Wahrscheinlichkeitsverteilung Szenario 7

Tabelle 20 Auswertung des Szenarios 7

j	0	1	2	3	4	5
d_0	14087	292	259	914	141	0
d_0 (%)	89.77	1.86	1.65	5.82	0.9	0

d_j	Ablehnung (0)		Annahme (1)		Lagerentnahme (2)		Lagerproduktion (3)	
	Anz.	%	Anz.	%	Anz.	%	Anz.	%
d_1	15371	97.95	30	0.19	0	0	292	1.85
d_2	13644	86.94	1403	8.89	586	3.71	60	0.38
d_3	14690	93.61	21	0.13	66	0.42	916	5.8
d_4	12400	79.02	1695	10.75	1554	9.85	44	0.28
d_5	12279	78.25	2298	14.57	1116	7.07	0	0

$j = 3$ für die Lagerproduktion verwendet.

Abbildung 26 zeigt für das Szenario 7 den Verlauf der optimalen Politik d_0 und $d_j \forall j \in \mathcal{J}$. Für die optimale Politik d_0 zeigt sich, dass die relativ ertragreichen Produkthanfragen $j \in \mathcal{J}$ kurz vor dem Leistungserstellungszeitpunkt \hat{t} dem Lager ergänzt werden. D. h. eine Entscheidung, die Kapazität für die Produkthanfragen $j = 2$ und $j = 4$ für das Lager zu verwenden und dementsprechend die aus einer

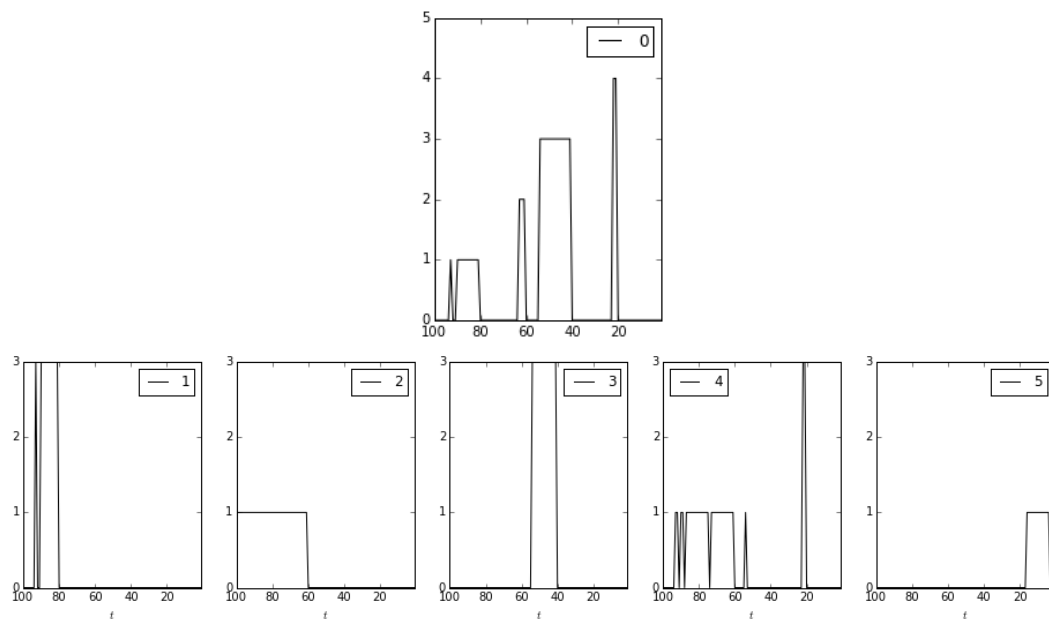


Abbildung 26 Veränderung der optimalen Politik für das Szenario 7 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 5)

Annahme resultierenden Erträge r_j zu verlieren, ist erst kurz vor Verderben der Kapazität $c^{\hat{t}}$ optimal. Bei der optimalen Politik d_j ergibt sich ein differenziertes Bild. Für die ertragsarmen Produkthanfragen $j = 1$ und $j = 3$ ist die Lagerproduktion hauptsächlich die optimale Politik. Bei Anfragen $j = 2$ ist die Auftragsannahme die optimale Politik d_2 bis zur produktspezifischen Leistungserstellung \hat{t} . Sofern im späteren Verlauf die sehr ertragreiche Produkthanfrage $j = 5$ eintrifft, ist die optimale Politik die Annahme einer solchen Anfrage. Die optimale Politik d_j für eine Produkthanfrage $j = 4$ ist hauptsächlich die Annahme der Anfrage. Treffen jedoch keine Anfragen ein, dann ist die optimale Politik kurz vor Verlust der Möglichkeit die Ablehnung einer eintreffenden Anfrage inkl. der Lagerproduktionsentscheidung. Damit steigt die Wahrscheinlichkeit, nicht nur einen Ertrag von $r_4 = 300$ zu erzielen, sondern einen Ertrag in Höhe von $r_5 = 5000$.

Szenario 8

Beim Szenario 8 wird eine veränderte Normalverteilung bei den Eintrittswahrscheinlichkeiten $p_j(t)$ der Produkthanfragen $j \in \mathcal{J}$ angenommen. Die Wahrscheinlichkeit des Eintreffens der Produkthanfragen ist am Anfang des Buchungshorizonts am größten. D. h. der Erwartungswert μ einer Produkthanfrage j liegt eher in den anfänglichen Buchungsperioden t . Abbildung 27 zeigt den Kurvenverlauf der Ein-

trittswahrscheinlichkeiten $p_j(t)$ der Produkthanfragen $j \in \mathcal{J}$. Die Erträge r_j der Produkthanfragen $j \in \mathcal{J}$ ist bei dem Szenario 5 in der Form verteilt, dass die ersten drei Produkthanfragen j einen eher niedrigen Ertrag und die letzten zwei Produkthanfragen j einen eher hohen Ertrag erzielen. Tabelle 21 zeigt die Parameter für das Szenario 8.

