

Effektivität und Effizienz durch problemspezifische Abstraktion – ein Beitrag zum maschinellen Lernen von Regeln zur Steuerung von Produktionsnetzwerken der Serienfertigung

Dissertation
zur Erlangung der Würde des
DOKTORS DER WIRTSCHAFTSWISSENSCHAFTEN
(Dr. rer. pol.)
der Universität Paderborn

vorgelegt von
Dipl.-Inform. Andre Döring
32756 Detmold

Paderborn, April 2009

Dekan: Prof. Dr. Peter F. E. Sloane
Referent: Prof. Dr.-Ing. habil. Wilhelm Dangelmaier
Korreferent: Prof. Dr. Leena Suhl

Erstellt an der Universität Paderborn
Heinz Nixdorf Institut
Wirtschaftsinformatik, insb. CIM
Prof. Dr.-Ing. habil. W. Dangelmaier
Fürstenallee 11
33102 Paderborn

Für meine Eltern und Claudia

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter in der Fachgruppe Wirtschaftsinformatik, insb. CIM am Heinz Nixdorf Institut der Universität Paderborn. Ich danke Herrn Prof. Dr.-Ing. habil. Wilhelm Dangelmaier, der mir im Rahmen meiner Tätigkeit durch die Übertragung vielfältiger herausfordernder Aufgaben in Industrie- und Forschungsprojekten ermöglichte, viele Erfahrungen zu sammeln und mich fachlich sowie persönlich weiterzuentwickeln.

Mein Projekt „Promotion“ war ein langer Prozess. Mein besonderer Dank zu dessen Gelingen gilt Herrn Prof. Dr.-Ing. habil. Wilhelm Dangelmaier, der durch seine engagierte Betreuung und Unterstützung, sowie die fachlichen und stets hilfreichen Anregungen zum Erfolg meiner Promotion beigetragen hat. Weiterhin danke ich Frau Prof. Dr. Leena Suhl für die Übernahme des Korreferats und Herrn Prof. Dr. Stefan Betz und Herrn Prof. Dr. Eckardt Steffen für die Teilnahme an meiner Promotionskommission und dem wertvollen Feedback zu meiner Arbeit.

Ich danke meinen Kollegen aus der Fachgruppe für die hilfreichen Diskussionen und Feedback sowie die freundschaftliche Unterstützung bei der Erstellung meiner Arbeit. Frau Claudia Weber danke ich für die Korrekturhilfen.

Mein besonderer Dank gehört meiner Freundin Claudia. Ihre liebevolle und ausdauernde Unterstützung hatte maßgeblichen Einfluss auf den Erfolg dieser Arbeit. Nicht weniger Dankschätzung gilt meinen Eltern, ohne deren fortwährende Unterstützung eine Ausbildung in dieser Form nicht möglich gewesen wäre.

Paderborn, April 2009
Andre Döring

Inhaltsverzeichnis

1. Einleitung	1
2. Problemstellung	9
2.1. Steuerung von Produktionsnetzwerken der Serienfertigung	9
2.1.1. Klassifikation des Untersuchungsgegenstandes	11
2.1.2. Objekte in einem Produktionsnetzwerk	13
2.1.3. Ablauf kooperativer Steuerung in Produktionsnetzwerken . .	18
2.2. Anforderungen einer automatisierten Steuerung der Änderungsplanung durch Regeln	21
2.2.1. Problem der Entscheidungsfindung	22
2.2.2. Formalisierung der Regeln - lokale und globale Entscheidungen	24
2.2.2.1. Regeln für deterministische Änderungsplanungsprozesse	25
2.2.2.2. Regeln für nicht-deterministische Änderungsplanungsprozesse	26
2.2.3. Verfahren zur automatisierten Regelsystemerstellung	27
2.2.3.1. Ziel der automatisierten Regelerstellung	27
2.2.3.2. Verwendbare Verfahren zur automatisierten Regelerstellung	28
2.2.4. Maschinelles Lernen der Regeln	29
2.2.4.1. Maschinelle Lernverfahren zur Produktionssteuerung	30
2.2.4.2. Maschinelles Lernen in komplexen Umgebungen .	32
2.2.4.3. Einsatz von Q-Learning	33
2.2.5. Übertragung der Q-Learning-Konzepte auf die Problemstellung	36
2.2.5.1. Zustand	37
2.2.5.2. Aktionen	37
2.2.5.3. Reward und Rewardfunktion	37
2.2.5.4. Zustand, Reward und Q-Update	38
2.2.5.5. Ausgangsdaten für den automatisierten Lernprozess	40
2.3. Zusammenfassung der Problembereiche	41
2.3.1. Effektive Abstraktion des Zustandsraumes	41
2.3.2. Effektive Lernfunktion	42
2.3.3. Automatisierte Regelgenerierung durch effizientes Training .	44
2.3.4. Systemarchitektur und Zusammenfassung	45

3. Stand der Forschung	47
3.1. Zustandsreduktionsverfahren für Produktionsnetzwerke der Serienfertigung	47
3.1.1. Approximation der Value-Funktion	48
3.1.1.1. Feature-Vektoren	49
3.1.1.2. Regressionsbäume	50
3.1.1.3. Relationale Zustandsbeschreibungen	50
3.1.1.4. Bewertung	50
3.1.2. Approximation durch Zustandsaggregation	51
3.1.2.1. Entscheidungsbäume	52
3.1.2.2. Clusterverfahren	52
3.1.2.3. Bewertung	53
3.1.3. Anwendung von <i>k-means</i> -Clustering	54
3.1.3.1. Bewertung	57
3.1.4. Konvergenz von Q-Learning auf einem abstrahierten Zustandsraum	59
3.2. Q-Learning zum Lernen von Steuerungsregeln in Produktionsnetzwerken	60
3.3. Durchführung von Training und Generierung von Ausgangsdaten . . .	63
3.3.1. Training mit einer Multiagentensystemarchitektur	64
3.3.2. Ausgangsdaten für das Training	67
3.4. Zusammenfassung	68
4. Zu leistende Arbeit	69
5. Konzeption	73
5.1. Reduktion des Zustandsraumes durch Clustering	74
5.1.1. Vorüberlegungen	74
5.1.1.1. Ausreißer	74
5.1.1.2. Prinzipablauf und Nutzen des Clusterings	75
5.1.2. Aufbau der Abstraktionsfunktion	76
5.1.2.1. Auswahl relevanter Merkmale für die Zustandsbeschreibung	76
5.1.2.2. Unterscheidungskriterien zur Zustandsabstraktion	79
5.1.2.3. Auswahl der charakteristischen Pläne	81
5.1.3. Erlernen charakteristischer Pläne mit <i>k-means</i> -Clustering	84
5.1.3.1. Anforderungen an die Distanzfunktion	84
5.1.3.2. Strukturelle Distanz	85
5.1.3.3. Quantitative Distanz	88
5.1.3.4. Kombinierte Distanzfunktion und Beispiel	88
5.1.3.5. Einfluss der Gewichtungsfaktoren	91
5.1.3.6. Aktualisierung der Clustermittelpunkte	91
5.1.3.7. Auswahl der initialen Clustermittelpunkte	94
5.1.3.8. Trainingsdaten für das Clustering	94

5.1.3.9.	Terminierung des Clusterverfahrens	98
5.1.4.	Zusammenfassung	99
5.2.	Konzeption der Lernfunktion für das Q-Learning	100
5.2.1.	Planungsverfahren und Varianten im Lernsystem	100
5.2.2.	Rewardbewertung auf Clusterebene	101
5.2.3.	Strafkostenarten in der Rewardfunktion	103
5.2.4.	Grundprinzip bei der Rewardberechnung	105
5.2.5.	Vorlaufzeiten von Planungsprozessen in der Rewardfunktion	106
5.2.6.	Bewertung von Restriktionsverletzungen	107
5.2.7.	Bereitstellungsstrafkosten am Fertigungsobjektknoten	108
5.2.7.1.	Parameter der Kostenfunktion	108
5.2.7.2.	Vergleichbarkeit der Strafkosten	109
5.2.7.3.	Periodenweise Strafkostenfunktion für lokale Plan- änderungen	111
5.2.7.4.	Periodenweise Strafkosten am FOK	113
5.2.7.5.	Kumulierte Strafkosten am FOK	114
5.2.8.	Betriebsmittelstrafkosten am Kapazitätsobjektknoten	115
5.2.8.1.	Parameter der Strafkostenfunktion	116
5.2.8.2.	Periodenweise Strafkosten am KOK	117
5.2.8.3.	Kumulierte Strafkosten am KOK	118
5.2.9.	Beschaffungsstrafkosten am Fertigungsobjektknoten	119
5.2.9.1.	Strafkosten für unterschiedliche Beschaffungssteue- rungen	120
5.2.9.2.	Leistungsvereinbarungen im Beschaffungsprozess	122
5.2.9.3.	Strafkosten bei Bestellpunktverfahren	123
5.2.9.4.	Strafkosten bei Bestellzyklusverfahren	125
5.2.9.5.	Übertragung der Konzepte auf die angebotsseitige Koordination	126
5.2.9.6.	Koordination zwischen FOK/KOK und FOK/FOK	126
5.2.9.7.	Bewertung globaler Beschaffungsprozesse durch lo- kal berechnete Strafkosten	127
5.2.9.8.	Globale Koordination mit mehreren Partnern	129
5.2.10.	Bewertung eines Endzustandes	130
5.2.11.	Anmerkung zum Q-Update auf Clusterebene	131
5.2.12.	Gesamtrewardfunktion	132
5.3.	Konzeption des Trainings, der Lernepisoden und der Regelgenerierung	133
5.3.1.	Lernepisoden und deren Ausgangsdaten	134
5.3.1.1.	Lernschritte am Objektknoten	135
5.3.1.2.	Sequenz von Lernepisoden	135
5.3.1.3.	Auswahl der Änderungsplanungsverfahren in einer Lernepisode	138
5.3.2.	Funktionale Einbindung der Lernepisoden in das Training	139
5.3.2.1.	Lernrate und Abbruchbedingungen	139

5.3.3.	Generierung und Verwendung von Regeln	142
5.3.3.1.	Regelgenerierung – Von Q-Werten zum Regelsystem	142
5.3.3.2.	Partielle Aktualisierung von Regeln	143
5.3.3.3.	Steuerung – Vom Zustand zur Regelanwendung . .	144
5.3.4.	Konvergenzbetrachtung des Lernverfahrens	144
5.3.5.	Zusammenfassung Training und Regelanwendung	146
5.4.	Zusammenfassung und Bewertung der Konzeption	147
6.	Validierung	153
6.1.	Validierung des Abstraktionsverfahrens	153
6.1.1.	Szenario zur Validierung der problemspezifischen Abstraktion	154
6.1.2.	Validierung der Parametereinstellungen	157
6.1.2.1.	Anzahl der Cluster	157
6.1.2.2.	Gewichte der Distanzfunktion	159
6.1.2.3.	Maximale Anzahl Iterationen und Konvergenztoleranz	160
6.1.3.	Effektivität der Clusteranwendung	161
6.1.4.	Zusammenfassung	165
6.2.	Lernverfahren und Training	165
6.2.1.	Szenario zur Validierung des Lernverfahrens	165
6.2.2.	Validierung der Lernfunktion	167
6.2.2.1.	Szenario 1 - Effektivität der Lernfunktion	167
6.2.2.2.	Szenario 2 - Effizienz der Lernfunktion	173
6.2.3.	Validierung des Trainingsprozesses	178
6.2.3.1.	Konvergenz des Verfahrens	178
6.2.3.2.	Dauer des Trainings bei variierender Zustandsraum- größe	180
6.2.3.3.	Effizienz des Trainingsprozesses	182
6.2.4.	Lernen im Netzwerk	183
6.3.	Abschließende Diskussion	184
7.	Zusammenfassung und Ausblick	187
7.1.	Zusammenfassung	187
7.1.1.	Reduktion des Zustandsraumes	188
7.1.2.	Konzeption einer Lernfunktion	188
7.1.3.	Ausgangsdaten und Trainingskonzept	190
7.1.4.	Umsetzung	190
7.2.	Grenzen der Arbeit	191
7.3.	Ausblick	191
A.	Liste Planungsverfahren und Varianten	207
B.	Generieren von Trainingsdaten	211

C. Implementation	213
C.1. Gesamtsystem	213
C.1.1. JADE als Plattform	213
C.1.1.1. Klassenhierarchie und Funktion	214
C.1.1.2. Technische Umsetzung der Koordination	219
C.1.1.3. Konfiguration	220
C.2. Clustering	222
C.2.1. Ausgabemöglichkeiten	223
C.2.2. Integration in das Lernverfahren	224
C.3. Lernfunktion und Training	226
C.3.1. Umsetzung und Überwachung des Trainings	226

Abbildungsverzeichnis

1.1.	Effektivität und Effizienz durch problemspezifische Abstraktion	6
2.1.	Grafische Repräsentation eines elementaren MFERT-Modells	13
2.2.	Beispielplan für einen FOK	15
2.3.	Arten der Kunden-Lieferanten-Beziehungen	16
2.4.	Beispiel für die Klassifikation einer elementaren Planungsstrategie . .	20
2.5.	Input-Output des Lernverfahrens und der Regelanwendung	31
2.6.	Beispiel für das Lernproblem	32
2.7.	Reinforcement-Learning-Systemarchitektur	34
2.8.	Lernpfad des Q-Learnings	39
2.9.	Architektur des Lernsystems	45
3.1.	Framework zur Approximation der Value-Funktion	48
3.2.	Beispiel für die Anwendung des k -means-Algorithmus	56
3.3.	Zusammenhang von Zustand, Cluster und Q-Werten	57
4.1.	Abfolge der zu leistenden Arbeit	71
5.1.	Beispielhafte Darstellung einer Plannormierung	78
5.2.	Entscheidungsbaum zur Herleitung charakteristischer Pläne im Clu- stering	80
5.3.	Charakteristische Pläne und Planungsverfahren	82
5.4.	Beispiel für ähnliche Pläne und deren charakteristischen Plan	83
5.5.	Problematik bestehender Ähnlichkeitsmetriken für binäre Vektoren . .	86
5.6.	Beispiel zur Berechnung der strukturellen Distanz	87
5.7.	Beispiel zur Berechnung der Gesamtdistanzfunktion	90
5.8.	Aktualisierung der Clustermittelpunkte für unterschiedliche w_{RV}^C	93
5.9.	Extraktion und Art der realen Ausgangsdaten für eine Lernepisode . .	95
5.10.	Aktivität: Extraktion Realdaten	96
5.11.	Merkmale der Ausgangsdaten	98
5.12.	Berechnung der Q-Werte im abstrahierten Zustandsraum	102
5.13.	Aktionspfad Q-Learning auf Clusterebene	103
5.14.	Beispiel für die Abzinsung der Strafkosten durch den Diskontfaktor DF in Bezug zum Plan eines FOK	107
5.15.	Wirkungsweise der Abzinsung	108

5.16. Skizzierung der Wirkungsweise der Strafkostenfunktion für Bestands- werte ohne Betrachtung verschobener Restriktionsgrenzen	111
5.17. Strafkostenberechnung am Beispiel eines FOK-Plans	115
5.18. Prinzip der globalen Koordination	119
5.19. Skizzierung der Strafkostenparameter am FOK	121
5.20. Zusammenhang von Bedarfstermin, LTS und $p(b)$	122
5.21. Sequenz: Vereinfachtes Beispiel der Varianten der paarweisen Koordi- nation	128
5.22. Problem global verteilter Anfragen in der Strafkostenberechnung . . .	131
5.23. Aktivität: Lernepisode (bedarfsorientiert)	136
5.24. Aktivität: Lernepisode (angebotsorientiert)	137
5.25. Aktivität: Erweitertes Koordinationsprotokoll für Angebotsänderun- gen eines Knotens bei komplementären Zugängen im Nachfolgeknoten	140
5.26. Aktivität: Schrittweiser Ablauf einer Regelanwendung	145
5.27. Einbindung des Regelsystems in den Anwendungskontext	146
5.28. Aktivität: Erstellung des Regelsystems	149
6.1. Beispielproduktionsnetzwerk für die Validierung des Clusterings . . .	155
6.2. Planverläufe für unterschiedliche Zu- und Abgangsmuster	156
6.3. Summe quadrierter Distanzen für verschiedene k	158
6.4. Entwicklung des quadrierten Fehlers und Anteil neuer Clusterzuwei- sungen über die Iterationen des Clusterings	160
6.5. Verteilung der Strafkosten für die Planverläufe verschiedener Cluster .	163
6.7. Beispielproduktionsnetzwerk zur Validierung des Lernverfahrens . . .	166
6.8. Ausgangsplan Szenario 1	168
6.9. Szenario 1, Fall 1 - Lokale Aktion wird bevorzugt	170
6.10. Szenario 1, Fall 2 - Globale Koordination wird bevorzugt	172
6.11. Originalplan und Centroid des Clusters	174
6.12. Szenario 2, Fall 1 - Lokale Aktion wird bevorzugt	175
6.13. Szenario 2, Fall 2 - Globale Koordination wird bevorzugt	177
6.14. Benötigte Lernschritte im Training	180
6.15. Laufzeitabschätzung des Trainings (1)	181
6.16. Laufzeitabschätzung des Trainings (2)	182
C.1. Screenshot: JADE Agentenmonitor mit initialisiertem Produktionsnetz- werk	215
C.2. Klassendiagramm: Vereinfachte Darstellung des Lernsystems	216
C.3. Screenshot: Konfiguration des RLPP	221
C.4. Sequenz: Ablauf eines Lernschrittes	225
C.5. Technischer Ablauf einer Koordination im Lernsystem	227
C.6. Screenshot: Trainingsprotokoll mit Koordinationen in Eclipse	231
C.7. Screenshot: UI-Konzept eines Überwachungsmonitors für das Lern- verfahren	232