Tabelle 21 Szenario 8

	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	200	1	1	Normalverteilt $10 \cdot \mathcal{N}(90, 10)$
2	300	1	2	Normalverteilt $10 \cdot \mathcal{N}(80, 10)$
3	800	1	3	Normalverteilt $10 \cdot \mathcal{N}(70, 10)$
4	2500	1	4	Normalverteilt $10 \cdot \mathcal{N}(50, 30)$
5	5000	1	5	Normalverteilt $10 \cdot \mathcal{N}(20, 10)$

Rechenzeit Systemzustände (h):	0:00:00.133609
Anzahl möglicher Systemzustände:	785376
Anzahl benötigter Systemzustände:	35324
Rechenzeit DP (h):	2:54:15.295931
Max. Erwartungswert (GE):	22428.8

Auch die Auswertung des Szenarios 8 in Tabelle 22 zeigt, dass relativ selten die Annahme von Produkthanfragen $j < 5$ als optimale Politik vom Modell gewählt ist. Für die Produkthanfrage $j = 5$ wird tendenziell häufig die optimale Politik der Auftragsannahme vom Modell bestimmt. Im Vergleich zum Szenario 7 stieg die Anzahl der optimalen Politiken bzgl. der Lagerproduktion auf ein Niveau der anderen Szenarien. Sofern keine Anfragen eintreffen, zeigt sich bei der optimalen Politik d_0 , dass mit ablaufenden Buchungshorizont T die Anzahl der optimalen Politik bzgl. der Lagerproduktion steigt.

Unter Beachtung der Eintrittswahrscheinlichkeit $p_j(t)$ des Szenarios 7 ergibt sich der in der Abbildung 28 aufgezeigte Verlauf der optimalen Politik, sofern keine Anfragen akzeptiert werden. Der Verlauf zeigt, dass für die optimale Politik d_0 je-

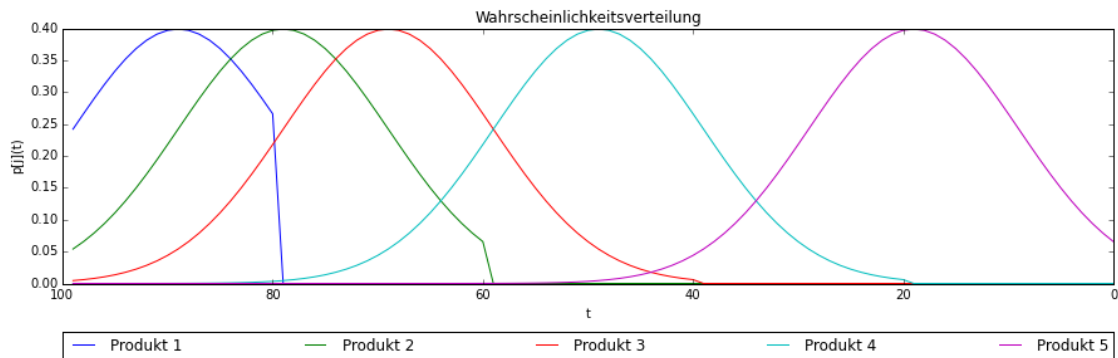


Abbildung 27 Wahrscheinlichkeitsverteilung Szenario 8

Tabelle 22 Auswertung des Szenarios 8

j	0	1	2	3	4	5
d_0	26733	1069	2466	1712	2980	0
d_0 (%)	76.47	3.06	7.05	4.9	8.52	0

d_j	Ablehnung (0)		Annahme (1)		Lagerentnahme (2)		Lagerproduktion (3)	
	Anz.	%	Anz.	%	Anz.	%	Anz.	%
d_1	33713	96.43	176	0.5	0	0	1071	3.06
d_2	32271	92.31	147	0.42	0	0	2542	7.25
d_3	32923	94.17	171	0.49	154	0.44	1712	4.88
d_4	30042	85.93	48	0.14	1584	4.52	3286	9.38
d_5	8557	24.48	15990	45.71	10413	29.73	0	0

weils die im Buchungshorizont T nachfolgende Produkthanfrage j für eine mögliche Lagerproduktion verwendet wird. Auch für Produkthanfragen $j \in \mathcal{J}$ ergibt sich dieses Bild. Die ertragsarmen Produkthanfragen $j < 5$ werden für die Lagerproduktion verwendet, sofern diese eintreffen. Selbst für die Produkthanfrage $j = 4$ wird eine Lagerproduktion als optimale Politik ermittelt, obwohl diese Produkthanfrage einen relativ hohen Ertrag $r_4 = 2500$ erzielen würde.

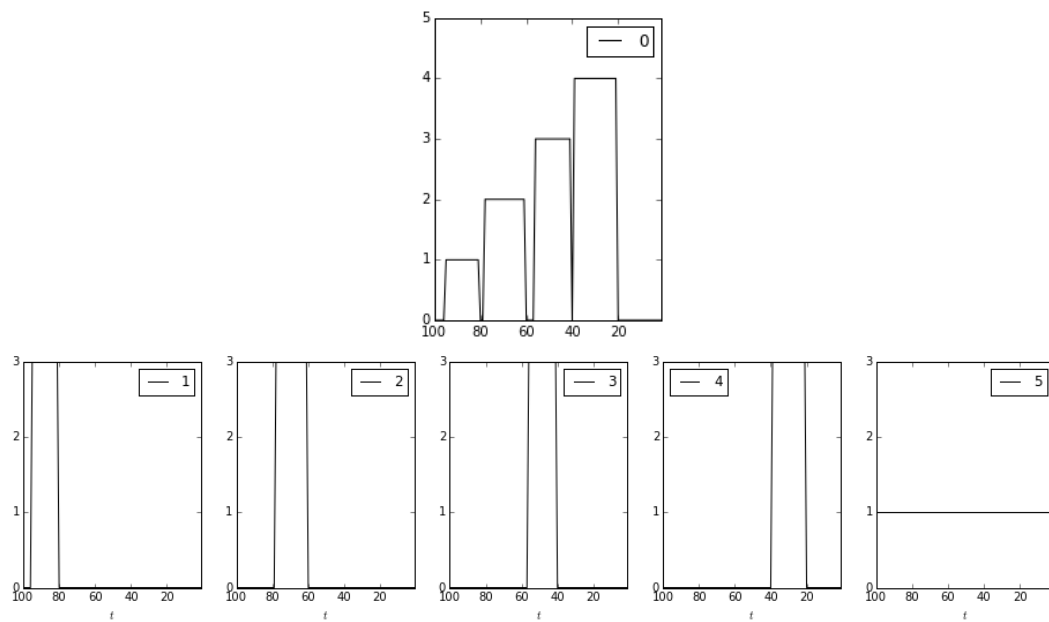


Abbildung 28 Veränderung der optimalen Politik für das Szenario 8 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produktanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 5)

Szenario 9

Im Szenario 9 weisen die Produktanfragen $j \in \mathcal{J}$ eine Cauchyverteilung auf, die den höchsten Wahrscheinlichkeitswert $p_j(t)$ in der letztmöglichen Buchungsperiode t besitzt. Damit wird unterstellt, dass die Nachfrage in Annäherung des Leistungserstellungszeitpunkts \hat{t} einer Produktanfrage j stark ansteigt. Die Erträge r_j sind für dieses Szenario um 200 GE gestaffelt und verlaufen von $r_1 = 200$ bis $r_5 = 1000$. Die Parameter des Szenarios sind in der Tabelle 23 aufgeführt und die Abbildung 29 zeigt die Wahrscheinlichkeitsverteilung für die Produktanfragen $j \in \mathcal{J}$.

Die Auswertung des Szenarios zeigt, dass erneut die Produktanfrage $j = 5$ mit einem hohen Ertrag r_j relativ häufig als optimalen Politik der Auftragsannahme vom Modell gewählt wird. Eine Lagerhaltungsentscheidung wird anders als bei den vorherigen Szenario häufiger als optimale Politik ermittelt. Unter der Annahme einer solchen Verteilung wird öfters eine solche Politik neben der Politik der Auftragsannahme aufgezeigt. Sofern keine Anfragen eintreffen, wird ebenfalls eine Steigung der optimalen Politik d_o für eine mögliche Lagerproduktion festgestellt. Tabelle 24 zeigt die komplette Auswertung des Szenarios 9.

Tabelle 23 Szenario 9

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	200	1	1	Cauchyverteilt $5 \cdot \mathcal{C}(81, 5)$
2	400	1	2	Cauchyverteilt $5 \cdot \mathcal{C}(61, 5)$
3	600	1	3	Cauchyverteilt $5 \cdot \mathcal{C}(41, 5)$
4	800	1	4	Cauchyverteilt $5 \cdot \mathcal{C}(21, 5)$
5	1000	1	5	Cauchyverteilt $5 \cdot \mathcal{C}(1, 5)$

Rechenzeit Systemzustände (h):	0:00:00.117737
Anzahl möglicher Systemzustände:	785376
Anzahl benötigter Systemzustände:	35324
Rechenzeit DP (h):	3:04:07.716158
Max. Erwartungswert (GE):	3809.91

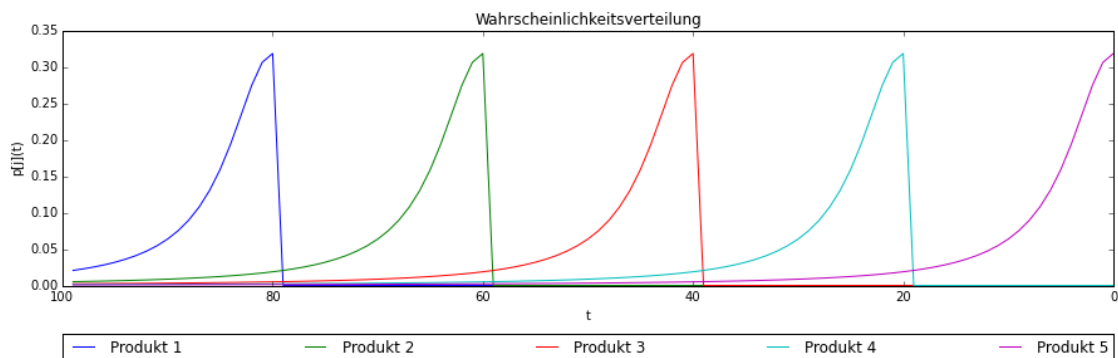


Abbildung 29 Wahrscheinlichkeitsverteilung Szenario 9

Die optimale Politik für dieses Szenario zeigt einen besonderen Verlauf für d_0 und d_j . Für die optimale Politik d_0 , bei der keine Anfragen eintreffen und damit die Wahrscheinlichkeitsverteilung $p_0(t)$ gilt, wechselt die für die Lagerproduktion verwendeten Ausführungsmodi für die Produkthanfragen $j \in \mathcal{J}$ in den Buchungs-