Tabellenverzeichnis

2.1.	Organisatorische Einordnung des Untersuchungsgegenstandes in das Klassifikationsschema nach Eisenführ	12
2.2.	Zulässige Verknüpfungen im Modell der Fertigung	14
2.3.	Möglichkeiten der Beschaffungssteuerung	18
2.4.	Aufbau der Steuerungsregeln nach Heidenreich	24
2.5.	Gegenüberstellung durchzuführender Vorbereitungsmaßnahmen zur Regelerstellung mit und ohne automatisierte Methoden	28
5.1.	Distanzfunktion für verschiedene w_S und w_Q	91
5.2.	Attribute der Ausgangsdaten	97
5.3.	Zusammenfassung Forschungsfrage A	150
5.4.	Zusammenfassung Forschungsfrage B	150
5.5.	Zusammenfassung Forschungsfrage C	151
6.1.	Definition der Testszenarien	155
6.2.	Konfiguration EM im Lernszenario	166
6.3.	Planungsergebnisse Szenario 1.1	168
6.4.	Parameter im Lernszenario „Effektivität“	169
6.5.	Zuordnungsdauer von Plan zu Cluster	179
A.1.	Planverfahren bei Änderung der Restriktionen am FOK	207
A.2.	Elementare Planungsverfahren am KOK	208
A.3.	Bedarfsseitige elementare Planungsverfahren am FOK	209
A.4.	Angebotsseitige elementare Planungsverfahren am FOK	210
B.1.	Parameter für das Generieren von Trainingsdaten	212
C.1.	Nachrichtenfelder und Inhalte der RLPP-Koordinationsnachrichten	219

Abkürzungsverzeichnis

α	Lernfaktor im Q-Learning
γ	Diskontfaktor im Q-Learning
\hat{p}	Binäre Repräsentation eines Planes
$\hat{p}(k)$	Binäre Repräsentation einer Planungsperiode zum Zeitpunkt k
\mathcal{A}	Menge zulässiger Aktionen im Reinforcement Learning
\mathcal{S}	Zustandsraum im Reinforcement Learning
ω_{loc}^{aon}	Gewichtungsfaktor für Gesamtstrafkosten eines FOK
ω_{pc}^{aon}	Gewichtungsfaktor für Gesamtstrafkosten eines FOK
ω_{proc}^{aon}	Gewichtungsfaktor für Strafkosten bestellpunktbasierter Beschaffungsprozesse eines FOK
ω_{rv}^{aon}	Gewichtungsfaktor für Strafkosten eines FOK verursacht durch Restriktionsverletzungen
ω_{cost}^{con}	Gewichtungsfaktor der allgemeinen Strafkosten eines KOK
$\omega_{maxtemp}^{con}$	Gewichtungsfaktor für temporäre Überschreitung des Leistungsgrades eines KOK
ω_{max}^{con}	Gewichtungsfaktor für Überschreitung des Leistungsgrades eines KOK
$\omega_{mintemp}^{con}$	Gewichtungsfaktor für temporäre Unterschreitung der Minimalauslastung eines KOK
ω_{min}^{con}	Gewichtungsfaktor für Unterschreitung der Minimalauslastung eines KOK
ω_{rv}^{con}	Gewichtungsfaktor der Strafkosten eines KOK verursacht durch Restriktionsverletzungen
ω_{cost}^{crep}	Gewichtungsfaktor für Strafkosten durch Zusatzkosten eines bestellzyklusbasierten Beschaffungsprozesses
ω_{max}^{crep}	Gewichtungsfaktor für Strafkosten durch Überschreitung festgelegter Bestellmengen in bestellzyklusbasierten Beschaffungsprozessen
ω_{min}^{crep}	Gewichtungsfaktor für Strafkosten durch Unterschreitung festgelegter Bestellmengen in bestellzyklusbasierten Beschaffungsprozessen

ω_{aon}^{glob}	Gewichtungsfaktor für globale Strafkosten eines FOK
ω_{aon}^{loc}	Gewichtungsfaktor für lokale Strafkosten eines FOK
ω_{cost}^{loc}	Gewichtungsfaktor für allgemeine Strafkosten eines FOK
$\omega_{maxtemp}^{loc}$	Gewichtungsfaktor für Strafkosten bei temporärer Änderung des Maximalbestandes eines FOK
ω_{max}^{loc}	Gewichtungsfaktor für Strafkosten bei Überschreitung des Maximalbestandes eines FOK
$\omega_{mintemp}^{loc}$	Gewichtungsfaktor für Strafkosten bei temporärer Änderung des Sicherheitsbestandes eines FOK
ω_{min}^{loc}	Gewichtungsfaktor für Strafkosten bei Unterschreitung des Sicherheitsbestandes eines FOK
ω_{rv}^O	Gewichtungsfaktor für Restriktionsverletzungen für einen Objektknoten
ω_{cost}^{proc}	Gewichtungsfaktor für Strafkosten durch Zusatzkosten in bestellpunktbasierten Beschaffungsprozessen
ω_{max}^{proc}	Gewichtungsfaktor für Strafkosten durch Überschreitung festgelegter Bestellmengen bestellpunktbasierter Beschaffungsprozesse
ω_{min}^{proc}	Gewichtungsfaktor für Strafkosten durch Unterschreitung festgelegter Bestellmengen bestellpunktbasierter Beschaffungsprozesse
π	Policy im Reinforcement Learning
A^*	Menge aller Abbruchbedingungen einer Lernepisode
a_i	Aktion aus der Menge zugelassener Aktionen eines Zustandes oder Clusters PA
$abort_{tnle}$	Abbruch nach Anzahl n durchgeführter Lernschritte
$abort_{tnrt}$	Abbruch nach einer definierten Laufzeit des Trainings
$abort_{ac}$	Abbruch nach x Ablehnungen während der Koordination in den Lernschritten
$abort_{ls}$	Abbruch nach Durchführung von n Lernschritten
$abort_{nc}$	Abbruch gesteuert durch die Anzahl der verarbeiteten Änderungen in den Lernepisoden
$abort_{rnd}$	Zufälliger Abbruch nach dem Ziehen einer Zufallszahl aus einem vorgegebenen Intervall
$abort_{ur}$	Abbruch nach Unterschreitung eines prozentual anteiligen minimalen Reward vom letzten berechneten Reward bei Anwendung des derzeit Q-besten Planungsverfahrens für ein Cluster-/Aktionspaar
AON	Assembly Object Node (FOK)
$best(c_m, a_n)$	Bester gewählter Cluster zu einem ungültigen Zustand zu dessen Auflösung durch Anwendung der Regeln
C_i	Cluster i im Lernsystem

c_i	Initiale Clustermittelpunkte
C_s	Charakteristischer Plan des Clusters s
CON	Capacity Object Node (KOK)
$CREP$	Bestellpunktverfahren (Continuous Replenishment)
$d(P_i, P_j)$	Distanz zwischen zwei Plänen P_i und P_j
D_Q	Quantitative Distanz
D_S	Strukturelle Distanz
DF	Diskontfaktor zur Abzinsung der Gewichtung von Strafkosten zu Planungsperioden
$discount$	Abzinsungsfaktor für zukünftige Restriktionsverletzungen in der Rewardfunktion
$dist(k)$	Strukturelle Distanz zwischen zwei Restriktionsverletzungen zweier Pläne
$DPC(P_s^O, P_{s+1}^O)$..	Differenzkosten zwischen zwei Plänen eines Objektknotens O zum Zeitpunkt s und $s + 1$
LTS	Vorlaufzeitverschiebung (Lead Time Shift)
$max(q_i)$	Höchster gelernter Q-Wert eines Clusters
NUM_MF_{all} ...	Anzahl aller Beschaffungen einer Koordination
NUM_MF_{out} ...	Anzahl verletzter Leistungsvereinbarungen eines Beschaffungsprozesses
NUM_MF_{raise} ..	Anzahl verletzter Leistungsvereinbarungen zu vereinbarten Beschaffungszeitpunkten
$p(b)$	Periode unter Berücksichtigung einer Vorlaufzeitverschiebung (LTS)
$p(b)^{mf}$	Materialfluss in einer Periode zwischen zwei FST
$p(b)_{absmax}^{mf}$	Absolute Obergrenze für die Erhöhung des Materialflusses zwischen zwei FST
$p(b)_{max}^{mf}$	Maximaler Materialfluss zwischen zwei FST je Periode
$p(b)_{min}^{mf}$	Minimaler Materialfluss zwischen zwei FST je Periode
$p(k)$	Planungsperiode zum Zeitpunkt k
$p(k)^{pl}$	Zugewiesenes Kapazitätsangebot eines KOK je Periode
$p(k)_{absmax}^{pl}$	Physikalisches absolutes Maximum an möglichem Leistungsgrad eines KOK
$p(k)_{maxtemp}^{pl}$	Temporäre Obergrenze für den Leistungsgrad eines KOK
$p(k)_{max}^{pl}$	Leistungsgrad eines KOK
$p(k)_{mintemp}^{pl}$	Temporäre Untergrenze für die minimale Auslastung eines KOK
$p(k)_{min}^{pl}$	Minimale Auslastung eines KOK
$p(k)^{sup}$	Bestand eines FOK in Periode $p(k)$

$p(k)_{absmax}^{sup}$	Physikalische Obergrenze für Planwerte eines FOK
$p(k)_{maxtemp}^{sup}$	Temporäre Obergrenze für Planwerte eines FOK
$p(k)_{max}^{sup}$	Maximale Lagermenge eines FOK
$p(k)_{minlog}^{sup}$	Logische Untergrenze für Planwerte eines FOK
$p(k)_{mintemp}^{sup}$	Temporäre Untergrenze für Planwerte eines FOK
$p(k)_{min}^{sup}$	Minimale Lagermenge eines FOK
$p^*(k)$	Normierter Plan einer Periode k
P_s	Plan zum Zeitpunkt s
PA	Menge aller zugelassener Änderungsplanungsverfahren für den Untersuchungsgegenstand
$PC(p(k))$	Strafkosten einer Periode $p(k)$
$PC(P^O)$	Strafkosten eines Planes eines Objektknotens O
PC_AON	Gesamtstrafkosten eines FOK
PC_AON^{crep} ...	Strafkosten eines bestellzyklusbasierten Beschaffungsprozesses
$PC_AON_{cost}^{crep}$...	Strafkosten für Zusatzkosten in bestellzyklusbasierten Beschaffungsprozessen
$PC_AON_{max}^{crep}$...	Strafkosten für die Überschreitung festgelegter Bestellmengen in bestellzyklusbasierten Beschaffungsprozessen
$PC_AON_{min}^{crep}$...	Strafkosten für Unterschreitung festgelegter Bestellmengen in bestellzyklusbasierten Beschaffungsprozessen
$PC_AON_{OFF}^{crep}$...	Strafkosten für durchgeführte Beschaffungsprozesse außerhalb eines Bestellzyklus
$PC_AON_{ON}^{crep}$...	Strafkosten für durchgeführte Beschaffungsprozesse innerhalb eines Bestellzyklus die dennoch Leistungsvereinbarungen verletzen
PC_AON^{glob} ...	Strafkosten für beschaffungsseitige Koordinationsprozesse eines FOK
PC_AON^{loc}	Strafkosten eines FOK für lokale Planungsverfahren
$PC_AON_{cost}^{loc}$	Allgemeine Strafkosten eines FOK
$PC_AON_{max}^{loc}$	Strafkosten bei Überschreitung von Restriktionsgrenzen eines FOK
$PC_AON_{min}^{loc}$	Strafkosten bei Unterschreitung von Restriktionsgrenzen eines FOK
PC_AON^{proc} ...	Strafkosten für bestellpunktbasierte Beschaffungsprozesse
$PC_AON_{cost}^{proc}$...	Strafkosten für Zusatzkosten eines bestellpunktbasierten Beschaffungsprozesses
$PC_AON_{max}^{proc}$...	Strafkosten bei Überschreitung festgelegter Bestellmengen in bestellpunktbasierten Beschaffungsprozessen
$PC_AON_{min}^{proc}$...	Strafkosten bei Unterschreitung festgelegter Bestellmengen in bestellpunktbasierten Beschaffungsprozessen
PC_CON	Gesamtstrafkosten eines KOK
PC_CON^{loc}	Kumulierte gewichtete Strafkosten eines Planes P_s eines KOK

$PC_CON_{cost}^{loc}$	Kosten der Leistungserstellung eines KOK
$PC_CON_{max}^{loc}$	Strafkosten für Überschreitung der logischen Leistungsgradgrenze eines KOK
$PC_CON_{min}^{loc}$	Strafkosten für die Unterschreitung einer minimalen Auslastung eines KOK
PC_OOP	Strafkosten für Beschaffungsprozesse außerhalb von Leistungsvereinbarungen
PC_RV	Strafkosten für Restriktionsverletzungen des Planes eines Objektknotens
PH	Anzahl Perioden im Planungshorizont PHZ
PHZ	Planungshorizont
$PROC$	Bestellrhythmusverfahren (Procurement)
Q	Q-Wert im Q-Learning
$Q(C_s, a_i)$	Q-Update auf Clusterebene
$RV(P^O)$	Strafkosten der Restriktionsverletzungen eines Planes
RV_P	Restriktionsverletzungen eines Planes P beim Clustering
S_i	Zustandscluster i aus der Abbildung von Zuständen in einen partitionierten Zustandsraum
s_i	Zustand aus der Menge aller Zustände eines Clusters C_i
V^π	Value-Funktion im Reinforcement Learning
w_{RV}^c	Gewichtungsfaktor für Restriktionsverletzungen bei der Aktualisierung der Clustermittelpunkte
w_{RV}^d	Gewichtungsfaktor für Restriktionsverletzungen bei der quantitativen Distanz
w_q	Gewichtungsparameter der quantitativen Distanz in der Gesamtdistanz
w_s	Gewichtungsparameter der strukturellen Distanz in der Gesamtdistanz
AC/DC	Advanced Chassis Development for 5DayCars (Europäisches Verbundprojekt)
AR	Age Replacement
BA	Bruttoangebot
BB	Bruttobedarf
COR	Coefficient of Operational Readiness
DSS	Descison-Support-System
EBNF	Extended Backus Naur Form
ERP	Enterprise Resource Planning
FOK	Fertigungsobjektknoten

FST	Fertigungsstufe
GE	Geldeinheit
KNN	Künstliches Neuronales Netz
KOK	Kapazitätsobjektknoten
MAS	Multiagentensystem
MDP	Marcov-Decision-Process
MFERT	Modell des Fertigungsgeschehens
MRP	Material Requirements Planning
NA	Nettoangebot
NB	Nettobedarf
OOP	Engl. <i>out of plan</i> . Index bei der Berechnung globaler Strafkosten
OR	Operations Research
PC	Penalty Costs
PK	Prozessknoten
PPS	Produktionsplanung und -steuerung
R	Reward
REQ	Anfrage in der kooperativen Änderungsplanung
RESP	Antwort auf eine Anfrage in der kooperativen Änderungsplanung
RLV	Reinforcement-Learning-Verfahren
RV	Restriktionsverletzungen eines Planes
SCM	Supply Chain Mangement
SEZ	Spätester Endzeitpunkt
ST	Stück

1. Einleitung

Die Neugier steht immer an erster Stelle eines Problems, das gelöst werden will.

(Galileo Galilei)

Moderne Wertschöpfungsprozesse in der Serienfertigung werden zu globalen *Produktionsnetzwerken* zusammengeschlossen, um durch Outsourcing Wertschöpfungstiefen einzelner Standorte zu reduzieren. Es wird das Ziel verfolgt, die Effizienz der globalen Produktion durch Spezialisierung der Produktionsprozesse an verteilten Standorten hinsichtlich Produktivität und Flexibilität zu erhöhen. Ziel ist es, Kosten im gesamten Produktionsnetzwerk zu senken.¹

Der Materialfluss in Produktionsnetzwerken erfolgt innerhalb einer sogenannten *Supply Chain* vom Rohstofflieferanten bis zum Endkunden. Die Herausforderung im Management von Supply Chains liegt in der „integrierten prozessorientierten Planung und Steuerung der Waren-, Informations- und Geldflüsse entlang der gesamten Wertschöpfungskette vom Kunden bis zum Rohstofflieferanten“². Diese Planungs- und Steuerungsprozesse erfolgen innerhalb des Produktionsnetzwerkes sowohl über kooperativ, als auch kompetitiv kooperierende Unternehmen.³ Insbesondere bei der kompetitiven Zusammenarbeit stehen Planungsziele der einzelnen Unternehmen im Konflikt zueinander. Dieser Konflikt bedingt sich durch die wirtschaftliche und so auch planerische Autonomie der einzelnen Partner im Produktionsnetzwerk.⁴

Um heterogene Supply Chains planen und steuern zu können, werden im *Supply Chain Management* (SCM) geeignete Methoden entwickelt, mit denen den entstehenden Herausforderungen im SCM begegnet werden kann. Im Bereich der europäischen Automobilindustrie, aber auch in anderen Branchen wie der Textilindustrie⁵, wird für den Forschungskomplex Supply Chain Management diskutiert, welche neuen und intelligenten, standortübergreifenden Planungs- und Steuerungskonzepte für Produktionsnetzwerke zukünftig benötigt werden.⁶

¹[FM04]

²[KH02], S. 10

³[BHHB06]

⁴[DDKT07]

⁵Z. B. bei der Firma Gerry Weber in Halle/Wstf. ([DN07])

⁶Z. B. EU-Projekt AC/DC, ein FP6 Verbundprojekt, Fördernummer 031520, Laufzeit 2006-2010 ([DDKT07])

Ein Fokus in der aktuellen akademischen, sowie industriell orientierten Forschung in der Wirtschaftsinformatik für das Supply Chain Management liegt auf der intelligenten Steuerung von Produktionsnetzwerken.⁷ Durch die Kombination von Methoden aus der künstlichen Intelligenz in der Informatik und Methoden der Wirtschaftswissenschaft sollen z. B. Steuerungssysteme für Produktionsnetzwerke entwickelt werden, mit denen ungültige Zustände des Produktionsnetzwerkes weitgehend automatisiert und möglichst schnell in gültige Zustände überführt werden können.⁸ In dieser Arbeit wird ein solches intelligentes Verfahren zur Steuerung der Änderungsplanung in Produktionsnetzwerken über einen interdisziplinären Ansatz aus Methoden der Informatik und Wirtschaftswissenschaften konzeptionell erarbeitet.

Steuerung zur Beseitigung ungültiger Zustände

Ein Zustand ist die Folge eines *Ereignisses*⁹ innerhalb eines Produktionsnetzwerkes. Ereignisse haben ihren Ursprung in vorausgeplanten Aktivitäten im Produktionsnetzwerk. Diese Aktivitäten umfassen die planmäßige Übermittlung von Zu- und Abgängen an Produktionsfaktoren, wie z. B. Material und Betriebsmittel, zwischen den Partnern des Produktionsnetzwerkes. Ereignisse können aber auch durch unplanmäßige Änderungsanfragen für Produktionsfaktoren entstehen, z. B. nach dem Ausfall einer Maschinenkapazität, die eine Angebotsreduzierung für Produkte eines Lieferanten bewirkt.