Tabelle 24 Auswertung des Szenarios 9

j	0	1	2	3	4	5
d_0	23029	1037	2370	3981	4543	0
d_0 (%)	65.87	2.97	6.78	11.39	12.99	0

d_j	Ablehnung (0)		Annahme (1)		Lagerentnahme (2)		Lagerproduktion (3)	
	Anz.	%	Anz.	%	Anz.	%	Anz.	%
d_1	33604	96.12	285	0.81	0	0	1071	3.06
d_2	31847	91.1	447	1.28	14	0.04	2652	7.57
d_3	28318	81	1438	4.1	616	1.76	4588	13.09
d_4	21140	60.47	4906	14.01	4073	11.63	4841	13.81
d_5	8392	24	15990	45.71	10578	30.2	0	0

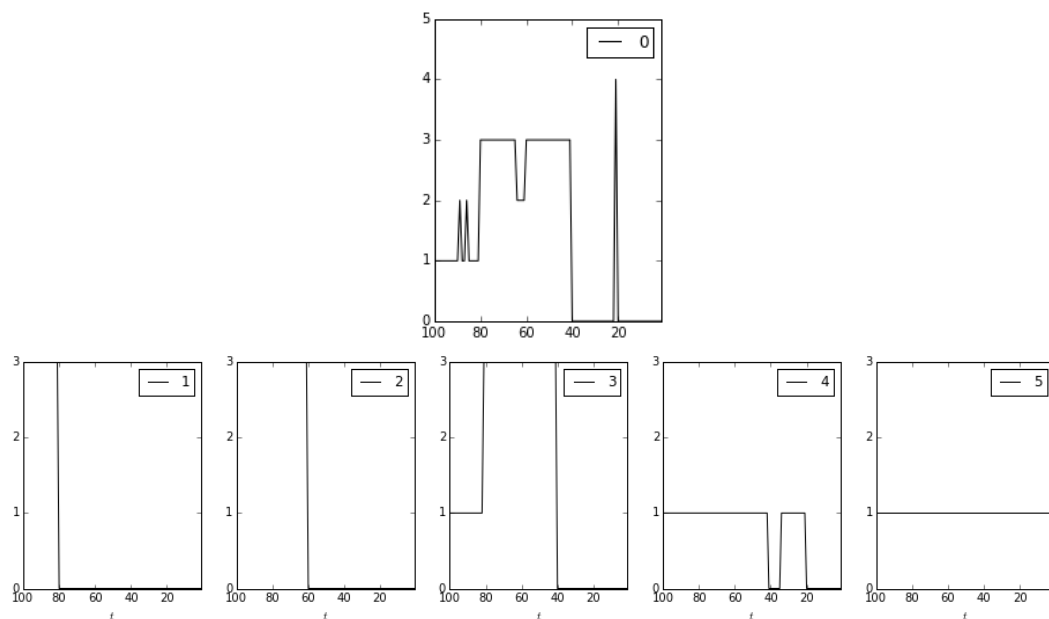


Abbildung 30 Veränderung der optimalen Politik für das Szenario 9 sofern keine Anfragen eintreffen ($d_0 = 0$) und je Produkthanfrage bei konstanter Ressourcenkapazität (d_j entspricht 1 bis 5)

perioden $t \in T$. Bspw. wechselt die optimale Politik zweimal im Buchungsabschnitt $100 \geq t \geq 81$ zwischen $d_0 = 1$ und $d_0 = 2$. Auch im Buchungsabschnitt $80 \geq t \geq 41$ wechselt in einem längeren Zeitraum die optimale Politik zwischen

$d_0 = 2$ und $d_0 = 3$. Im Buchungsabschnitt $40 \geq t \geq 21$ wird eine optimale Politik für die Lagerproduktion durch den Ausführungsmodus der Anfrage $j = 4$ nur kurz vor Verfall der Kapazität \mathbf{c}^t ermittelt. Für die Produkthanfragen $j = 1$ und $j = 2$ gilt ausschließlich die optimale Politik der Lagerproduktion über den gesamten Buchungshorizont T , sofern die Anfragen eine Eintrittswahrscheinlichkeit $p_j(t) > 0$ besitzen. Für eine Produkthanfrage $j = 3$ gilt die optimale Politik der Auftragsannahme für den Buchungsabschnitt $100 \geq t \geq 81$ und nachfolgend die Auftragsablehnung inkl. einer Lagerproduktion. Sofern eine Anfrage $j = 4$ eintrifft, ist die optimale Politik die Auftragsannahme über den möglichen Buchungshorizont T , wobei diese optimale Politik im Buchungsabschnitt $40 \geq t \geq 21$ kurz unterbrochen ist. In diesem Zeitraum ist die Ablehnung die optimale Politik d_4 . Für Produkthanfragen $j = 5$ gilt über den gesamten Buchungshorizont T die optimale Politik der Auftragsannahme ($d_5 = 1$).

7 Schlussbemerkung

Zusammenfassung

Die Arbeit zeigt, dass eine mögliche Lagerhaltung die Inanspruchnahme der Kapazität von endlichen Ressourcen verbessert und den erwarteten Ertrag erhöht. Die verbesserte Kapazitätsallokation kommt zustande, da die Ressourcenkapazitäten zur Erhöhung eines Lagerbestands Verwendung finden. Dies erfolgt indem Anfragen zur Instandsetzung von defekten Produkten mit niedrigem Ertrag abgelehnt und durch die Entscheidungen einer Lagerproduktion zur Erhöhung eines Lagerbestands von bereits reparierten Produkten ersetzt werden. Bei dem Lagerbestand handelt es sich um Kapazitäten, die mit hoher Wahrscheinlichkeit für spätere Anfragen mit höherem Ertragswert Beanspruchung finden, damit eine Maximierung des erwarteten Ertragswerts möglich wird. Durch die Verwendung der Kapazität erfolgt kein Verlust der endlichen Ressource, sondern die Sicherung dieser Kapazität für nachfolgende Produktanfragen in Form des Lagerbestands. Zusätzlich ermöglicht das Modell aus dieser Arbeit auch das Betrachten einer möglichen Lagerproduktion von neuwertigen Produkten sofern keine Anfrage eintrifft. Damit erhält ein Unternehmen für die Planung der Kapazitätsverwendung mehr Flexibilität und erhält dementsprechend eine umfangreichere Politik der möglichen Auftragsannahmen für den Betrachtungszeitraum.

Die Planung der optimalen Politik ist dabei erheblich abhängig von der Wahrscheinlichkeitsverteilung des Eintreffens und der möglichen Erträge der Produktanfragen. Dies resultiert aus der Gleichung (11). Für jede mögliche Entscheidungsalternative der Modellgleichung gibt es bestimmte OK die durch diese Parameter beeinflusst werden. Im Verlauf des Buchungshorizonts eines beispielhaften Netzwerks verändern sich die Wahrscheinlichkeiten des möglichen Eintreffens der Produktanfragen und damit auch die OK. Je wahrscheinlicher der Verfall einer Ressource wird, desto höher ist die Wahrscheinlichkeit das eine Lagerproduktion zur Sicherung der Kapazitätsbeanspruchung der Ressource den noch möglichen Ertrag maximiert. Der

mögliche Ertrag durch Annahme einer Anfrage abzgl. der OK ist nicht mehr ausreichend rentabel und das Modell empfiehlt für die Verwendung der Kapazität die Lagerproduktion. Damit ist im weiteren Verlauf ein höherer Ertrag für das Netzwerk möglich. Anders formuliert bedeutet dies, dass ab einem solchen Punkt bzw. ab einem solchen Systemzustand der Übergang in einen anderen Systemzustand mit einem vorhandenen Lagerbestand nicht gegen die Grenzbedingungen verstößt und im weiteren Verlauf einen höheren Ertragswert verspricht. Für ein solches Szenario existiert damit ein besserer Pfad im Netzwerk-Graphen, der potentiell mehr Ertrag verspricht.

Die numerische Untersuchung zeigt, dass die Anzahl der Ausprägungen der optimalen Politik mit der Wahrscheinlichkeitsverteilung der Produkthanfragen variiert. Es gab in der Untersuchung kein Szenario, dass eine höhere Anzahl der optimalen Politik der Lagerproduktion gegenüber der Auftragsannahme aufweist. Auch der Fall, dass die optimale Politik der Lagerproduktion die Auftragsannahme zumindest für eine bestimmte Art von Produkthanfragen dominiert konnte nicht gezeigt werden. Dieser Sachverhalt wird dadurch erklärt, dass für die optimale Politik der Auftragsannahme (und auch für die Lagerentnahme) die jeweiligen Erträge der Produkthanfragen bei dem Entscheidungsmodell Berücksichtigung finden. D. h. die Annahme der Produkthanfrage verbessert den maximalen Erwartungswert aufgrund der dadurch generierten Erträge. Die Entscheidung der Lagerproduktion ist bei diesen Modellannahme eine Möglichkeit die Kapazitäten für nachfolgende Produkthanfragen zu sichern und tritt damit vermehrt kurz vor dem Verfall der Kapazität auf. Sofern eine bestimmte Reihenfolge der Auftragseingänge unterstellt wird, kann die Untersuchung des Verlaufs der optimalen Politik erfolgen. Die Szenarien zeigen unterschiedliche Verläufe für die optimale Politik, sofern keine Anfragen eintreffen und sofern eine jeweilige Produkthanfrage eintrifft. In den Szenarien ist damit gezeigt, dass die Wahrscheinlichkeitsverteilung erheblichen Einfluss auf die optimale Politik hat und damit auf den Zeitpunkt an dem die Sicherung der Ressourcenkapazität beginnt.

Limitation

In dieser Arbeit wird die Lagerproduktion eines Produkts betrachtet, bei der die Instandsetzung von defekten Produkten aufgrund der Inanspruchnahme der Kapazität einer einzelnen Ressource erfolgt. Das Modell ermöglicht die Berücksichtigung

von unterschiedlichen Ressourcen, wie z. B. Maschinen- und Personaleinsatzstunden. Für die numerische Untersuchung ist die Vereinfachung auf nur eine Ressource jedoch zielführend. Damit war zum einen die schnelle Berechnung der Szenarien möglich und zum anderen ist dadurch die Funktionsweise des Modells im Grundsatz explorierbar. Jedoch sollte für spätere Analysen auch die Berechnung von umfangreicheren Szenarien mit unterschiedlichen Ressourcen zur Erstellung differenzierter Produkte in Betracht kommen.

In der Modellerweiterung (9) ist die Entscheidung der Lagerproduktion bei eintreffenden Anfragen nur für den jeweiligen Ausführungsmodi des Produkthanfrage möglich. Damit steht dem Netzwerk jeweils nur die Produktionskapazität einer abgelehnten Anfrage zur Verfügung. D. h. ein Unternehmen ist immer gezwungen die Ressourcenkapazität für die Lagererhöhung zu verwenden, die aktuell als Auftrag vorliegt. Damit geht jedoch viel Flexibilität verloren. Daher sollte bei der Modellformulierung (9) die Möglichkeit bestehen, dass sofern eine Anfrage abgelehnt wird, ein beliebiger Ausführungsmodi aller möglichen Produkthanfragen für die Lagerproduktion genutzt werden können. Damit wäre ebenfalls bei dieser Entscheidung über die Auftragsannahme einer eintreffenden Produkthanfrage das Maximum über alle Produkthanfragen zu bilden. Bei der in dieser Arbeit umgesetzten Modellerweiterung ist dieses nicht berücksichtigt.