Die Ermittlung der Konsequenzen eines Ereignisses auf die Pläne der Partner im Produktionsnetzwerk wird durch die Planbestandsrechnung¹⁰ ermöglicht. In der Planbestandsrechnung werden alle Zu- und Abgänge in bestehende Material- und Betriebsmittelpläne eingerechnet. Aus der Planbestandsrechnung resultieren Zustände eines Produktionsnetzwerkes zu einem bestimmten Zeitpunkt. Diese Zustände werden durch Pläne detailliert. Ein Zustand kann, je nach Betrachtungstiefe, z. B. für das gesamte Produktionsnetzwerk, einzelne Unternehmen oder einzelne Fertigungslinien modelliert werden. Für die Steuerung ist die detaillierte Betrachtung von Zuständen als Pläne der in der Supply Chain involvierten Fertigungs- und Kapazitätsobjekte¹¹ unter definierten Restriktionen notwendig.

Unter Berücksichtigung eines gegebenen Planes resultiert ein neuer *Zustand* nach einem Ereignis aus

- der Art der Planänderung als Änderung des Zu- oder Abgangs,

⁷Siehe ebd., [HM07], [DDLT07]

⁸Ebd.

⁹Ein Ereignis ist eine „atomare Begebenheit, die eine Zustandsänderung bewirkt und keine Zeit verbraucht.“ ([VDI93]) Oder analog: “Ein Ereignis ist die Änderung eines Zustandes, bezogen auf einen Zeitpunkt.“ ([Sch96], S. 108)

¹⁰Z. B. in [Hei06], S. 192

¹¹Details zu Fertigungs- und Kapazitätsobjekten siehe [Sch96]

-
- dem Zeitpunkt der Planänderung als Planungsperiode und
 - der Höhe der Planänderung als Menge.

Da die Zusammenarbeit einzelner Partner im Produktionsnetzwerk z. B. durch eine differierende Beschaffungssteuerung ausgestaltet sein kann, ist zusätzlich zur Charakterisierung eines Zustandes

- der anfragende Partner

zu betrachten.

Zustände können per Definition *gültig* oder *ungültig* sein. Ein Zustand ist ungültig, wenn eine gegebene planerische Restriktion, wie die Unterschreitung des Sicherheitsbestandes, durch den Plan des Zustandes verletzt wird. Die Steuerung setzt auf den ungültigen Zuständen auf und versucht diese durch die Anwendung von Planungsverfahren im Rahmen einer Änderungsplanung zu beseitigen.

Steuerung in Produktionsnetzwerken mit Hilfe der Änderungsplanung

Ziel einer effizienten Steuerung von Produktionsnetzwerken ist es, durch geringe Änderungen eines bestehenden ungültigen Planes in einen gültigen Plan einzuschwingen.¹² Die erforderlichen Planungsprozesse zur Umsetzung der Steuerung werden durch die *Änderungsplanung* realisiert, z. B. durch die Umplanung von Losen, die Änderung von Bedarfen beim Lieferanten, sowie die Änderungen von Restriktionsgrenzen eines Plans. Bei der Änderungsplanung wird zur Beseitigung eines ungültigen Zustandes innerhalb des Produktionsnetzwerkes sowohl der quantitative Materialfluss, als auch die Auslastung verfügbarer Kapazitäten durch die Anwendung von *Änderungsplanungsverfahren* angepasst.

Ungültige Zustände können durch die Änderungsplanung sowohl durch *lokale*, als auch durch *globale* Planungsmaßnahmen aufgelöst werden. Einer Bedarfserhöhung kann z. B. durch Verwendung des Sicherheitsbestandes lokal begegnet werden. Ist die lokale Behebung eines ungültigen Zustandes nicht möglich oder gewünscht, können global *Bedarfe* oder *Angebote* von *Material* oder *Betriebsmitteln* innerhalb des Produktionsnetzwerkes reduziert oder erhöht, also geändert werden. Die Regelung der erforderlichen Abstimmungsvorgänge erfolgt durch koordinative Prozesse mit festgelegtem Protokoll zwischen den einzelnen Partnern des Produktionsnetzwerkes. Heidenreich¹³ bezeichnet den ersten Fall als Änderungsplanung mit einer *lokalen Planungsstrategie*¹⁴ und den zweiten Fall als Änderungsplanung mit einer *globalen Planungsstrategie*¹⁵.

¹²Z. B. in [Pat01] oder [VHL03]

¹³[Hei06]

¹⁴Im Folgenden als *lokale Änderungsplanung* bezeichnet

¹⁵Im Folgenden als *globale Änderungsplanung* bezeichnet

1. Einleitung

Bei der globalen Änderungsplanung müssen, je nach Art der für die Zusammenarbeit geschlossenen *Leistungsvereinbarungen* zwischen den Partnern, die Zielfunktionen des anfragenden¹⁶ Partners wie auch des antwortenden¹⁷ Partners berücksichtigt werden. Dieses führt zu Koordinationsaufwand zwischen den beteiligten Partnern, der im Interesse einer effizienten Steuerung durch die Minimierung der notwendigen Koordinationsschritte bis zur Bestätigung einer Änderung möglichst gering gehalten werden muss. Der effizienteste Weg dieses zu erreichen besteht darin, Materialströme und Betriebsmittel nur in Höhe der erforderlichen Änderung anzupassen und so einen ungültigen in einen gültigen Plan zu überführen.

Änderungsplanung im industriellen Umfeld

Soll die Steuerung von Produktionsnetzwerken durch die Änderungsplanung effizient umgesetzt werden, so ist die weitestmögliche *Automatisierung* des Änderungsplanungsprozesses erstrebenswert. Ziel ist es, Daten schnell zu übermitteln, erforderliche Abstimmungsprozesse zwischen Partnern zu beschleunigen und unabhängig von Arbeitszeiten und sonstigen limitierenden Faktoren zu gestalten.

Grundvoraussetzung hierfür ist, dass alle notwendigen Daten maschinell lesbar bereitgestellt werden. Hierzu gehören Plandaten als dynamische Ausgangsdaten, Maschinenkapazitäten, Übergangszeiten zwischen Produktionsstufen oder Lieferbedingungen zwischen Partnern im Produktionsnetzwerk als Stammdaten. Der Austausch und Abgleich solcher Daten ist heutzutage durch die Nutzung integrierter ERP-Systeme¹⁸ wie SAP¹⁹, SAGE Bäurer²⁰ oder NAV²¹ in der Industrie technisch ein geringes Problem.²² Diese Daten sind durch die Vernetzung der ERP-Systeme über das Internet mittels standardisierter Schnittstellen im Prinzip immer und überall verfügbar.²³

Schwieriger zu automatisieren ist die Entscheidung, ob, wie und wann welches Änderungsplanungsverfahren angewendet werden soll. Die Anwendung implementierter Änderungsplanungsverfahren²⁴ erfordert a priori die Festlegung des Anwendungsfokusses, ohne dass letztendlich eine effiziente Anwendung der Verfahren zur Steuerung der Änderungsplanung sichergestellt ist. Dieses trifft insbesondere auf die globale Änderungsplanung zu, weil bei der Festlegung des jeweiligen anzuwendenden

¹⁶Initiator

¹⁷Partizipant

¹⁸ERP: Enterprise Resource Planning. Z. B. MRP I, MRP II etc. Siehe z. B. [DW97a], [Sch05]

¹⁹<http://www.sap.com>

²⁰<http://www.sagebaeurer.de/>

²¹Vorher bekannt als NAVISION: <http://www.microsoft.com/germany/dynamics/nav/default.aspx>

²²Das Problem besteht in der Weigerung der Partner, Daten flexibel auszutauschen. Siehe z. B. [CUGI05].

²³Dessen ungeachtet ist die Konzeption und Umsetzung von Schnittstellen zwischen ERP-Systemen kostenintensiv und zeitaufwendig.

²⁴Z. B. in SAP-APO aus dem Produkt SAP SCM:<http://www.sap.com/germany/solutions/business-suite/scm/index.epx>

Änderungsplanungsverfahrens zusätzlich zu lokalen Bedingungen die Wechselwirkungen zwischen den beteiligten Partnern im Produktionsnetzwerk berücksichtigt werden müssen.

Änderungsplanung in der anwendungsorientierten Forschung

Im Gegensatz zur industriellen Anwendung existieren im Bereich der anwendungsorientierten Forschung des Supply Chain Managements verschiedene prototypische Ansätze zur automatisierten und *kooperativen*²⁵ *Änderungsplanung* durch Multiagentensysteme.²⁶ Die *Agenten* verhandeln kooperativ und automatisiert über die Material- und/oder Kapazitätsbedarfe im Produktionsnetzwerk.

In einer aktuellen Arbeit stellt Heidenreich mit MASCOPP einen solchen Ansatz vor.²⁷ Kern dieses Konzeptes ist die Klassifizierung von geeigneten Planungsverfahren zur *Änderungsplanung* durch *Planungsstrategien*. Eine Planungsstrategie definiert, welche Art von Änderungsplanungsverfahren für ungültige Zustände nach einem Ereignis angewendet werden kann und schränkt dadurch die Anzahl möglicher Planungsalternativen ein. Heidenreich beschreibt eine umfassende Anzahl elementarer Änderungsplanungsverfahren, durch die einzeln oder durch deren Kombination ungültige Zustände aufgelöst werden können.

Um die oben beschriebene Umsetzungslücke der Zuordnung von Änderungsplanungsverfahren zur Beseitigung ungültiger Zustände zu schließen, definiert Heidenreich eine *Regelsprache*, die eine solche Zuordnung ermöglicht. Regeln der Form

```
WENN Zustand Z  
DANN WENDE AN Änderungsplanungsverfahren A  
IN Variante V
```

realisieren so a priori die manuellen Zuordnungen von ungültigen Zuständen zu Änderungsplanungsverfahren. Diese stehen dann für die automatisierte Steuerung von Produktionsnetzwerken durch eine *Änderungsplanung* zur Verfügung und werden auch als *Steuerungsregeln* bezeichnet.

Problem der Regelformalisierung

Produktionsnetzwerke können quasi unendlich viele ungültige Zustände annehmen, die in der *Änderungsplanung* zu verarbeiten sind und für die entsprechende Regeln zur

²⁵Kooperativ meint hier die Zusammenarbeit zur Erreichung eines gemeinsamen Zieles, nämlich der Beseitigung ungültiger Zustände.

²⁶Z. B. Agile Agent Control Environment [Sti98], Planet AS [Man97], X-Cittic [Dud99], kollaborative *Änderungsplanung* in Unternehmensnetzwerken [Bus04], CoagenS [Pap06]

²⁷[Hei06]

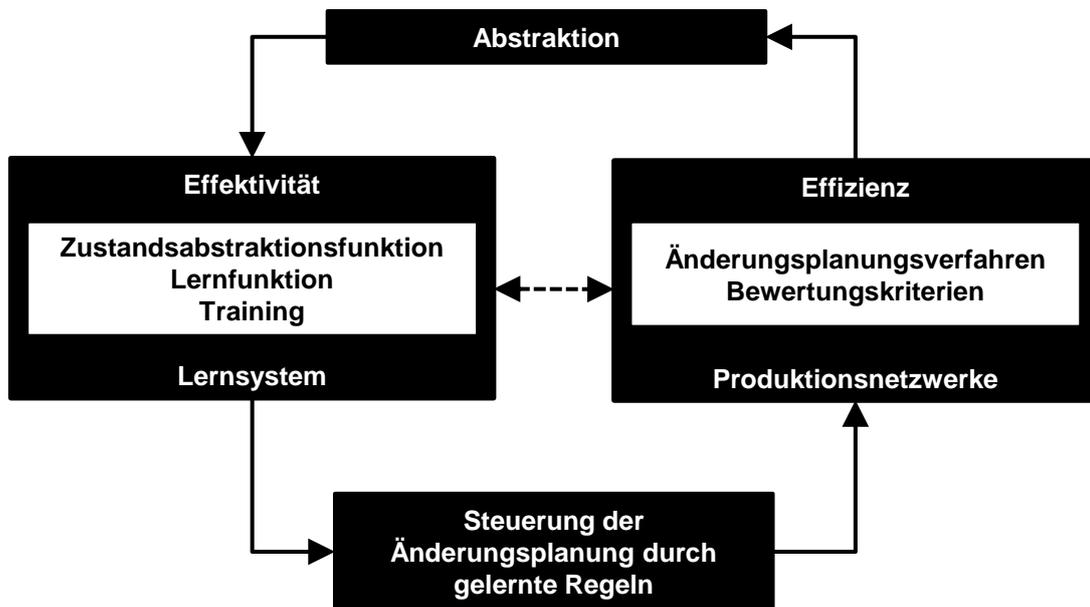


Abbildung 1.1.: Effektivität und Effizienz durch problemspezifische Abstraktion

effizienten Steuerung der Änderungsplanung formalisiert werden müssten. Die manuelle Erstellung eines Regelsystems wäre demzufolge sehr aufwendig, da durch einen Planer zunächst relevante Zustände ausgewählt und dann entsprechende Regeln bestimmt werden müssten. Verwendet jeder Partner im Produktionsnetzwerk sein eigenes Regelsystem, so ist zusätzlich die partnerübergreifende Abstimmung der Regeln notwendig. Heidenreich fasst dieses wie folgt zusammen: “Offen ist jedoch, welche Konfigurationen [des Regelsystems] unter welchen Bedingungen am besten geeignet sind.”²⁸

Ziel der Arbeit - Automatisierung der Regelerstellung durch Nutzung von maschinellem Lernen

In dieser Arbeit soll anhand der Verknüpfung von Methoden aus der Informatik (Künstliche Intelligenz) und den Wirtschaftswissenschaften (Produktionsplanung und -steuerung) ein anwendungsorientierter Ansatz zur Steuerung der Änderungsplanung in Produktionsnetzwerken konzipiert werden. Die zentrale Problemstellung für diese Arbeit lautet: Welche Möglichkeiten bestehen, um ein Regelsystem zur automatisierten Steuerung der Änderungsplanung in Produktionsnetzwerken mit geringem Aufwand automatisiert zu erstellen? Der in dieser Arbeit dargestellte Ansatz greift die industrielle²⁹ und forschungsbezogene Diskussion³⁰ zu diesem Thema auf und versucht, durch

²⁸Ebd. S. 180

²⁹Siehe [DDKT07] und [DN07]

³⁰Siehe [Hei06]

die Verwendung eines maschinellen Lernsystems einen ersten Beitrag zu leisten, ein solches Regelsystem automatisiert zu lernen.

Der Lernprozess soll *effektiv*³¹ durchgeführt werden. Dieses bedeutet, dass die zur Steuerung der Änderungsplanung relevanten Regeln gelernt werden. Ebenso soll der Lernprozess *effizient*³² sein. Es sollen trotz großem Zustandsraum von Produktionsnetzwerken Regeln schnell gelernt und Steuerungsregeln erzeugt werden können.

Aus dem angestrebten Ziel folgen drei wesentliche Forschungsfragen für diese Arbeit:

1. Wie kann der Zustandsraum ungültiger Zustände eines Produktionsnetzwerkes auf ein handhabbares Maß reduziert bzw. abstrahiert werden, sodass effektiv und effizient Regeln gelernt werden können?
2. Wie ist die Lernfunktion zu definieren, sodass auf einem abstrahierten Zustandsraum effektiv, also anwendungsbezogen, Regeln zur Steuerung der Änderungsplanung gelernt werden können?
3. Wie ist das Lernsystem mit welchen Ausgangsdaten zu trainieren?

Struktur der Arbeit

Kapitel 2 grenzt den Untersuchungsgegenstand der kooperativen Steuerung in Produktionsnetzwerken ab und erläutert die spezifischen Herausforderungen der automatisierten Erstellung von Steuerungsregeln für die Änderungsplanung für Produktionsnetzwerke der Serienfertigung durch ein maschinelles Lernsystem. Es wird diskutiert, welche Teilprobleme für diesen Komplex insgesamt gelöst werden müssen und welche Teilprobleme durch diese Arbeit aufgelöst werden können. Die herausgestellten Anforderungen werden abschließend als Forschungsfragen dieser Arbeit zusammengefasst.

In Kapitel 3 folgt die Analyse aktueller Forschungsergebnisse aus den Bereichen der Zustandsabstraktionsverfahren für Produktionsnetzwerke und angewendeten maschinellen Lernverfahren zur Steuerung von Produktionsnetzwerken im Hinblick auf deren Eignung zur Lösung der Forschungsfragen aus Kapitel 2. Aus der vorhergegangenen Diskussion in Kapitel 2 und 3 leitet sich logisch die in der Konzeption zu leistende Arbeit ab, welche in Kapitel 4 strukturiert dargestellt wird.

In Kapitel 5 wird das Konzept zur Reduktion des Zustandsraumes, der Aufbau der Lernfunktion und der Ablauf des Trainings und der dazu notwendigen Ausgangsdaten

³¹*Effektivität*: „Ausmaß, in dem geplante Tätigkeiten verwirklicht und geplante Ergebnisse erreicht werden“ (DIN 9000:2000). „Ein Verfahren ist *effektiv*, wenn es durch einen Algorithmus Schritt für Schritt in eindeutiger Weise festgelegt ist und in jedem Falle der Durchführung nach einer endlichen Anzahl von Schritten eine Lösung ergibt.“ ([KB87], S. 288)

³²*Effizienz*: „Verhältnis zwischen dem erzielten Ergebnis und den eingesetzten Mitteln“ (DIN 9000:2000)

1. Einleitung

beschrieben. Es wird erläutert, wie die Steuerungsregeln aus den Trainingsergebnissen des Lernverfahrens generiert und angewendet werden können. In Kapitel 6 erfolgt anhand exemplarischer Szenarien die Validierung der erarbeiteten Konzepte aus Kapitel 5. Kapitel 7 zieht ein Fazit bzgl. der Validierungsergebnisse und stellt neben der kritischen Würdigung der Grenzen der Arbeit potenzielle Erweiterungsmöglichkeiten der erarbeiteten Konzepte dar.

2. Problemstellung

Das Problem zu erkennen ist wichtiger, als die Lösung zu erkennen, denn die genaue Darstellung des Problems führt zur Lösung.

(Albert Einstein)

Zielstellung der Arbeit ist das Lernen von Regeln zur Steuerung der Änderungsplanung in Produktionsnetzwerken. Der Aufbau von Produktionsnetzwerken und die darin stattfindenden Planungsprozesse und deren Steuerung sind die Untersuchungsgegenstände dieser Arbeit. Diese werden in Kapitel 2.1 abgegrenzt. Produktionsnetzwerke weisen durch die Vielzahl möglicher Ausprägungen operativer Pläne quasi unendlich viele Zustände auf. Die Verknüpfung aller Zustände mit Planungsverfahren zu Steuerungsregeln ist eine nahezu unlösbare Aufgabe. Um Regeln lernen zu können, muss der Zustandsraum auf ein handhabbares Maß verdichtet werden. Die entstehenden Herausforderungen werden in Kapitel 2.2 diskutiert. Aus dem Ergebnis der Diskussionen in diesem Kapitel leiten abschließend die Forschungsfragen dieser Arbeit ab. Die Kombination aus Untersuchungsgegenstand und dessen spezifischer Eigenschaften einerseits und der Herausforderungen des Lernens der Regeln andererseits, ermöglichen die Formulierung der zentralen Forschungsfragen dieser Arbeit, die in Kapitel 2.3 dargestellt werden.

2.1. Steuerung von Produktionsnetzwerken der Serienfertigung

Die Analyse geht von Unternehmen aus, deren ökonomische Aktivitäten auf den Bereich der *Produktion* beschränkt sind. Diese sind *Produktionsunternehmen*.¹ Mit dem Begriff *Produktion* werden in Produktionsunternehmen stattfindende Prozesse bezeichnet. Aufgabe der Produktion ist es, bei einem gegebenen Input einen spezifischen Output zu erzeugen.

¹Für weiterführende Informationen zu Art und Ausgestaltung von Produktionsunternehmen, besonders aus dem Bereich der Automobilindustrie, sei z. B. auf [LHNH00] und [Wer00] verwiesen.

2. Problemstellung

Definition 2.1 (Produktion) *Produktion ist „die sich in betrieblichen Systemen vollziehende Bildung von Faktorkombinationen im Sinne einer Anwendung technischer oder konzeptioneller Verfahren zur Transformation der dem Unternehmen zur Verfügung stehenden originären und derivativen Produktionsfaktoren in absetzbare Leistungen oder in derivative Produktionsfaktoren [...], die in weiteren Faktortransformationsprozessen unmittelbar genutzt oder in absetzbare Leistungen transformiert werden, um das Sachziel unter der Maßgabe der Formalziele zu erfüllen.“²*

Produktionsunternehmen können sich zur gemeinsamen Herstellung, z. B. komplexer Erzeugnisse für Kraftfahrzeuge, zu einem *Produktionsnetzwerk* zusammenschließen, um dadurch strategische Vorteile am Markt zu erzielen.

Definition 2.2 (Produktionsnetzwerk) *Ein Produktionsnetzwerk kennzeichnet sich durch die koordinierte Zusammenarbeit mehrerer rechtlich selbstständiger Unternehmen bei Beschaffungs-, Herstellungs- und Lieferprozessen zur Erzielung von Wettbewerbsvorteilen.³ Bei der Modellierung eines Produktionsnetzwerkes werden „einzelne Produktionsprozesse sowie deren strukturelle Abhängigkeiten in Form eines Netzwerkes“ abgebildet.⁴*

Bei der Art der Koordination des Materialflusses innerhalb eines Produktionsnetzwerkes können nach Wildemann⁵ zwei grundlegende idealtypische Ausprägungen unterschieden werden. Die Koordination des Materialflusses innerhalb eines hierarchisch-pyramidal ausgerichteten Netzwerkes wird durch die Vorgaben eines dominanten fokalen Unternehmens bestimmt. Polyzentrisch ausgerichtete Produktionsnetzwerke zeichnen sich aufgrund bestehender Abhängigkeiten zwischen den beteiligten Produktionsunternehmen durch ein partnerschaftliches und gleichberechtigtes Koordinieren des Materialflusses innerhalb des Produktionsnetzwerkes aus.⁶

In dieser Arbeit werden polyzentrisch ausgerichtete Produktionsnetzwerke betrachtet. Die Zusammenarbeit der Partner wird über Rahmenverträge und darin festgelegten Leistungsvereinbarungen zwischen logistisch verbundenen Partnern geregelt. Durch die Leistungsvereinbarungen wird die Planung des Materialflusses erleichtert und die Planungssicherheit innerhalb des Produktionsnetzwerkes erhöht.⁷ Die Wertschöpfung in einem Produktionsnetzwerk kann als ein über Leistungsvereinbarungen geregelter Materialfluss, als Kette sequenzieller Arbeitsfolgen zur Herstellung eines Erzeugnisses durch Wertschöpfungsprozesse, angefangen vom Rohstoff über Halbfabrikate bis zum Enderzeugnis, modelliert werden. Diese Arbeitsfolgen können, je nach Art des

²[Cor00], S. 2. Weiteres siehe z. B. in [Web91]

³Vgl. [LW00], S. 193

⁴[Kla99], S. 9

⁵[Wi97]

⁶Weiteres siehe z. B. in [Bus04], [Sch02], [Wi97]

⁷Anmerkungen zum Nutzen und Zweck von Rahmenverträgen siehe z. B. [Hei06], S. 6.

Produktionsprozesses, in *Fertigungsstufen*⁸ (FST) gegliedert werden. Die Gliederung der Supply Chain in abgrenzbare Fertigungsstufen mit unterschiedlichen Fertigungsverfahren grenzt den Untersuchungsgegenstand auf diskrete bzw. diskontinuierliche⁹ Produktionsprozesse ein. Die Produktionsprozesse der einzelnen Partner bzw. Fertigungsstufen des Produktionsnetzwerkes finden in *Produktionssystemen*¹⁰ statt.

2.1.1. Klassifikation des Untersuchungsgegenstandes

Die Anwendung der in dieser Arbeit konzipierten Verfahren ist möglich, wenn das betrachtete Produktionssystem der oben durchgeführten allgemeinen Abgrenzung des Untersuchungsgegenstandes entspricht. Dessen spezifischen Merkmale werden zur weiteren Abgrenzung durch Einordnung in ein Klassifikationsschema im Folgenden detaillierter differenziert. Eisenführ¹¹ unterscheidet Produktionssysteme durch die Merkmale Produktionsprogramm und Produktionssystem. Nach einer Analyse von Kuhn¹² orientiert sich das Klassifikationsschema von Eisenführ¹³ aus Tabelle 2.1 an den Merkmalen der Art des Produktionsprogramms und dem Aufbau des Produktionssystems.

Als Beispiel für ein typisches der Klassifikation entsprechendes Produktionsnetzwerk dient das auf Ottomotoren eingeschränkte, dreistufige Produktionssystem für Motoren (Vor- und Endmotormontage) und Teile (Teilefertigung) eines großen deutschen Automobilherstellers. Es werden standardisierte Fließgüter (Teile, Rumpfmotoren und Motoren) in Serien¹⁴- bzw. Großserienfertigung produziert. Die Produktion erfolgt sortenbasiert, wobei aufgrund der Ähnlichkeiten der gefertigten Produkte Rüstzeiten sowie Durchlaufzeiten der einzelnen aufgelegten Sorten zu vernachlässigen sind. Die bereitgestellten Kapazitäten der einzelnen Fertigungsstufen werden für alle Sorten gleich behandelt. Die Fertigungsstufen (Teilfertigung, Rumpfmontage, Endmontage) sind dabei unverbunden, d. h. durch Puffer voneinander entkoppelt. Es handelt sich um eine Fließfertigung. Die Ergebnisse dieser Arbeit können z. B. auch auf das Produktionssystem eines großen deutschen Systemlieferanten für Bremsen und Reifen oder andere

⁸„Die Fertigung umfasst alle technischen Maßnahmen zur Herstellung von Material oder Erzeugnissen. Sie ist grundsätzlich ein diskontinuierlicher Prozess.“ ([Dan99], S. 3)

⁹Eine Unterscheidung von diskontinuierlichen und kontinuierlichen Prozessen findet sich in [Bec91].

¹⁰Ein *System* ist „das Zusammengesetzte, Zusammenstellung (zu einem Ganzen). Geordnete Mannigfaltigkeit irgendwelcher (materiellen oder ideellen) Objekte. [...]“ ([KB75], S. 1199 ff.) Eine ähnliche Definition ist in [HM88] nachzulesen.

„Ein *Produktionssystem* ist die Summe aller Arbeitsmittel in einem festgelegten Bereich, dessen Aufgabe es ist, am Eingang in das System eingehende Materialien in einen definierten Ausgangszustand zu transformieren und am Ausgang abzugeben.“ (VDI Richtlinie 3633 [VDI93])

¹¹Siehe [Eis89]. In der Literatur werden eine Vielzahl von Klassifikationsschemata für Produktionssysteme wie z. B. [GO74] beschrieben. Kuhn analysiert diese und weitere in seiner Arbeit [Kuh99].

¹²Ebd., S. 29

¹³Siehe [Eis89]

¹⁴„Die Serienfertigung stellt die wiederholte Produktion einer bestimmten Stückzahl (Serie) eines Erzeugnisses dar.“ ([Sch96], S. 10)

2. Problemstellung

Tabelle 2.1.: Organisatorische Einordnung des Untersuchungsgegenstandes in das Klassifikationsschema nach Eisenführ

Gliederungskriterium	Merkmale	Merkmalsausprägungen	Relevant
Produktionsprogramm	Sachgüter	- Stückgüter	-
		- Fließgüter	X
	Produktart	- Kundenspezifische Produkte	-
		- Standardprodukte	X
	Wiederholungsgrad	- Einzelfertigung	-
- Serienfertigung		X	
- Großserienfertigung		-	
- Massenfertigung		-	
Produktionsbreite	- Einproduktbetrieb	-	
	- Sortenfertigung	X	
	- Partie-/Chargenfertigung	-	
	- Mehrproduktfertigung	-	
Erzeugungstechnische Interdependenzen	- Unverbundene Produktion	X	
	- Konkurrierende Produktion	-	
	- Kupplungsproduktion	-	
Produktionssystem	Produktorientierte Systeme	- Baustellenfertigung	-
		- Fließfertigung	X
		- Bestellproduktion	-
		- Lagerproduktion	-
	Verfahrensorientierte Systeme	- Werkstattfertigung	-
	Automatisierung und Flexibilität des Produktionssystems	- Industrieroboter	-
		- NC-Maschinen	-
		- Bearbeitungszentren	-
		- Flexible Fertigungszellen	-
		- Flexible Fertigungssysteme	-
- Flexible Fertigungsstraße	-		

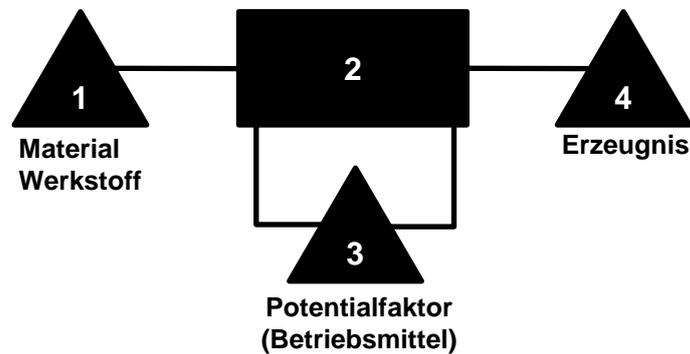


Abbildung 2.1.: Grafische Repräsentation eines elementaren MFERT-Modells. (1) zeigt einen Fertigungsobjektknoten, (2) einen Fertigungsprozessknoten und (3) einen Kapazitätsobjektknoten.

Hersteller ähnlicher Produkte übertragen werden, solange das Produktionssystem der obigen Klassifikation genügt.

2.1.2. Objekte in einem Produktionsnetzwerk

Ein Produktionsnetzwerk kann durch Objektknoten mit der Modellierungsmethode MFERT, entwickelt von Schneider et. al.¹⁵, modelliert werden. In MFERT wird zur Modellierung von Produktionsnetzwerken und deren Produktionssystemen zwischen Fertigungsobjektknoten¹⁶ (FOK), Kapazitätsobjektknoten¹⁷ (KOK) und Prozessknoten¹⁸ (PK) unterschieden. Abbildung 2.1 zeigt einen elementaren MFERT-Graphen.

Fertigungsobjektknoten modellieren *Material*¹⁹, Kapazitätsobjektknoten modellieren verfügbare *Betriebsmittel* und Prozessknoten modellieren *Produktionsprozesse*, die Materialien mit gegebenen Betriebsmitteln²⁰ in eines oder mehrere Erzeugnisse transformieren.²¹ Prozessknoten modellieren die Belegung der Betriebsmittel in Abhängigkeit der in den Produktionsprozess eingehenden Materialmenge und den zur Verfügung stehenden Betriebsmittelkapazitäten.

¹⁵Z. B. in [DW97b], [DW97a], [DW93] oder [Sch96]

¹⁶Dreieck vor oder hinter einem Prozessknoten

¹⁷Dreieck unterhalb eines Prozessknotens

¹⁸Viereck

¹⁹[REF91], S. 62

²⁰Betriebsmittel umfassen hier Arbeitsmittel (Geräte und Maschinen), die in dem Produktionssystem aktiv daran beteiligt sind, das Material gemäß der Arbeitsaufgabe vom Eingangs- in den Zielzustand zu versetzen [REF84], als auch menschliche Arbeitskräfte.

²¹In dieser Arbeit jeweils ein Erzeugnis

2. Problemstellung

Die Fertigungsobjektknoten der vorgelagerten Fertigungsstufe eines Prozessknotens werden im Rahmen dieser Arbeit als *Lieferanten* und die jeweils nachgelagerten Fertigungsobjektknoten als *Kunden* bezeichnet. Der Prozessknoten modelliert die Fertigungsstufe im Materialfluss. Die Modellierung alternativer oder komplementärer Lieferanten von Material oder alternativen Betriebsmitteln sowie alternativer oder komplementärer Kunden je Fertigungsstufe ist möglich.²² Modelle von Produktionsnetzwerken können zwischen Objektknoten die in Tabelle 2.2 dargestellten alternativen und komplementären Verknüpfungen aufweisen. Um zeitliche Abläufe im Produk-

Tabelle 2.2.: Zulässige Verknüpfungen im Modell der Fertigung

PK	FOK (Zugang)	FOK (Abgang)	KOK (Zugang)
Abgang	\wedge, \vee	—	—
Zugang	—	\wedge, \vee	—
Transformation	—	—	\vee

tionsnetzwerk modellieren zu können, wird in dieser Arbeit ein diskretes Zeitmodell verwendet. Jedem Objektknoten wird ein Plan P , bestehend aus Zeitabschnitten zugewiesen.

Definition 2.3 (Plan) *Der Plan eines Objektknotens hat für jeden Zeitabschnitt innerhalb des Planungshorizontes $PHZ = 1, \dots, PH$, $PH \in \mathbb{N}$, jeweils repräsentiert durch eine Periode $p(k)$, einen Planwert. Die Planbestandsrechnung²³ ermittelt für Fertigungsobjektknoten aus dem bestehenden Planwert einer Periode und unter Berücksichtigung des periodenspezifischen Zugangs (Angebot) und Abgangs (Bedarf) der Periode einen neuen Planwert der folgenden Perioden. Die Verrechnung von Zu- und Abgängen erfolgt zu Beginn einer Periode.²⁴ Für Kapazitätsobjektknoten ermittelt die Plankapazitätsrechnung²⁵ einen neuen Planwert der folgenden Perioden aus der Verrechnung des zugewiesenen Kapazitätsangebotes und dem offenen Kapazitätsbedarf.²⁶ Ein Plan wird allgemein als PH -dimensionaler Vektor P mit Planwerten $p(k)$ für jede Planungsperiode $k \in \{1, \dots, PH\}$ dargestellt. Der diskrete Bestand oder die genutzte Kapazität in einer Periode kann nicht negativ, aber 0 sein.*

Im Fall von Fertigungsobjektknoten repräsentiert der Plan die zur Produktion bereitstehende Materialmenge in Stück [ST] je Periode. Diese wird als Bestand $p(k)^{sup}$ bezeichnet. Der Bestand ist durch Restriktionen auf minimal $p(k)_{min}^{sup}$ und maximal $p(k)_{max}^{sup}$ mögliche Bestandsmenge innerhalb der Perioden eines Planes beschränkt. Für Kapazitätsobjektknoten repräsentiert der Plan das zugewiesene Kapazitätsangebot²⁷

²²Vgl. [Sch96]

²³Siehe [Hei06], S. 190 ff.

²⁴Siehe [Sch96]. Zugang und Abgang sind Ereignistypen in MFERT.

²⁵Siehe [Hei06], S. 197 ff.