Die Implementierung in das Computersystem bietet weiteres Optimierungspotential. Der Algorithmus ist zwar mit einigen Funktionen des Softwarepakets `Numpy` entwickelt, aber eine komplette Implementierung in die Programmiersprache `C++` oder `Java` könnte die Leistung weiter verbessern bzw. die Rechenzeit reduzieren. Sofern weiter an dieser Implementierung festgehalten wird, empfiehlt es sich die Memofunktion zu überarbeiten. In der aktuellen Fassung der Implementierung basiert die Memofunktion auf der Programmiersprache `Python`. Auch hier sollte ein Versuch getätigt werden, eine Memofunktion mit dem Softwarepaket `Numpy` zu entwickeln. Ggf. kann dadurch die vorherige Berechnung aller möglichen Systemzustände des Netzwerks entfallen. Diese Berechnung aller möglichen Systemzustände ist zwar ein sehr effizienter Algorithmus, aber die Funktion zur Berechnung hat einen hohen Hauptspeicherbedarf. Sofern die Berechnung von umfangreichere Szenarien erfolgen soll, muss diese Funktion parallelisiert werden. Dadurch erfolgt die Berechnung der Funktion auf mehreren Knoten (Computern) des Clustersystems unter Nutzung der jeweiligen Speicher. Die Umsetzung dieser Leistungsverbesserungen des Algorithmus konnte in dieser Arbeit nicht mehr erfolgen.

Weiter ist die verwendete Implementierung des DP in rekursiver Form programmiert. Ein iterativer Lösungsweg könnte die Leistung des Algorithmus verbessern, aber auch eine mögliche Parallelisierung der Teilprobleme des Auftragsannahmeproblems sollte weiter untersucht werden. In dem begrenzten Zeitraum der Erstellung dieser Arbeit erfolgte keine abschließende Umsetzung einer möglichen Parallelisierung der Problemstellung. Die Implementierung nutzt zwar viele aktuelle Funktionen in Bezug des wissenschaftlichen Rechnens, aber es ist der erste Versuch einer Implementierung des Auftragsannahmeproblems des Netzwerk RM unter Berücksichtigung einer Lagerhaltung. Es besteht daher Potential zur Verbesserung der Implementierung.

Ausblick

Der Schwerpunkt weiter Forschung in Bezug des Auftragsannahmeproblems im Netzwerk RM mit Lagerhaltungsentscheidung sollte die Konkretisierung der Problemstellung sein. Bei der hier gezeigten Modellerweiterung handelt es sich um eine einfache Umformung des Grundmodells des Netzwerk RM (siehe Gleichung (1) im Vergleich zu (9)). Es berücksichtigt z. B. keine Lagerkosten oder eine zeitgebundene Erhöhung von Ressourcen. Daher sollte nachfolgend ein Modell entwickelt werden, welches aus einem tatsächlichen Anwendungsfall resultiert und die Problemstellung der auftragsbezogenen Instandhaltungsprozesse umfangreicher abbildet.

Jedoch zeigt die aktuelle Entwicklung im Forschungsbereich des Netzwerk RM in der Auftragsfertigung, dass ein exaktes Lösungsverfahren keine realen Probleme in einer angemessenen Berechnungszeit löst. Die Integration eines Verfahrens der Approximation der Problemstellung sollte daher ebenfalls vorangetrieben werden. Durch Integration von Verfahren, wie z. B. dem Verfahren des Bid-Preises, reduziert die Berechnungszeit und ermöglicht die Übertragung in moderne Anwendungssystemen für das RM. Ebenfalls können die Ansätze aus dem Literaturüberblick aus Kapitel 4 auf mögliche Umsetzung untersucht werden. Vielversprechend erscheint die Ermittlung des Bid-Preises anhand eines Knapsack-Verfahrens.

Anhang

DP-Implementierung des Auftragsannahmeproblems

Quellcode 1 Definition der Parameter für die Dynamische Programmierung im Netzwerk RM

```
# coding: utf-8

# In [1]:

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

# In [2]:

import numpy as np
import datetime

# #Parameter

# ## Parameter für das Grundmodell
# ###Produktanfrage
# Eine Produkthanfrage  $j$  aus der Menge an Produkthanfragen  $\mathcal{J}$ :

# In [3]:

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

```
# ### Ressourcen
#
# Eine Ressourcen  $h$  aus der Menge an Ressourcen  $\mathcal{H}$ :

# In[4]:

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

# ### Kapazität der Ressourcen
#
# Für jeder Ressource  $h \in \mathcal{H}$  gibt es eine  $\leftarrow$ 
# vordefinierte Kapazität  $c_h$ :

# In[5]:

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

# ### Ressourcenverbrauch
#
# Die Matrix  $\textbf{A}$  zeigt für eine jede Produkthanfrage  $j \leftarrow$ 
#  $\in \mathcal{J}$  den dafür notwendigen Ressourcenverbrauch.  $\leftarrow$ 
# Ein Vektor  $\textbf{a}_{\{j\}}$  zeigt für eine Produkthanfrage  $j \leftarrow$ 
#  $\in \mathcal{J}$  den jeweiligen Ressourcenverbrauch:

# In[6]:

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

# ### Ertrag (Revenue)
#
```

```

# Jede Produkthanfrage  $j \in \mathcal{J}$  erzielt einen Ertrag  $\leftarrow$ 
   $r_j$ :

# In [7]:

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

# ### Buchungsperioden
#
# Der Buchungshorizont wird durch den Parameter  $T$  definiert.  $\leftarrow$ 
  Die jeweilig betrachtete Periode entspricht dem Parameter  $t$   $\leftarrow$ 
  . Dabei verläuft der Buchungshorizont rückwärts in der Zeit,  $\leftarrow$ 
  so dass die erste Buchungsperiode  $t=T$  und die letzte  $\leftarrow$ 
  Buchungsperiode  $t=0$  entspricht:

# In [8]:

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

# ### Wahrscheinlichkeit der Produkthanfrage  $j$ 
#
# Eine Produkthanfrage  $j$  zum Zeitpunkt  $t$  hat eine bestimmte  $\leftarrow$ 
  Eintrittsfahrscheinlichkeit  $p_j(t)$ :

# In [9]:

def Prob(j, t):
    '''Matrix  $\text{'Prob'}$  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)))
    return matrix

```

```

#### Gegenwahrscheinlichkeit keiner Produkthanfrage
#
# Die Wahrscheinlichkeit, dass keine Anfrage eintrifft  $\leftrightarrow$ 
# entspricht  $p_0(t)$ 

# In[10]:

def Against_Prob(prob):
    ''' Gegenwahrscheinlichkeit keiner Produkthanfrage.
    Dabei entspricht der Parameter '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 matrix[i].sum() > np.float32(1.0):
            raise ValueError('Wahrscheinlichkeit ist > 1 in der  $\leftrightarrow$ 
                               Buchungsperiode %i.' % i)
        else:
            matrix2[i] = np.float32(1.0) - np.float32(matrix[i]. $\leftrightarrow$ 
                                                         sum())
    return matrix2

### Parameter für die Memofunktion
#### Mögliche Systemzustände
#
# Die möglichen Systemzustände werden in Abhängigkeit der  $\leftrightarrow$ 
# Kapazitäten der Ressourcen ermittelt. Es handelt sich um das  $\leftrightarrow$ 
# Kartesische Produkt der Ressourcenzapazitäten.

# In[11]:

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]
```

```
# ### Mögliche Systemzustände inkl. Lagerhaltung
#
```

Die möglichen Systemzustände werden bei der Auftragsannahme mit Lagerhaltung in Abhängigkeit der Kapazitäten der Ressourcen und des Lagerbestands ermittelt. Es handelt sich um das Kartesische Produkt der Ressourcenkapazitäten.

In[12]:

```
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 (b-a)
    return conditions[::-1]
```

Parameter für die Lagerhaltung

Lagerbestand

Es existiert ein Lagerbestand $y_{\{h\}}$ an bereits reparierten Produkten. Dabei sind die Produkte mit den jeweiligen Ressourcen substituierbar. Die Lagerbestände lassen sich in Vektorschreibweise als \textbf{y} schreiben:

In[16]:

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

Maximal möglicher Lagerbestand

#

Es existiert für ein jedes bereits repariertes Produkt ein maximal möglicher Lagerbestand $y_{\{h\}}^{\{max\}}$. Die Lagerbestände aller reparierten Produkte lassen sich in Vektorschreibweise als $\textbf{y}^{\{max\}}$ schreiben:


```
# In [17]:

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
```

Quellcode 2 Python-Algorithmus für die Dynamische Programmierung im Netzwerk RM

```
# 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, consumtions, times, revenues, probs, stocks, ←
max_stocks):
    '''Berechnung der Erwartungswerte des ←
    Auftragsannahmeproblems inkl. Lagerhaltung.'''

    # 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←
    glichen 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)
```

```
# Liste für die mögliche Lagerproduktion ohne ↵
# Auftragseingang wird erstellt.
```

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

```
# For-Schleife über alle möglichen Produktanfragen.
```

```
product = (j for j in products[1:] if probs[j][times↵
[0]] > 0)
```

```
for j in product:
```

```
# Kapazitätsveränderung aufgrund Auftragsannahme↵
```

```
.
```

```
change = capacities-consumptions[j]
```

```
# Prüfung, ob Kapazität keinen negativen Wert ↵
# annimmt.
```

```
if np.all((change) >= np.float32(0)):
```

```
# Initialisierung der Rechnung für ↵
# Auftragsannahme.
```

```
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 eine Produkthanfrage j ←
    # entspricht
    # der Grenzbedingung  $V(c,y,t)=-\infty$ , ←
    # falls  $c[j] < 0$ .
    order = np.float16(0)
    oc = np.float16(0)

# Lagerveränderung aufgrund Auftragsannahme.
change_stock = np.cumsum(np.array(consumptions[j]←
    ], dtype=np.uint8))
decrease_stock = np.copy(stocks)
decrease_stock = np.array(decrease_stock+←
    change_stock, dtype=np.int16)
# Prüfung, ob Lagerbestand keinen negativen Wert←
# annimmt.
if np.all((decrease_stock) >= np.float32(0)):
    # Initialisierung der Rechnung für ←
    # Lagerentnahme.
    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 eine Produkthanfrage j ←
    # entspricht
    # der Grenzbedingung  $V(c,y,t)=-\infty$ , ←
    # falls  $c[j] < 0$ .
    order_stock = np.float16(0)
    oc_stock = np.float16(0)

# Lagerveränderung aufgrund Lagerproduktion.
change_workup = 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_workup, dtype=np.int16)
# Prüfung, ob Lagerproduktion möglich ist.
if np.all((change) >= np.float32(0)) and np.all(
    ((increase_stock) <= (max_stocks)) and np.
count_nonzero(change_workup)!=0:
    # Initialisierung der Rechnung für
    Lagerproduktion.
    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 eine Produkthanfrage j
    entspricht
    # der Grenzbedingung  $V(c,y,t)=-\infty$ , falls  $c[j] < 0$ .
    order_workup = np.float16(0)
    oc_workup = np.float16(0)
    reject_list[j-1] = order_workup

# Speicherung der Ergebnisse.
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 zum Erwartungswert.
value = np.float32(reject + np.sum(accept) + (probs
    [0][times[0]]*max_reject))
# Beste Lösung wird ermittelt.
index = np.argmax(max_order)

```

```

# Der Zustand wird mit dem Erwartungswert in das Dict "solutions" gespeichert.
if np.all(max_order == 0):
    solutions[state] = [conditions[state], value, 0, 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])

    # 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 ↵
            Produktanfragen.
            for j in products:
                # Kapazitätsveränderung aufgrund der ↵
                Produktanfrage 'j' wird erfasst.
                change = np.copy(solutions[i][0][:len(↵
                    resources)-1])
                change = change-consumptions[j][1:]

            # Zielzustand zur Ermittlung der Opportunitä↵
            tskosten 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 mö←
    glich.
    if np.all((change) >= np.float32(0)):
        # Zielzustand wird ermittelt.
        reduction = np.hstack((change, ←
            solutions[i][0][len(resources←
                )-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])

# Lagerreduzierung aufgrund der ←
  Produktanfrage 'j' wird erfasst.
change_stock = np.cumsum(np.array(←
    consumtions[j][1:], dtype=np.uint8))
decrease_stock = np.copy(solutions[i←
    ][0][(len(resources)−1):−1])
decrease_stock = decrease_stock←
    change_stock
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 ←
Produktanfrage 'j' wird erfasst.
change_workup = 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_workup
if np.all((change) >= np.float32(0)) and←
np.all((increase_stock) <= (←
max_stocks[1:])) and np.count_nonzero←
(change_workup)!=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)

# Speicherung der Ergebnisse.

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

    db = pd.DataFrame(index=solutions)

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

    # 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

```

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. (1952): On the Theory of Dynamic Programming. In: Proceedings of the National Academy of Sciences of the United States of America. Bd. 38, Nr. 8, S. 716–719.
- Bellman, R. (1954): The theory of dynamic programming. Arbeitspapier. DTIC Document.
- Bertsimas, D. und Popescu, I. (2003): Revenue Management in a Dynamic Network Environment. In: Transportation Science. Bd. 37, Nr. 3, S. 257–277.
- Cheraghi, S.H.; Dadashzadeh, M. und Venkitachalam, P. (2010): Revenue management in manufacturing: a research landscape. In: Journal of Business and Economics Research (JBER). Bd. 8, Nr. 2.
- Christopher, M. (1998): Logistics and supply chain management: Strategies for reducing cost and improving service. In: International Journal of Logistics Research and Applications.
- 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: OR spectrum. 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.
- Dickersbach, J. (2004): Production Planning. In: Supply Chain Management with APO. Springer Berlin Heidelberg, S. 131–234.
- DIN 31051 (2012): Grundlagen der Instandhaltung. DIN Deutsches Institut für Normung e. V.
- 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.
- Heikkilä, J. (2002): From supply to demand chain management: efficiency and customer satisfaction. In: Journal of Operations Management. Bd. 20, Nr. 6, S. 747 – 767.
- Helber, S. und Heinze, L. (2014): Working paper: Combining hedging point and bid-price control policies in a regenerate-to-order environment.
- Helbing, K. (2010): Instandhaltung. In: Handbuch Fabrikprojektierung. Springer Berlin Heidelberg, S. 983–989.
- Hetland, M.L. (2005): Instant Hacking: The Basics. Apress. S. 1–30.
- Hetland, M.L. (2010): Python Algorithms: Mastering Basic Algorithms in the Python Language. Apress.
- 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: Information Systems Frontiers. 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: OR Spectrum. 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.
- Langtangen, H.P. (2006): Numerical Computing in Python. Bd. 3 von Texts in Computational Science and Engineering. Springer Berlin Heidelberg. S. 131–181.
- Lindblad, T. und Kinser, J.M. (2013): NumPy, SciPy and Python Image Library. In: Image Processing using Pulse-Coupled Neural Networks. Springer, S. 35–56.
- 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.
- Olhager, J. (2010): The role of the customer order decoupling point in production and supply chain management. In: Computers in Industry. Bd. 61, Nr. 9, S. 863 – 868. Trends and Challenges in Production and Supply Chain Management.
- Oliver, R.K. und Webber, M.D. (1982): Supply-chain management: logistics catches up with strategy. In: Outlook. Bd. 5, Nr. 1, S. 42–47.

- Owsnicki-Klewe, B. und Wissner, B. (1999): Algorithmen und Datenstrukturen. Wißner.
- 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.
- Powell, W.B. (2007): Approximate Dynamic Programming: Solving the curses of dimensionality. Bd. 703. John Wiley & Sons.
- Puterman, M.L. (2009): Markov decision processes: discrete stochastic dynamic programming. Bd. 414. John Wiley and Sons.
- 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.
- Schubert, K. und Fricke, D. (2005): Produktionsfaktoren. VS Verlag für Sozialwissenschaften. S. 324–327.
- Sharman, G. (1984): The rediscovery of logistics. In: Harvard Business Review. Bd. 62, Nr. 5, S. 71–79.
- 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.
- Stadtler, H. und Kilger, C. (2005): Supply Chain Management —An Overview. Springer Berlin Heidelberg. S. 9–35.
- Stadtler, H.; Kilger, C. und Meyr, H. (2008): Demand Fulfilment and ATP. Springer Berlin Heidelberg. S. 181–198.
- 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.
- Weber, H. (1999): Springer-Lehrbuch. Springer Berlin Heidelberg. S. 116–305.
- Werner, H. (2013): Grundlagen. Springer Fachmedien Wiesbaden. S. 1–90.
- 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