²⁶Siehe [Hei06], S. 112

²⁷Z. B. Mensch- oder Maschinenkapazität

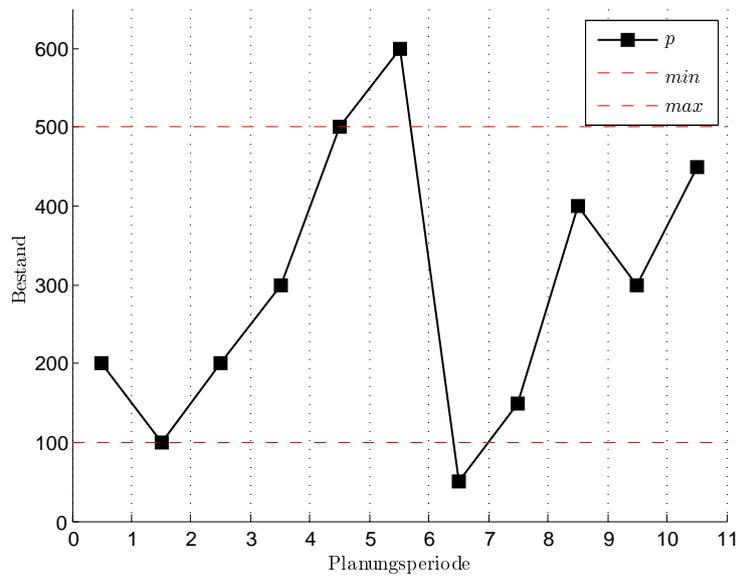


Abbildung 2.2.: Beispielplan für einen FOK

für die zur Produktion stehende Menge von Erzeugnissen in einer Periode in Stück. Die für einen Produktionsprozess eingeladete Kapazität wird als zugewiesenes Kapazitätsangebot $p(k)^{pl}$ bezeichnet. Die verfügbare Kapazität eines KOK zur Deckung von Kapazitätsbedarfen des Produktionsprozesses ist durch eine minimal auszulastende Kapazität $p(k)_{min}^{pl}$ und den verfügbaren Leistungsgrad²⁸ $p(k)_{max}^{pl}$ beschränkt. Der Plan des Prozessknotens repräsentiert die Belegung eines Produktionsprozesses je Periode abhängig vom Zu- und Abgang an Material und den für den Produktionsprozess bereitgestellten Betriebsmittelkapazitäten. Pläne sind als Diagramm darstellbar, wie in Abbildung 2.2 beispielhaft für den Plan eines FOK zeigt.²⁹

Für FOK ist eine Planänderung (Veränderung des geplanten Zu- und Abgangs) für einen Plan ein *Ereignis*. Die Art einer Planänderung wird je nach Planungsrichtung vorwärts als Angebotsänderung oder rückwärts als Bedarfsänderung bezeichnet. Für FOK werden durch die Planbestandsrechnung die zur Disposition stehenden Angebote und Bedarfe in den Plan eingerechnet und so aus Bruttobedarfen und Bruttoangeboten vom Planbestand abhängige Nettobedarfe und Nettoangebote ermittelt.³⁰ Für KOK

²⁸ „Der Leistungsgrad entspricht einem Sicherheitspuffer, indem die verfügbare Kapazität nicht vollständig verplant wird.“ (Ebd., S. 198)

²⁹ Aus technischen Gründen wird der Planwert in den Grafiken in der Mitte einer Periode angezeigt. Der jeweils rechte Skalenswert der x-Achse eines Funktionswertes repräsentiert die zugewiesene Periode dieses Funktionswertes. Unabhängig davon werden alle Zu- und Abgänge einer Periode an deren Beginn mit dem Planwert verrechnet.

³⁰ Ein Angebot ist eine „verbindliche Antwort auf eine Anfrage, die ein Lieferant seinem potenziellen Auftraggeber zusendet.“ ([BKP05], S. 7) Aus Sicht des Materialflusses ist ein Angebot die Menge an Material, die ein Lieferant in einer bestimmten Periode an einen Kunden liefern kann. Es wird zwischen Brutto- und Nettoangebot unterschieden. „Bedarf bezeichnet eine bestimmte [Menge] an

2. Problemstellung

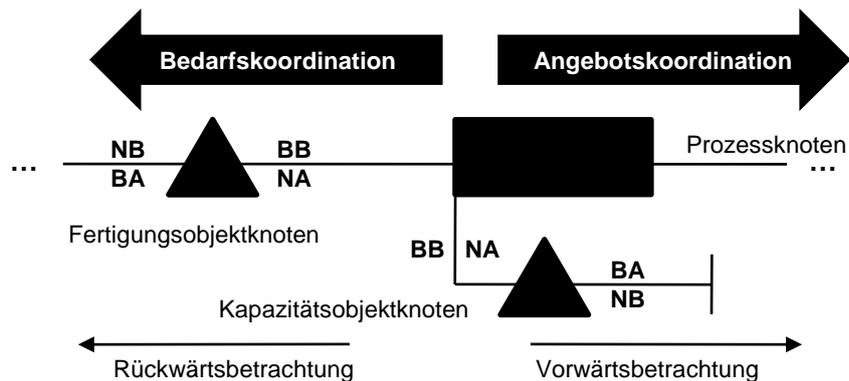


Abbildung 2.3.: Arten der Kunden-Lieferanten-Beziehungen

werden Änderungen des Nettoangebotes oder Bruttobedarfs an Kapazität als Ereignisse bezeichnet.³¹ Abbildung 2.3 verdeutlicht diese Begriffe.

Pläne eines FOK modellieren hier stets eine einzelne Materialart, die im Produktionsprozess verwendet wird. Die Pläne der KOK modellieren exklusiv die bereitgestellte Kapazität für einzelne Produkte je Periode. Das heißt, eine Betrachtung konkurrierender Kapazitätsbedarfe für unterschiedliche Erzeugnisse im Produktionsprozess innerhalb eines KOK entfällt. Für die Motorenfertigung bedeutet dieses beispielsweise, dass zwar unterschiedliche Motoren zur Produktion eingeplant werden, aber die Menge möglicher zu produzierender Motoren je Periode bei entsprechendem Kapazitätsangebot gleich ist, z. B. 500 Motoren pro Schicht.³²

Muss z. B. eine Änderung des zugewiesenen Kapazitätsangebotes am KOK durchgeführt werden, so kann dies unabhängig von der Materialart geschehen. Entscheidend ist die Durchführbarkeit der Änderung als Erhöhung oder Senkung des zugewiesenen Kapazitätsangebotes innerhalb des verfügbaren Leistungsgrades des angefragten KOK. Bei erfolgter Änderung wird entsprechend des neuen Planes am KOK die Belegung des Prozessknotens angepasst. Z. B. können nach einer Erhöhung des Leistungsgrades eines KOK mehr Erzeugnisse je Periode im PK produziert werden.

Das Ergebnis der unter Berücksichtigung von Zu- und Abgängen je Periode durchgeführten Planbestands- oder Plankapazitätsrechnung³³ nach einem Ereignis ist ein *Zustand* eines Objektknotens in der Planung der Produktion.

benötigtem [Material]. Man unterscheidet zwischen Bruttobedarf und Nettobedarf.“ ([BKP05], S. 17. Siehe auch [KK04], S. 39)

³¹Die Änderungen von Kapazitätsnettobedarfen werden nicht betrachtet, da keine Verhandlungsmöglichkeiten im Rahmen einer kooperativen Änderungsplanung möglich sind. (Vgl. [Hei06], S. 111)

³²Diese Einschränkung fußt auf Erfahrungen aus einem Projekt des Fraunhofer Anwendungszentrums für logistikorientierte Betriebswirtschaft (FhG-ALB) mit einem deutschen Automobilhersteller.

³³Siehe Planungsverfahren in [Hei06]

Definition 2.4 (Zustand) Ein Zustand eines Objektknotens ergibt sich durch Planbestands- oder Plankapazitätsrechnung auf einem bestehenden Plan, deren Ergebnis von der Art der Planänderung als Änderung des Zu- oder Abgangs, dem anfragenden Objektknoten, dem Zeitpunkt der Planänderung als Planungsperiode und der Höhe der Planänderung als Menge abhängt.

Die Beschränkungen der Pläne der Objektknoten werden als *Restriktionen* bezeichnet. Aus ihnen ergibt sich, dass auch Zustände in einem Produktionsnetzwerk Restriktionen unterliegen. Diese Restriktionen können sowohl für *alle* Perioden des Planungshorizontes gleich definiert, als auch speziell für *einzelne* Perioden des Planungshorizontes angepasst werden.

Für Materialbeschaffungsprozesse zwischen vor- und nachgelagerten FOK eines PK sind Beschaffungspläne definiert, in denen je Periode festgelegt wird, wie viel Material $p(k)^{mf}$ je Periode zwischen den entsprechenden Objektknoten zur Befriedigung von Nettobedarfen oder Nettoangeboten als Zu- oder Abgang fließt. Die Festlegung eines minimalen oder maximalen Materialflusses $p(k)_{min}^{mf}$ oder $p(k)_{max}^{mf}$ zwischen Objektknoten je Periode wird als *Leistungsvereinbarung* bezeichnet

Die Modellierung von Übergangszeiten zwischen einem aus Sicht des PK vorgelagerten und nachgelagerten FOK und PK setzt sich zusammen aus der Zeit, die zum Transport des Materials vom vorgelagerten FOK zum PK und vom PK zum nachgelagerten FOK benötigt wird, zuzüglich der Durchlaufzeit des Materials durch den Produktionsprozess. Diese Zeit wird als Vorlaufzeit bezeichnet und wird bei Beschaffungsprozessen zur Umterminierung von Bedarfs- oder Angebotsterminen verwendet. Diese Umterminierung wird als *Vorlaufzeitverschiebung* (LTS^{34}) bezeichnet und in Perioden angegeben.

In der Beschaffungssteuerung³⁵ wird die Steuerung des Materialflusses zwischen Kunden und Lieferanten festgelegt. Dabei wird unterschieden, ob Bedarfs- oder Angebotsänderungen nach dem *Bestellpunkt-* oder *Bestellzyklusverfahren* disponiert werden.³⁶ Ein *Bestellzyklus* ist „der regelmäßig wiederkehrende Zeitabstand zwischen zwei Bestellungen“³⁷, während das Bestellpunktverfahren einsetzt, wenn „[...] der verfügbare Bestand den Meldebestand unterschreitet und dadurch eine Bestellung auslöst.“³⁸ Tabelle 2.3 zeigt eine Zusammenfassung möglicher Varianten der Beschaffungssteuerung.

Die Über- oder Unterschreitung von Restriktionen des Planes eines Objektknotens oder die Verletzung definierter Leistungsvereinbarungen zwischen zwei Objektknoten

³⁴Engl. *lead time shift*

³⁵Siehe z. B. in [Sch05], [Gud04] oder [DW97a]

³⁶Z. B. in [Hei06], S. 7 f. oder [Sch05]

³⁷[BKP05], S. 24

³⁸Ebd., S. 23

2. Problemstellung

Tabelle 2.3.: Möglichkeiten der Beschaffungssteuerung

	Lieferzyklus fest	Lieferzyklus variabel
Losgröße fest	nicht planungsrelevant	Bestellpunkt
Losgröße variabel	Bestellzyklus	freie Losgruppierung

wird als *Restriktionsverletzung* bezeichnet. Ein Plan, der eine Restriktionsverletzung aufweist, wird als *ungültiger Plan* im Gegensatz zum *gültigen Plan* ohne Restriktionsverletzungen bezeichnet.

Die Qualität eines Planes ist, bezogen auf die entstehenden Aufwände für die Bereitstellung und Nutzung von Material, Betriebsmitteln sowie der Leistungsvereinbarungen zur Beschaffung mit Kosten³⁹ quantifizierbar und dadurch auch mit anderen Plänen vergleichbar.⁴⁰ Verletzen Pläne Restriktionsgrenzen, entstehen höhere Kosten als bei Plänen ohne Restriktionsverletzungen, z. B. erhöhte Lagerkosten. Die Kostenbewertungen für Pläne orientieren sich hier an der Verletzung von Restriktionsgrenzen sowie an Kosten, die grundsätzlich für den Betrieb und Bereitstellung von Kapazitäten zur Lagerung und Produktion benötigt werden. Beschaffungsprozesse lösen Bestellvorgänge aus, die ebenso Kosten verursachen. Treten in einem Beschaffungsprozess Verletzungen der vereinbarten Leistungsvereinbarungen auf, so entstehen auch dort Kosten. Zusammenfassend können Pläne, z. B. hinsichtlich ihrer Effizienz, über Kosten miteinander verglichen werden. Hierzu müssen die vollständigen zu vergleichenden Pläne periodenweise analysiert werden.

2.1.3. Ablauf kooperativer Steuerung in Produktionsnetzwerken

Die Ausprägung des Materialflusses im Produktionsnetzwerk richtet sich nach den im Plan festgelegten Bedarfsmengen, welche durch entsprechende Angebote befriedigt werden müssen. Im Rahmen der Produktionsplanung und -steuerung (PPS)⁴¹ werden durch die herzustellenden Erzeugnismengen in den Perioden bestimmt. In der Mengenplanung werden hier entsprechende einstufige Sekundärbedarfe zwischen Objektknoten festgelegt.

³⁹Das hier verwendete Modell zu Bewertung von Kosten von Plänen ist einfach, aber zweckmäßig für die Verwendung im Lernverfahren.

⁴⁰Z. B. in [Gud04]

⁴¹„Die Produktionsplanung und -steuerung hat die Aufgabe, aufgrund erwarteter und/oder vorliegender Kundenaufträge den mengenmäßigen und zeitlichen Produktionsablauf unter Beachtung der verfügbaren Ressourcen durch Planungsvorgaben festzulegen, diese zu veranlassen, sowie zu überwachen und bei Abweichungen Maßnahmen zu ergreifen, sodass bestimmte Ziele erreicht werden.“ ([Zöp98]). Weitere Informationen zu PPS finden sich z. B. bei [Sch05], [Kur05], [Dan99], [Sch99b], [Hac84].

Das Ergebnis der Produktionsplanung ist ein Plan. Die Produktionsplanungsaufgabe⁴² gibt dabei die Rahmenbedingung für die Planung vor, Produktionsplanungsverfahren⁴³ erzeugen Pläne unter Berücksichtigung dieser Rahmenbedingungen. Wird die Planungsaufgabe ohne Berücksichtigung eines bestehenden Produktionsplanes gelöst, wird dieses als *Neuplanung* bezeichnet.⁴⁴ In dieser Arbeit wird die Planungsaufgabe stets unter Verwendung eines bestehenden Produktionsplanes gelöst, in dem Planänderungen verarbeitet werden. Dieses wird als *Änderungsplanung* bezeichnet, wobei die Änderungsplanung im Kurzfristbereich als *Steuerung* bezeichnet wird.

Definition 2.5 (Neuplanung und Änderungsplanung) „*Neuplanung bezeichnet die Vorgehensweise eines [Produktionsplanungsverfahrens], alte Solldaten zu ignorieren und lediglich auf der Basis der aktuellen Ausgangsdaten zu planen. Änderungsplanung dagegen die Vorgehensweise, auf der Basis des alten Plans und der aufgetretenen Änderungen zu planen.*“⁴⁵

Während unter dem Konzept der Änderungsplanung auch Planungsaufgaben zur Umpassung von Mengen innerhalb gültiger Pläne verstanden werden können, so liegt in dieser Arbeit der Fokus auf der Änderungsplanung zur Beseitigung ungültiger Zustände im Produktionsnetzwerk. Liegen nach erfolgter Planbestandsrechnung im Produktionsnetzwerk ungültige Zustände vor, so werden diese durch die Änderungsplanung in einen gültigen Plan transformiert.

Die Modularisierung der Produktionsplanungsaufgabe⁴⁶ ermöglicht eine objektknotenindividuelle Änderungsplanung innerhalb des Produktionsnetzwerkes durch ein entsprechendes Produktionsplanungsverfahren.⁴⁷ Es kann bereits im Vorfeld durch einen Planer ein Planungsablauf zur Steuerung der Änderungsplanung des Produktionsnetzwerkes festgelegt werden. Diese Festlegung bezeichnet Heidenreich als *Planungsstrategie*⁴⁸. Eine Planungsstrategie formuliert für ungültige Zustände eines Objektknotens unter Berücksichtigung

⁴²„Die Produktionsplanungsaufgabe ist die Aufgabe, für ein gegebenes Produktionssystem – ausgehend von gegebenen Ausgangsdaten – Plandaten, die in sich und mit den Ausgangsdaten konsistent sind für einen definierten, zielgerichteten Fertigungsprozess festzulegen, dem Fertigungsprozess vorzugeben und auf Inkonsistenzen abzu prüfen. Die gegebenen und gesuchten Daten sind dem Modell des Fertigungsgeschehens zugeordnet, wobei ein operables Modell verwendet wird.“ ([Rüt05], S. 27). Diese Definition verwendet die von Schneider aufgestellten Definitionen der Fertigungssteuerungsaufgabe (Siehe [Sch96], S. 121).

⁴³„Ein Produktionsplanungsverfahren ist ein festgelegter oder erzeugter Ablauf von Verfahrensschritten, der die Plandaten der Produktionsplanungsaufgabe so erzeugt, dass die gestellten Anforderungen der Aufgabe gelöst werden. Anforderungen beziehen sich auf Sach- und Formalziele und die Konsistenz der Lösung.“ ([Rüt05], S. 23)

⁴⁴Vgl. [Hol00], S. 14 f.

⁴⁵Ebd., S. 14

⁴⁶Die zu bearbeitende Produktionsplanungsaufgabe ist hier die Änderungsplanung.

⁴⁷Das verwendete Produktionsplanungsverfahren ist hier ein Änderungsplanungsverfahren.

⁴⁸„Eine Planungsstrategie umfasst eine Menge an elementaren Planungsstrategien, die die Produktionsplanung für einen Knoten festlegt.“ ([Hei06], S. 14.)

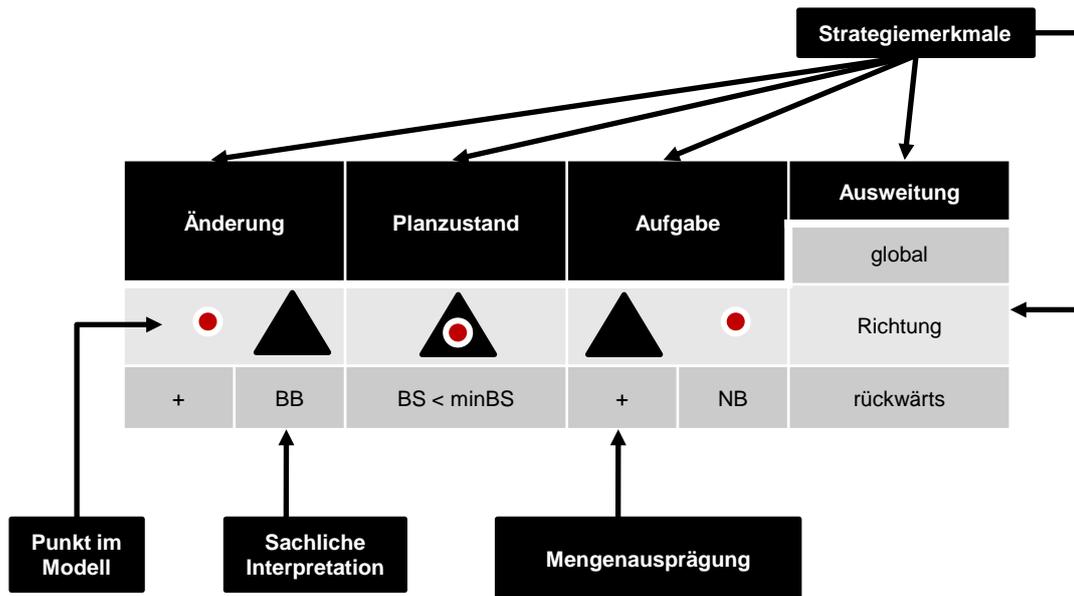


Abbildung 2.4.: Beispiel für die Klassifikation einer elementaren Planungsstrategie

- der Art der aufgetretenen Planänderung,
- dem Planzustand,
- der Planungsrichtung und
- der Ausweitung des vorhergegangenen Ereignisses

eine Planungsaufgabe für die Änderungsplanung, die durch ein entsprechend geeignetes Änderungsplanungsverfahren umzusetzen ist. Beispielhaft ist dieses in Abbildung 2.4 für einen ungültigen Planzustand nach einer Bruttobedarfserhöhung dargestellt.

Ist die Planungsstrategie bekannt, so kann ein ungültiger Zustand durch eine in dieser festgelegten Ausgestaltung einer kooperativen Änderungsplanung⁴⁹ aufgelöst werden. Dabei stehen je nach Ausweitung zwei Möglichkeiten zur Verfügung. Bei der lokalen Kompensation werden ausschließlich Planungsverfahren angewendet, deren Effekte sich auf einzelne Objektknoten beschränken, z. B. die Verwendung des Sicherheitsbestandes. Bei der globalen Kompensation erfolgt die festgelegte Planung verhandlungsbasiert zwischen den Partnern im Produktionsnetzwerk.

Es ergeben sich für lokale und globale Planungsstrategien grundlegende Planungsaufgaben:⁵⁰

- Lokal:

⁴⁹Ebd.

⁵⁰Vgl. [Hei06], S. 42

2.2. Anforderungen einer automatisierten Steuerung der Änderungsplanung durch Regeln

- *Kompensation* der Restriktionsverletzung durch knotenspezifische Änderungen der Restriktionsgrenzen
- Global:
 - *Weitergabe* der geänderten Angebote bzw. Bedarfe an benachbarte Knoten unter Beibehaltung der Koordinationsrichtung
 - Unterbreitung eines *Gegenvorschlages* an den anfragenden Knoten bei Umkehrung der Koordinationsrichtung

Um den Koordinationsaufwand bei der kooperativen Änderungsplanung zu senken, werden nur Anfragen in der benötigten Höhe als Angebote oder Bedarfe angefragt. Die Höhe einer Anfrage, sowohl an FOK als auch an KOK, wird durch die Änderungsplanungsverfahren rechnerisch ermittelt und so bei Bestätigung einer Änderungsanfrage ein ungültiger Zustand aufgelöst.⁵¹

Die Art der Ermittlung der Höhe der Änderungsanfrage, die Verteilung von Änderungsanfragen auf unterschiedliche Partner in unterschiedlicher Höhe und die unterschiedlichen ermittelten neuen Bedarfs- oder Angebotszeitpunkte sind Varianten der Änderungsplanungsverfahren. Dabei sollen die Restriktionen und Leistungsvereinbarungen berücksichtigt werden. Jede neue Variante eines anwendbaren Änderungsplanungsverfahrens in einem ungültigen Zustand bedeutet potenziell eine neue anwendbare Regel zur Auflösung dieses Zustandes.

2.2. Anforderungen einer automatisierten Steuerung der Änderungsplanung durch Regeln

In der Änderungsplanung in einem Produktionsnetzwerk kann durch die Anwendung unterschiedlicher Methoden auf ungültige Planzustände reagiert werden. Ein erneut gültiger Plan kann z. B. durch ein Optimierungsverfahren berechnet werden. Planungsaufgaben zur Auflösung eines ungültigen Planes können mithilfe erfahrungsbasierter Entscheidungen formuliert werden.⁵² In dieser Arbeit sollen Regeln genutzt werden, durch deren Anwendung die Änderungsplanung zur Auflösung ungültiger Zustände ad hoc gesteuert werden kann. Dabei soll der Entscheidungsprozess selbst, also die Auswahl einer Regel, ohne aufwendige Planungsberechnungen auskommen. Die gelernten Regeln sollen ein Änderungsplanungsverfahren derart auswählen, dass durch

⁵¹Ebd.

⁵²Berechnen von Lösungen z. B. [Tem06]. Nutzen von Erfahrungswissen [Flo98] und dessen Kombination [SS00]

2. Problemstellung

die Anwendung des gewählten Verfahrens auf einen ungültigen Zustand mit hoher Wahrscheinlichkeit ein *guter* gültiger Zustand erzeugt wird.

Im folgenden Kapitel wird analysiert, welche Voraussetzungen erfüllt sein müssen, um solche Regeln für Produktionsnetzwerke formulieren und formalisieren zu können.

2.2.1. Problem der Entscheidungsfindung

Bei einer Änderungsplanung in Produktionsnetzwerken ist die Entscheidung, mit welcher Reaktion, umgesetzt durch ein Änderungsplanungsverfahren, auf einen ungültigen Zustand reagiert wird, ein wichtiger Faktor zur effektiven Auflösung ungültiger Zustände. Um diesen Entscheidungsprozess bestmöglich durchführen zu können, müssen alle verfügbaren Informationen über den aktuellen Status der Produktion aus dem Produktionsplan berücksichtigt werden. Für lokale Pläne, z. B. innerhalb einer Fabrik, liegen in der Regel mehr Informationen über den aktuellen Planzustand vor als über die Pläne der Partner im Produktionsnetzwerk. Es ist z. B. für den Planer eines Werkes nicht zwingend bekannt, ob ein Zulieferer zeitnah angefragte Zusatzmengen an Material zum kapazitären Bedarfsausgleich für Produkte des Kunden liefern kann.

Das heißt, eine Entscheidung über eine mögliche Reaktion auf einen ereignisbedingten ungültigen Zustand muss in der täglichen Planungsarbeit in den Produktionsunternehmen oft unter unvollständiger Informationslage durchgeführt werden. Das Problem der unvollständigen Informationslage, insbesondere die Unkenntnis über den Produktionsstatus von Zulieferern oder Kunden, entspringt dem hohen Wettbewerbsdruck⁵³ und der dadurch gehemmten kooperativen Zusammenarbeit der Partner in kompetitiven, hierarchischen Produktionsnetzwerken.⁵⁴ Der Wettbewerbsdruck bestärkt eine rigide Informationspolitik zwischen den Produktionsnetzwerkpartnern und beschränkt die Möglichkeiten kollaborativer Zusammenarbeit.⁵⁵

Die Basis zur Umsetzung der erforderlichen Entscheidungen in der Änderungsplanung in Produktionsnetzwerken bildet in vielen Unternehmen die Erfahrung eines Produktionsplaners. Dieser entscheidet über den Anstoß eines geeigneten Änderungsplanungsverfahrens zur Auslösung eines ungültigen Zustandes.⁵⁶ Längere Berufserfahrung umfasst eine höhere Anzahl erfahrener Entscheidungssituationen und kann bessere Resultate für getroffene Entscheidungen ermöglichen.⁵⁷

⁵³Siehe z. B. [Lar07], [DKT07] oder EU-Projekt AC/DC [AC/10]

⁵⁴Eine Feststellung, die ein Teilprojekt von AC/DC behandelt. Näheres siehe z. B. in [DDKT07].

⁵⁵Beschreibungen der allgemeinen Situation im Bezug auf Autonomiebestrebungen der Unternehmen in der Automobilindustrie finden sich in [DKT07].

⁵⁶Die Verwendung von Erfahrung spielt grundsätzlich in Bereichen der Planung sowie im Operations Research bei schwer exakt formalisierbaren Modellen eine Rolle, wie [SM06] darstellt. Ähnliches gilt in der Logistik, wie z. B. [Flo98] erläutert.

⁵⁷Florian bezeichnet dieses Erfahrungswissen als *Informationskapital* (Vgl. [Flo98], S. 14).

2.2. Anforderungen einer automatisierten Steuerung der Änderungsplanung durch Regeln

Das Erfahrungswissen eines Planers kann bereits als implizites Regelsystem, wie es hier zur Steuerung der Änderungsplanung verwendet werden soll, betrachtet werden. Der Planer memoriert Entscheidungen für vergangene Situationen und assoziiert das Resultat der bei dieser Entscheidung durchgeführten Aktion in Kategorien wie Erfolg und Misserfolg. Das Erfahrungswissen ist netzwerkspezifisch, das heißt so lange gültig, wie die Objekte und deren Beziehungen, z. B. die Partnerstruktur und deren Leistungsvereinbarungen im Produktionsnetzwerk unverändert bleiben. Finden dort Anpassungen statt, so müssen auch Teile des Erfahrungswissens überprüft und neu gelernt werden.

Im Umfeld von Planungssystemen wird Erfahrungswissen z. B. in Decision-Support-Systemen (DSS) für die Produktionsplanung und -steuerung als Entscheidungsunterstützung für Produktionsplaner verwendet.⁵⁸ Hierzu ist es notwendig, vor der Inbetriebnahme des DSS zustandsbasierte Regeln zu hinterlegen, durch welche Entscheidungssituationen erkannt werden können, um mit den Regeln zustandsbezogene Lösungsmöglichkeiten zur Auflösung von ungültigen Zuständen abzuleiten. Die gespeicherten Regeln können automatisiert zur Anwendung kommen, nachdem ein Planer die Gültigkeit der Regel überprüft hat.

Erfahrungswissen kann über Methoden des Knowledge-Engineering⁵⁹, einer Methode aus der Expertensystemforschung, zu einem Regelsystem expliziert werden. Heidenreich beschreibt in seiner Arbeit das Konzept eines DSS für Produktionsnetzwerke der Serienfertigung. Es ermöglicht Experten, mithilfe einer Regelsprache, über Knowledge-Engineering apriori zustandsbasierte Regeln zur Auflösung definierter Ereignisse festzulegen. Grundlage der Regeln von Heidenreich ist die formale Regelsprache EBNF nach Backus und Naur.⁶⁰

Die Regeln ermöglichen nach ihrer Formalisierung die Auswahl von Änderungsplanungsverfahren für spezifische Entscheidungssituationen. Die Regeln sind ergo die Umsetzung der oben skizzierten Planungsstrategien für Produktionsnetzwerke. Die Regeln von Heidenreich besitzen einen Bedingungsteil und eine damit assoziierte Planungsaufgabe, welche die Variante eines anzuwendenden Planungsverfahrens bestimmt. Die Regeln werden mit den Konstrukten aus Tabelle 2.4 formalisiert.⁶¹

Durch *Priorität* wird festgelegt, welche Regel für ein Ereignis prioritär verwendet werden soll, sofern verschiedene Regeln für einen Objektknoten definiert sind. Der

⁵⁸Eine Architektur wurde von [HRB⁺99] vorgestellt. [Sch99a] stellt ein System speziell für die wissensbasierte Materialflusssteuerung vor.

⁵⁹Siehe z. B. eine Einführung in [GRS03]. Für weiterführende Informationen zum Thema Wissensakquisition und Formalisierung von Expertensystemen sei z. B. auf [KL90] verwiesen. Vorgehensmodelle hierzu finden sich z. B. in [PSS03], [KL90]. Siehe auch Übersichten zu einzelnen Verfahren wie D3 [PGPB96], CommonKADS [SAA⁺99] und der Protegé-II Ansatz [EST⁺95].

⁶⁰Details zu EBNF z. B. in [Sch97], [SW01] und [Wal03]

⁶¹[Hei06], Kap. 5., S. 139 ff.

Tabelle 2.4.: Aufbau der Steuerungsregeln nach Heidenreich

Bereich	Regelstruktur
Priorität	ID <priorität>
Bedingung	WENN <zustand> NACH <ereignistyp e> UND OPTIONAL <initiator>
Reaktion	DANN PLANE MIT <planungsverfahren p> OPTIONAL <variante v> UNTER BEACHTUNG <parameter p>

Zustand beschreibt den aktuellen Zustand eines Objektknotens im Modell der Fertigung.⁶² Der *Ereignistyp* beschreibt, ob das Ereignis durch eine Senkung oder Erhöhung eines Bedarfs oder Angebots ausgelöst wurde. Dieses ist wichtig, da Heidenreich in seiner Analyse gezeigt hat, dass nur bestimmte Planungsverfahren in bestimmten Zuständen sinnvoll anwendbar sind.⁶³ Der *Initiator* liefert den spezifizierten Bezug des Ereignistyps zu einem Kunden oder Lieferanten oder abstrakt: zu einem vor- oder nachgelagerten Objektknoten. Das *Planungsverfahren* referenziert ein anzuwendendes Änderungsplanungsverfahren, z. B. einem Verfahren aus der Menge der von Heidenreich aufgezeigten Algorithmen zur Änderungsplanung⁶⁴. Die *Variante* bestimmt die Art des verwendeten elementaren Planungsverfahrens und dessen Ausweitung. Die *Parameter* geben zusätzlich Informationen über zulässige Grenzen für Änderungen und Anfragen an. Jede neue Variante eines anwendbaren Änderungsplanungsverfahrens in einem ungültigen Zustand bedeutet potenziell eine neue anwendbare Regel zur Auflösung dieses Zustandes.

2.2.2. Formalisierung der Regeln - lokale und globale Entscheidungen

Regeln zur Steuerung der Änderungsplanung sind mit der in Tabelle 2.4 skizzierten Grammatik formalisierbar. Die Extraktion der Regeln aus dem Wissen eines Experten kann sich jedoch als schwierig erweisen. Ein Problem bei der manuellen Erstellung von Regeln zu Steuerung der Änderungsplanung in Produktionsnetzwerken ist der große Zustandsraum von Produktionsnetzwerken. Für die vielen möglichen Ereignisse und die daraus resultierenden Zustände in einem Produktionsnetzwerk müssen entsprechend viele Regeln formalisiert werden, um eine effiziente Steuerung der Änderungsplanung durch das Regelsystem zu ermöglichen. Bei der Formalisierung von

⁶²Siehe Kap. 2.1.1, S. 11

⁶³Siehe [Hei06], S. 54. Übersicht für Verfahren am FOK, und S. 73 für Verfahren am KOK

⁶⁴Ebd., S. 110. Übersicht S. 201 ff.

2.2. Anforderungen einer automatisierten Steuerung der Änderungsplanung durch Regeln

Regeln durch Knowledge-Engineering werden nicht alle Regeln erfasst, sodass umfassende Maßnahmen zur Absicherung der Integrität des Regelsystems notwendig werden.⁶⁵ Aufgrund des hohen Aufwandes der manuellen Regelsystemerstellung ist diese Art der Regelsystemerstellung als aufwendiger Prozess einzustufen. Insbesondere Regeln, die Entscheidungen bei unvollständiger Informationslage abdecken, sind schwer zu erfassen. Regeln nicht klar abgrenzbarer Zustände sind für einen Planer schwerer aussagekräftig zur formalisieren als Regeln für vollständig bekannte Zustände im Produktionsnetzwerk.

2.2.2.1. Regeln für deterministische Änderungsplanungsprozesse

Mit *lokalen* Regeln werden für die Änderungsplanung lokal anwendbare Änderungsplanungsverfahren ausgewählt. Diese Planungsverfahren ändern den Plan eines einzelnen Objektknotens, sind auf diesen beschränkt. Zur Formalisierung lokaler Regeln sind nur lokale, objektknotenbezogene (Planungs-)Informationen notwendig. Um z. B. eine Regel zur Aktion „Senkung des Sicherheitsbestandes in bestimmten Planungsperioden“ für einen Fertigungsobjektknoten zu denotieren, sind die Informationen über Min/Max-Restriktionen für Bestände eines Objektknotens erforderlich.

Da bei der lokalen Anwendung eines Änderungsplanungsverfahrens alle zur Planung notwendigen Informationen bekannt sind, wird die lokale Änderungsplanung für identische ungültige Ausgangszustände nach Anwendung eines identischen Planungsverfahrens gleiche resultierende Zustände erzeugen. Die Berechnung eines Planes in der Änderungsplanung unterliegt eindeutiger Berechnungsvorschriften, definiert durch die Planungsalgorithmen der Änderungsplanungsverfahren. Die lokale Änderungsplanung kann als *deterministischer* Prozess klassifiziert werden. Ein Regelsystem mit lokalen Regeln kann gut formalisiert und überprüft werden, da das Resultat der Regelanwendung des Regelsystems bezogen auf einen bestimmten ungültigen Zustand stets gleich ist. Die Komplexität der Regelformulierung wird gesenkt, da je relevantem Zustand und angewendetem Planungsverfahren eine Regel formuliert werden kann. Verifizierte lokale Regeln erreichen in der Anwendung eine hohe Zuverlässigkeit.

Bei der Formalisierung *globaler* Regeln ist zusätzlich zu den lokal verfügbaren Informationen die Wechselwirkung der beteiligten Objektknoten an einer kooperativen Änderungsplanung zu berücksichtigen. Die Art und Weise der Interaktion wird in der Änderungsplanung durch geänderte Bedarfs- oder Angebotsanfragen zwischen Kunden oder Lieferanten bestimmt. Die Interaktion folgt bestimmten Protokollen, welche die möglichen Handlungsalternativen bei der Durchführung der Änderungsplanung einschränken. Heidenreich definiert als Handlungsalternativen auf eine globale Anfrage die Protokolle „Bestätigung“, „Ablehnung“ und „geänderte Bestätigung“.⁶⁶ Die

⁶⁵Vgl. Kap. 2.2.1, S. 23.

⁶⁶Siehe [Hei06] und vgl. Kap. 2.1.3

2. Problemstellung

Anzahl möglicher Reaktionen auf eine Anfrage und das Ergebnis einer globalen Koordination sind durch diese drei möglichen Varianten begrenzt. Wäre dieses nicht der Fall, so wäre es für den Planer schwerer, wiederkehrende Situationen bei der Interaktion zwischen Kunden und Lieferanten zu erkennen und entsprechende Regeln oder Reaktionsmuster für entsprechende ungültige Zustände zu memorieren. Darüber hinaus schränken Leistungsvereinbarungen den Handlungsspielraum in der Anwendung globaler Verfahren zusätzlich ein.

2.2.2.2. Regeln für nicht-deterministische Änderungsplanungsprozesse

Dem Initiator sind bei einer Anfrage in einer globalen Änderungsplanung nicht alle Informationen zur sicheren Einschätzung der Reaktion des Partizipanten bekannt. Für identische Zustände beim Initiator können sich unterschiedliche Reaktionen des Partizipanten ergeben, abhängig von dessen Plan. Das Ergebnis der Änderungsplanung ist nicht vorhersehbar und kann für diesen Fall als *nicht-deterministischer* Prozess klassifiziert werden. Daraus folgt die Fragestellung, ob bzw. unter welchen Voraussetzungen für *nicht-deterministische Änderungsplanungsprozesse* Regeln formuliert werden können. Um diese Fragestellung zu beantworten, ist es notwendig, die spezifischen Besonderheiten eines Produktionsnetzwerkes und die Beziehungen und Interaktionen der Partner näher zu untersuchen.

Ein Partner, oder im Modell der Fertigung ein Objektknoten, kann z. B. als Kunde oder Lieferant aufgefasst werden. Die durchgeführten Planänderungen im Rahmen der Änderungsplanung definieren in diesem Kontext das Verhalten der Objektknoten. Für jeden einzelnen Objektknoten eines Produktionsnetzwerkes gelten festgelegte Restriktionen⁶⁷, wie z. B. Grenzen für Sicherheitsbestände oder maximal oder minimal bereitgestellte Betriebsmittelkapazitäten, innerhalb derer gültige Pläne des Objektknotens definiert sind. Diese werden als *charakteristische Merkmale* der Zustände in Produktionsnetzwerken bezeichnet.

Charakteristische Zustände

Je nach Objektknoten sind diese Grenzen anders ausgestaltet, und deren Pläne schwanken innerhalb, aber auch außerhalb dieser Grenzen.⁶⁸ Auftretende Schwankungen können verschiedene Charakteristika aufweisen. Z. B. wird bei einem Unternehmen mit starker Arbeitnehmervertretung⁶⁹ der Schichtplan aufgrund personaltechnischer Rahmenbindungen⁷⁰ für einen längeren Zeitraum fixiert sein und die bereitgestellten Be-

⁶⁷Vgl. Kap. 2.1.2

⁶⁸Diese Grenzen eines Produktionsnetzwerkes werden in AC/DC z. B. im Frame-Planning festgelegt ([DDKT07]). Die lokalen Pläne schwanken, je nach verfolgter Planungsstrategie, wie Bestellpunkt oder Bestellzyklus ([Sch05]).

⁶⁹Wie in der deutschen Automobilindustrie vorzufinden

⁷⁰Z. B. durch einen Tarifvertrag

2.2. Anforderungen einer automatisierten Steuerung der Änderungsplanung durch Regeln

triebsmittel und deren Kapazität werden wenig schwanken. Ein Lieferant hat z. B. flexible Schichtmodelle, um stark schwankende Kapazitätsauslastungen und Bedarfsspitzen kapazitär abzufangen, oder er versucht Bedarfsschwankungen seiner Kunden zusätzlich durch die Verwendung von Sicherheitsbeständen zu begegnen. Durch das Zusammenspiel der Objektknoten eines Produktionsnetzwerkes unter charakteristischen Merkmalen ergeben sich *charakteristische Zustände* eines Produktionsnetzwerkes.⁷¹ Charakteristische Zustände ermöglichen die Analyse charakteristisch auftretender Probleme bei der Koordination des Materialflusses im Produktionsnetzwerk. Diese Probleme drücken sich in *charakteristischen ungültigen Zuständen* aus.

2.2.3. Verfahren zur automatisierten Regelsystemerstellung

Zur Formalisierung der Regeln zur Steuerung der kooperativen Änderungsplanung ergeben sich nach dem Stand der Analyse folgende Möglichkeiten:

- Interviews mit Planern unter Verwendung von Knowledge-Engineering-Techniken
- Berechnungen im Rahmen einer mathematischen Analyse

Beide Varianten sind als manuelle Prozesse aufwendig zu realisieren. Die durch Knowledge-Engineering formalisierten Regeln weisen Schwächen auf.⁷² Aus diesen Gründen wird die Verwendung von Knowledge-Engineering zur Regelerstellung nicht weiter untersucht.

Im Weiteren werden mathematische Analysemethoden zur Regelerstellung betrachtet. Es wird diskutiert, inwiefern der bisher als aufwendig deklarierte Prozess der Regelformalisierung durch die Verwendung mathematischer Berechnungen algorithmisch formalisiert und automatisiert werden kann. Hierzu eignen sich z. B. Verfahren wie Heuristiken und Simulationsverfahren aus dem Operations Research (OR)⁷³, aber auch maschinelle Lernverfahren.⁷⁴

2.2.3.1. Ziel der automatisierten Regelerstellung

Ziel der Nutzung mathematischer Verfahren ist es, manuelle Prozesse, wie hier z. B. die Regelformalisierung, zu automatisieren. Die Automatisierung der Prozesse soll auf der Grundlage eines mathematischen Modells unter Vermeidung manueller Eingriffe

⁷¹Im Projekt AC/DC wird dieser Aspekt im Rahmen der Ereignisverarbeitung untersucht [DDLT07].

⁷²Vgl. Kap. 2.2.2, S. 24 ff.

⁷³Siehe [SM06]

⁷⁴Siehe eine Übersicht z. B. in [Mit97]

2. Problemstellung

durchgeführt werden können. Die automatisierte Regelsystemerstellung für die Steuerung der Änderungsplanung in Produktionsnetzwerken wird durch Nutzung des Modells des Produktionsnetzwerkes und dessen charakteristischer Zustände möglich.

Der Planer wird durch die Automatisierung des Regelerstellungsprozesses maßgeblich entlastet. Die erzeugten Regeln dienen den Planern eines Unternehmens zur Entscheidungsunterstützung bei der effektiven Auflösung ungültiger Zustände durch Auswahl eines zielführenden Änderungsplanungsverfahrens. Der Workflow in der Änderungsplanung wird durch diese zielorientierte Planungsunterstützung verbessert, Planungsentscheidungen werden für Dritte transparent und Steuerungsprozesse können kontinuierlich verbessert werden. Eine Gegenüberstellung des Aufwandes zur Regelerstellung durch Nutzung manueller oder automatisierter Methoden wird in Tabelle 2.5 vorgenommen.

Tabelle 2.5.: Gegenüberstellung durchzuführender Vorbereitungsmaßnahmen zur Regelerstellung mit und ohne automatisierte Methoden

Bereich	Manuell	Automatisiert
Modell	Muss formuliert werden	Muss formuliert werden
Zustände	Müssen formuliert werden, da Regeln auf Zuständen basieren	Müssen nicht formuliert werden, da sie berechnet werden
Regeln	Erfassung der bekannten Regeln (Erfahrung)	Erfassung der relevanten Regeln (automatisierte Enumeration)

2.2.3.2. Verwendbare Verfahren zur automatisierten Regelerstellung

Um die Anforderungen an die Problemstellung präzise formulieren zu können, werden im folgenden Kapitel verschiedene Verfahren analysiert. Es wird diskutiert, ob die vorgestellten Verfahren zur automatisierten Erzeugung der Regeln geeignet sind und ein geeignetes Verfahren ausgewählt.

Methoden des Operations Research

Zur automatisierten Lösung des adressierten Problems können diverse Verfahren des Operations Research verwendet werden.⁷⁵ Neben verschiedenen deterministischen Verfahren zur Berechnung von Produktionsplänen in der Änderungsplanung können nicht deterministische Probleme durch stochastische Methoden modelliert und Lösungen für solche Probleme berechnet werden. Die Berücksichtigung charakteristischer Zustände kann durch ein stochastisches mathematisches Modell erfolgen.

⁷⁵Siehe z. B. [SM06], [Tem06]

Monte-Carlo-Simulation

Eine populäre Methode zur Untersuchung stochastischer Prozesse ist die *Monte-Carlo-Simulation* und äquivalent klassifizierbare Methoden.⁷⁶ Die stochastischen Einflüsse auf die Objektknoten eines Produktionsnetzwerkes, wie Bedarfs- oder Angebotsänderungen, werden als stochastische Input/Output-Verteilung modelliert. „Nach dem Gesetz der großen Zahlen nähert sich die experimentelle Verteilung der Ausgangsdaten der theoretischen Verteilung, die der gegebenen Inputverteilung unterliegt, an.“⁷⁷

Der Fokus bei der Anwendung von Monte-Carlo-Methoden liegt auf Systemen, bei denen Entscheidungen mit Hilfe von Zufallszahlen durchgeführt werden müssen, wie z. B. Entscheidungen an einem Roulettetisch.⁷⁸ Die in dieser Arbeit zu lernenden Entscheidungen sind nicht ausschließlich zufallsbasiert, sondern werden durch die charakteristischen Merkmale der Zustände des Produktionsnetzwerkes bestimmt.

Reinforcement-Learning-Verfahren

Die Monte-Carlo-Simulation ist Teil der Klasse der *Reinforcement-Learning-Verfahren* (RLV).⁷⁹ Der Fokus von RLV liegt auf dem Lernen von schwer formalisierbaren Zusammenhängen oder Regeln in zufallsbehafteten Problemdomänen. Dieses ist hier der Fall, da formalisierte Modelle für das dargestellte Problem als NP-vollständiges Problem klassifiziert werden können.⁸⁰ Die stochastischen Einflüsse der Interaktion der Objektknoten sind schwer formal zu denotieren.

Aus diesem Grund werden zur weiteren zielorientierten Diskussion als möglicher Lösungsweg für die Problemstellung Reinforcement-Learning-Verfahren detaillierter betrachtet. Sie eignen sich zum automatisierten Erzeugen von Regeln für die Problemstellung dieser Arbeit.

2.2.4. Maschinelles Lernen der Regeln

Maschinelle Lernverfahren wurden bereits erfolgreich zur Umsetzung von Steuerungssystemen in technischen Systemen angewendet.⁸¹ Sie finden Anwendung, wenn das Problem durch ein mathematisches Modell nicht adäquat formalisiert werden kann oder der Aufwand dafür sehr hoch ist. Formal wird ein allgemeines *maschinelles Lernproblem* definiert als:

⁷⁶[SM06]

⁷⁷Ebd., S. 13

⁷⁸Ebd., S. 13

⁷⁹Vgl. [SB98], Kap. 5

⁸⁰Siehe [SM06] oder in Kap. 2.2, S. 21

⁸¹Siehe z. B. in [CA96], [HW98] oder [SB97]

2. Problemstellung

Definition 2.6 (Maschinelles Lernen) „A computer program is said to *learn* from Experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .“⁸²

Formal kann jedes definierte Lernproblem durch drei Merkmale beschrieben werden:

- Aufgabe T
- Leistungsmaß P
- Trainingserfahrung E ⁸³

Als Leistungsmaß des Lernprozesses kann die Effizienz⁸⁴ des maschinellen Lernsystems gemessen werden. Dieses ist Grundvoraussetzung, um automatisiert und objektiv feststellen zu können, ob ein maschinelles Lernsystem etwas „gelernt“ hat oder nicht.

2.2.4.1. Maschinelle Lernverfahren zur Produktionssteuerung

Holthöfer, der sich in seiner Dissertation mit der Steuerung von Produktionssystemen durch Regeln beschäftigt, schlägt vor, dass „statt einer expliziten Zuordnung [der Regeln] durch den Entscheider, das Entscheidungsverhalten durch ein lernendes Verfahren implizit zu erfassen.“⁸⁵ Dieser Gedanke Holthöfers wird in der Definition des Lernproblems dieser Arbeit aufgegriffen.

Definition 2.7 (Lernproblem) *Durch mathematische Bewertungen der Ergebnisse durchgeführter Änderungsplanungen, ausgehend von ungültigen Zuständen durch eine algorithmisierte Bewertungsfunktion, soll maschinell gelernt werden, welches Änderungsplanungsverfahren für einen ungültigen Zustand wahrscheinlich das beste Planungsergebnis bei einer Änderungsplanung erzielen wird. Dieses Bewertungsverfahren wird als Lernverfahren bezeichnet. Die erlernte Verknüpfung zwischen einem ungültigen Zustand und einem zu dessen Auflösung geeigneten Änderungsplanungsverfahren wird als Regel bezeichnet. Diese Regel wird zur Steuerung der Änderungsplanung für den Untersuchungsgegenstand verwendet. Da einem ungültigen Zustand je zugelassenem Planungsverfahren eine Regel zugeordnet werden kann, werden diese durch das Lernverfahren zustandsabhängig priorisiert. Es ist ein geeigneter Trainingsprozess für das Lernverfahren festzulegen, welcher durch wohldefinierte Lernepisoden*

⁸²[Mit97], S.2

⁸³Die Abkürzungen stehen für die englischen Begriffe *Task*, *Performance* und *Experience* [Mit97].

⁸⁴Engl. *performance*

⁸⁵Vgl. [Hol00], S.108. Holthöfer verwendet ein symbolisches Lernverfahren. In dieser Arbeit wird ein subsymbolisches Lernverfahren verwendet.

2.2. Anforderungen einer automatisierten Steuerung der Änderungsplanung durch Regeln

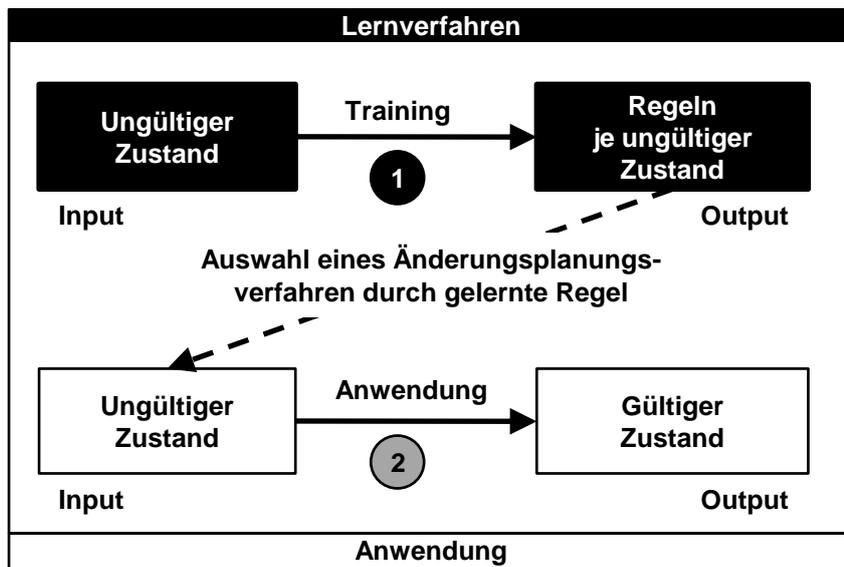


Abbildung 2.5.: Input-Output des Lernverfahrens und der Regelanwendung. In (1) werden die Regeln über das Training des Lernverfahrens unter Verwendung ungültiger Zustände erlernt. In (2) können die gelernten Regeln zur Auswahl eines Änderungsplanungsverfahrens in der Änderungsplanung angewendet werden.

strukturiert wird. Die Anzahl der betrachteten Zustände muss so reduziert werden, dass das Training effizient durchgeführt werden kann.

In der *Anwendung* können gelernte Regeln nach Abschluss des Lernprozesses zur Auswahl eines Änderungsplanungsverfahrens zur Auflösung eines ungültigen Zustandes verwendet werden, wie Abbildung 2.5 zeigt. Das Lernverfahren lernt a priori die anwendbaren Regeln. Als Eingabe für das Lernverfahren dient ein ungültiger Zustand, wie ihn auch ein Planer in der Planungsarbeit vorfindet. Ein ungültiger Zustand resultiert aus einem Ereignis mit nachfolgend durchgeführter Planbestandsrechnung.⁸⁶ Zur Auflösung des ungültigen Zustands wird im Lernverfahren ein Änderungsplanungsverfahren angewendet und das Resultat, ein neuer ungültiger Zustand oder ein gültiger Zustand, bewertet. Die wiederholte Anwendung von Änderungsplanungsverfahren auf verschiedene ungültige Zustände und die Bewertung des Planungsergebnisses realisiert das *Training* des Lernverfahrens.

Die Bewertung eines durchgeführten Änderungsplanungsprozesses soll ähnlich dem Entscheidungsverhalten eines Planers durchgeführt werden, um so anwendungsbezogene Regeln für ein Produktionsnetzwerk maschinell lernen zu können. Die maschinell durchgeführte Bewertung eines Änderungsplanungsprozesses orientiert sich dabei an der Entscheidungsaufgabe eines Planers, ob ein ungültiger Zustand durch ein lokales

⁸⁶Siehe Kap. 2.1.2, S. 17 f.

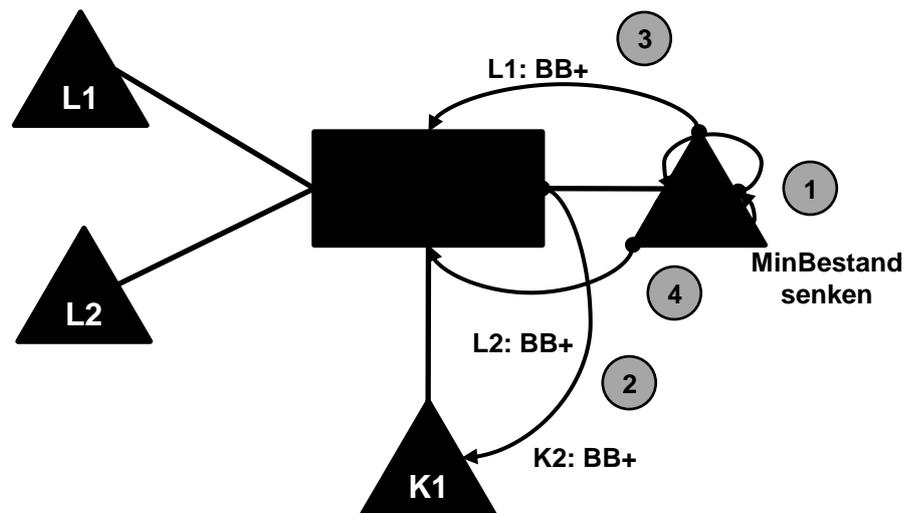


Abbildung 2.6.: Beispiel für das Lernproblem

oder globales Änderungsplanungsverfahren aufgelöst wird⁸⁷, wie Abbildung 2.6 beispielhaft zeigt. Die gelernten Regeln sollen festlegen, ob eine lokale Planungsstrategie wie eine Mindestbestandsreduzierung zur Anwendung kommt (1), oder ob eine globale Koordination mit Lieferant L1 in (3) oder mit Lieferant L2 in (4) je nach Variante des globalen Planungsverfahrens vorzuziehen ist. Bei einer globalen Koordination ist vorab in K1 anzufragen, ob Kapazitäten entsprechend bereitstehen oder gesenkt werden können (2).

Wird die Regelerstellung mit einem maschinellen Lernverfahren automatisiert, kann eine umfassendere Anzahl von ungültigen Zuständen in deutlich kürzerer Zeit analysiert und bewertet werden, als wenn diese Analyse und Bewertung manuell durch den Planer erfolgt wäre.

2.2.4.2. Maschinelles Lernen in komplexen Umgebungen

Maschinelle Lernverfahren werden oftmals in Problemdomänen mit einer großen Anzahl von Zuständen eingesetzt.⁸⁸ Produktionsnetzwerke weisen eine sehr große Anzahl möglicher Zustände auf. Beispielsweise ergeben sich für den Plan eines FOK mit einem Planungshorizont von 10 Perioden, mit Bestandsrestriktionen zwischen 0 und 100 Einheiten je Periode, bereits 101^{10} Planverläufe. Um alle Regelvarianten lernen zu können, müssten theoretisch alle Planungsverläufe durch das Lernverfahren mit der Lernfunktion analysiert werden. Dieses ist bei heutiger Rechenleistung kaum und für ein vollständiges Produktionsnetzwerk gar nicht in akzeptabler Zeit durchführbar. Das

⁸⁷Siehe Kap. 2.2.1 und 2.2.2, S. 22 ff.

⁸⁸Hierzu sei z. B. auf [TR96] verwiesen.

2.2. Anforderungen einer automatisierten Steuerung der Änderungsplanung durch Regeln

Problem wird verschärft, da Zustände im Lernprozess zur Rewardberechnung wiederholt bewertet werden müssen.

Der Einsatz maschineller Lernverfahren erfordert eine geschickte Modellierung des Problemraumes für die Konzeption des Lernverfahrens. Die Modellierung muss so durchgeführt werden, dass trotz quasi unendlich großer Problemräume in endlicher Zeit eine Lösung des Lernproblems möglich ist. Insbesondere das Lernen von globalen Regeln erfordert wegen der unvollständigen Informationslage zur Bewertung von Planungsläufen eine angepasste Modellierung des Problems, mit der effektiv Regeln gelernt werden können. Die geschickte Modellierung von Produktionsnetzwerken kann durch die Abstraktion diskreter Zustände erfolgen, wobei dem Abstraktionsprozess die charakteristische Merkmale⁸⁹ der Zustände des Produktionsnetzwerkes berücksichtigt. Die im Training benötigten ungültigen Zustände können über ihre spezifischen Merkmale auf charakteristische ungültige Zustände verdichtet werden. Der Zustandsraum wird so durch Abstraktion dieser Merkmale abstrahiert und so auf ein skalierbares Maß verkleinert.

2.2.4.3. Einsatz von Q-Learning

Maschinelle Lernverfahren aus der Klasse der Reinforcement-Learning-Verfahren⁹⁰ eignen sich besonders zur Lösung *unüberwachter*⁹¹ Lernprobleme. Dieses trifft auf die Problemstellung dieser Arbeit zu, da die Regeln automatisiert und ohne manuellen Eingriff durch das Lernverfahren erzeugt werden sollen.⁹² Reinforcement-Learning-Verfahren lassen sich auf alle Probleme anwenden, in denen ein autonomer Agent⁹³

⁸⁹Vgl. Kap. 2.2.2.2, S. 26 f.

⁹⁰Der Vollständigkeit halber seien auch die anderen wesentlichen unüberwachten Lernverfahren, Genetische Algorithmen ([Kle02], [Mig97], [Nis97], [Wei02]) und Künstliche Neuronale Netze ([Bra99] oder [Mit97]), erwähnt. Deren Ergebnisse sind schwer interpretierbar, wie verschiedene Anwendungen dieser Verfahren in der Literatur zeigen ([Bol03], [SRHGB05], [SRRF06]). Für die Anwendung von KNN zur Steuerung der Änderungsplanung in Produktionsnetzwerken, wie hier definiert, ist bisher keine Anwendung bekannt.

⁹¹Es gibt darüber hinaus Lernprobleme, die durch überwachte Lernverfahren gelöst werden können. Bekannte Vertreter überwachter Lernverfahren sind z. B. das Konzeptlernen (z. B. in [Sch99a]), das Entscheidungsbaumlernen (z. B. in [Mit97] oder [GRS03]), das allgemeine Regellernen (z. B. in [Qui90], [HSA99], [BKI03] oder [RN03]), das instanzbasierte Lernen (z. B. in [CH67], [AKM91], [Mit97] oder [GRS03]), das Bay'sches Lernen (z. B. in [PGPB96] oder [Mit97]) oder das analytische Lernen ([Mit97]). Alle Verfahren lernen eine bestimmte Fragestellung durch Analyse von Hypothesen. Im ersten Schritt müssen diese Hypothesen durch viele Trainingsbeispiele von einem Experten bewertet werden. Diese Trainingsbeispiele werden als Fakten in einer Wissensbasis abgelegt und ein algorithmischer Inferenzmechanismus kann aus dieser Faktenbasis etwaige neue Schlüsse auf unbekannte Probleme ziehen (z. B. in [Sch00]).

⁹²Die Anwendung der Regeln muss nicht automatisiert erfolgen. Der Lernprozess soll aber nachvollziehbar sein.

⁹³Eine Definition von Agent siehe z. B. in [Woo02], [RN03], [HR90] oder [FG96]. Als wichtigste Eigenschaften von Agenten werden: Autonomie, Pro-Aktivität, Reaktivität und soziales Verhalten genannt. Ein intelligenter Agent ist zusätzlich in der Lage, in einer nicht-deterministischen Umge-

2. Problemstellung

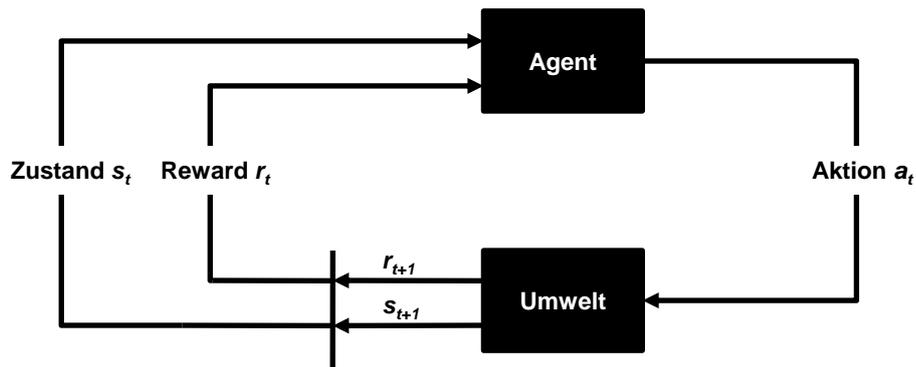


Abbildung 2.7.: Reinforcement-Learning-Systemarchitektur nach [SB98]

optimales Verhalten in einer definierten Umgebung lernen soll.

„Reinforcement Learning addresses the question of how an autonomous agent that senses and acts in its environment can learn to choose optimal actions to achieve its goals.“⁹⁴

Diese Definition korreliert mit der Interpretation der Definition des Lernproblems⁹⁵, da hier der Agent während des Lernprozesses die Entscheidungsaufgaben des Planers übernimmt und Rückschlüsse zu seinem Verhalten in seiner Umwelt gewinnen kann.

Im Modell des Reinforcement Learning ist ein Agent in einer Umwelt situiert, die er – teilweise⁹⁶ – wahrnehmen und durch eigene Aktionen beeinflussen kann. Jede Interaktion mit dieser Umwelt besteht darin, dass der Agent einen Zustand s aus dem Zustandsraum \mathcal{S} wahrnimmt und eine Aktion a aus der Menge der in diesem Zustand zulässigen Aktionen $A(s)$ auswählt. Die Aktion beeinflusst den Zustand der Umgebung, was einen Zustandsübergang von s nach s' bedeutet. Als Effekt dieser Aktion erhält der Agent einen Reward $r \in \mathbb{R}$, der als Bewertung des ausgelösten Zustandsüberganges interpretiert werden kann. Das Ziel des Agenten ist es, seine Aktionen derart zu wählen, dass er langfristig die Summe der Rewards maximiert, indem er die Auswahl der Aktionen stetig durch systematisches Probieren (*Trial-and-Error*) verbessert.

Die Strategie, nach welcher der Agent in einem Zustand s eine Aktion a wählt, wird allgemein als *Policy* bezeichnet und kann als Funktion $\pi : \mathcal{S} \mapsto \mathcal{A}$ aufgefasst werden. Sie ordnet jedem Zustand $s \in \mathcal{S}$ eine auszuführende Aktion $a \in \mathcal{A}$ zu. Die Bewertung von Zuständen mit Bezug auf eine bestimmte Policy wird allgemein als *Value-Funktion* bezeichnet. Ein Reinforcement Learning Problem ist gelöst, wenn die optimale Value-Funktion V^{π^*} bekannt ist. Eines der bekanntesten Verfahren, die optimale

bung zu agieren [Loc06].

⁹⁴[Mit97], S. 367

⁹⁵Vgl. Definition 2.7, S. 30 f.

⁹⁶Je nach Ausgestaltung des Lernsystems

2.2. Anforderungen einer automatisierten Steuerung der Änderungsplanung durch Regeln

Policy π^* bzw. die optimale Value-Funktion V^{π^*} zu lernen, ist das *Q-Learning*.⁹⁷ Es wird eine Q-Funktion festgelegt, die im Gegensatz zur Value-Funktion nicht nur auf Zuständen, sondern auf Zustands-/Aktionspaaren (s, a) definiert ist. Diese beschreibt den erwarteten Reward aus einem Zustand s , wenn als nächste Aktion a gewählt wird und anschließend weiter nach der optimalen Policy π^* verfahren wird. Die weiter verfolgte optimale Policy ist der maximale Q-Wert des Nachfolgezustandes s_{t+1} .

Es ist sehr aufwendig, alle Q-Werte rekursiv durch den Problemraum zu berechnen. Q-Learning approximiert iterativ mit einem gelenkten Trial-and-Error-Verfahren die jeweiligen Q-Werte der Q-Funktion der einzelnen Zustände durch \hat{Q} . Die Näherungen \hat{Q} werden je Zustand/Aktionspaar $\hat{Q}(s_t, a_i)$ in einer Tabelle gespeichert und nach einem Lernschritt, Q-Update genannt, aktualisiert. Dieses geschieht wie in Formel (2.1) dargestellt.

$$\hat{Q}(s_t, a_i) \leftarrow \hat{Q}(s_t, a_i) + \alpha \left[r(s_t, a_i) + \gamma \max_a \hat{Q}(s_{t+1}, a_j) - \hat{Q}(s_t, a_i) \right] \quad (2.1)$$

Das Q-Update aktualisiert die Bewertung eines Zustands-/Aktionspaares über die aktuelle Bewertung, den erzielten Reward und die Qualität des Folgezustandes. Wie man auf der rechten Seite der Formel (2.1) sehen kann, wird die Bewertung des Folgezustandes, wie oben erwähnt, über den dort maximal zu erzielenden Q-Wert approximiert.⁹⁸ Der approximierte Q-Wert $\hat{Q}(s, a)$ konvergiert nach theoretisch unendlich vielen Q-Updates gegen die Summe der Rewards, die zu erwarten ist, wenn in Zustand s Aktion a ausgeführt und anschließend weiter algorithmisch nach der optimalen Policy verfahren wird. Der Lernfaktor α gewichtet die jeweilige Stärke des Einflusses des Rewards bei der Verrechnung mit dem Q-Wert. Algorithmus 2.1 zeigt den vollständigen Ablauf des allgemeinen Q-Learning-Algorithmus.⁹⁹

Algorithmus 2.1 : Q-Learning

Initialisiere $\hat{Q}(s, a)$ beliebig für alle $s \in \mathcal{S}$ und $a \in A(s)$

Bestimme Zustand s

while true do

 Wähle $a \in A(s)$

 Führe Aktion a aus, beobachte Reward $r(s, a)$ und Folgezustand s'

$\hat{Q}(s, a) \leftarrow \hat{Q}(s, a) + \alpha [r(s, a) + \gamma \max_{a'} \hat{Q}(s', a') - \hat{Q}(s, a)]$

$s \leftarrow s'$

end

Nachdem das Q-Learning konvergiert, liegen mit der approximierten Q-Funktion Bewertungen für alle Paare von Zuständen und darin die zulässigen Aktionen vor. Diese

⁹⁷Siehe z. B. [Wat89] oder [WD92]

⁹⁸Dieses Vorgehen wird in der Literatur als *Bootstrapping* bezeichnet.

⁹⁹Vgl. [Mit97], Kap. 13.3

2. Problemstellung

Bewertungen können als Maß dafür angesehen werden, wie gut sich eine Aktion in einem bestimmten Zustand eignet, gemessen an den dafür zu erwartenden Rewards. Mit diesen Werten lässt sich eine Steuerungsstrategie definieren, die in jedem Zustand diejenige Aktion wählt, die die höchste Bewertung, also den höchsten Q-Wert, aufweist. Diese Strategie maximiert den langfristig zu erzielenden Reward.

Voraussetzung für ein effizientes Q-Learning

Die Effizienz von Q-Learning wird durch die problemorientierte Definition der Rewardfunktion und der Konvergenz der Q-Werte bestimmt. Q-Learning konvergiert zur Lösung einer Lernaufgabe unter der Voraussetzung, dass es sich bei der Lernaufgabe um einen endlichen Markov-Decision-Process handelt, alle Zustände vielfach besucht wurden und dass der Lernfaktor α hinreichend klein gewählt wurde.¹⁰⁰

Ein *Markov-Decision-Process* (MDP) ist dadurch charakterisiert, dass jede Entscheidung in einem Zustand *nicht* von einem vergangenen Zustand oder in der Vergangenheit ausgeführten Aktionen abhängt.¹⁰¹ Das Lernproblem dieser Arbeit ist durch einen MDP darstellbar. Jede Auswahl eines lokalen oder globalen Änderungsplanungsverfahrens verwendet den aktuellen Produktionsplan eines Objektknotens. Vergangene Pläne müssen bei dieser Entscheidung nicht explizit berücksichtigt werden. Der aktuelle Produktionsplan beinhaltet das Ergebnis aller vorher durchgeführten Planungsaufgaben.

2.2.5. Übertragung der Q-Learning-Konzepte auf die Problemstellung

Q-Learning ist geeignet, um im Sinne der Problemstellung dieser Arbeit die erforderlichen Regeln zur Steuerung der Änderungsplanung zu lernen.¹⁰² Jeder Objektknoten des modellierten Produktionsnetzwerkes wird durch einen lernenden Agenten repräsentiert. Die Höhe der Q-Werte¹⁰³ bestimmt nach Abschluss des Lernprozesses die Priorität eines anwendbaren Änderungsplanungsverfahrens zur Auflösung eines produktionsnetzwerkspezifischen ungültigen Zustandes.

Um Q-Learning anwenden zu können, muss definiert werden

- wie ein *Ausgangszustand* und *Endzustand* im Lernverfahren modelliert wird,
- welche *Aktionen* dem Lernverfahren zur Verfügung stehen,

¹⁰⁰Vgl. [Mit97], S. 372

¹⁰¹Siehe [KLM96]

¹⁰²Vgl. Definition 2.7, S. 30

¹⁰³Je größer der Q-Wert, desto besser ist das zu erwartende Planungsergebnis nach Anwendung der Regel.

2.2. Anforderungen einer automatisierten Steuerung der Änderungsplanung durch Regeln

- wie der *Reward* beim Q-Learning problemspezifisch berechnet wird und
- wie das *Training* durchzuführen ist und die notwendigen *Ausgangsdaten* bereitgestellt werden können.

2.2.5.1. Zustand

Der Zustand eines Agenten im Produktionsnetzwerk wird nach Definition 2.3¹⁰⁴ durch die Ausprägung seines Planes bestimmt. Im Kontext des Q-Learnings bleibt ein Agent so lange in einem Zustand, bis das Ereignis „Anfrage“ einen Zustandsübergang des Agenten durch eine Planänderung herbeiführt. Dieser Übergang wird durch die Planbestandsrechnung umgesetzt. Der Zustand eines Agenten wird durch die Merkmale aus Definition 2.4¹⁰⁵ beschrieben.

Die *Ausgangszustände* für das Q-Learning sind ungültige Zustände eines Agenten.¹⁰⁶ Hierfür werden die Regeln zur Steuerung der Änderungsplanung gelernt.¹⁰⁷ Ein *Endzustand* im Q-Learning ist ein gültiger Zustand eines Agenten. Der Plan des Agenten weist keine Restriktionsverletzungen mehr auf. Alle im Training auftretenden Zustände, die nicht Ausgangs- oder Endzustand sind, werden als *Folgezustand* bezeichnet.

2.2.5.2. Aktionen

Änderungsplanungsverfahren werden als Aktionen verstanden, die im Lernprozess des Q-Learnings zur Verfügung stehen. In dieser Arbeit werden beispielhaft die Änderungsplanungsverfahren von Heidenreich verwendet. Es sind beliebige weitere Aktionen zur Anwendung im Lernverfahren denkbar, sofern sie sich nach der Logik der Planungsstrategien für die Änderungsplanung anwenden lassen. Durch die Auswahl von Aktionen im Lernprozess werden in dieser Arbeit mögliche Planungsstrategien umgesetzt.

2.2.5.3. Reward und Rewardfunktion

Zur Bewertung der Qualität der Zustände in Produktionsnetzwerken können Kostenfunktion verwendet werden. Für die Rewardberechnung ist es jedoch nicht notwendig reale Kosten eines Produktionsplanes zu bewerten. Denn es sollen Strafkosten ermittelt

¹⁰⁴Siehe S. 14

¹⁰⁵Siehe S. 16

¹⁰⁶Das Abstraktionsverfahren operiert auf diskreten Ausgangszuständen repräsentiert durch ungültige Planzustände des Produktionsnetzwerkes. Diese Zustände werden zu charakteristischen ungültigen Zuständen als Ausgangszustände der Ausgangsdaten für das Lernverfahren durch das Abstraktionsverfahren abstrahiert. Siehe Kap. 2.2.2.2, S. 26 und Kap. 2.2.4.2, S. 33

¹⁰⁷Siehe Kap. 2.2, S. 21 ff.

2. Problemstellung

werden, die eine Bewertung des Lernfortschrittes im Sinne des Lernproblems, nämlich die Verbesserung eines Planes durch eine Änderungsplanung, ermöglichen. Bei der Bewertung der Leistung eines Lernschrittes über Strafkosten durch die Rewardfunktion sollen die in Kapitel 2.1.2 skizzierten Kosten zugrunde gelegt werden.

Betrachtet man die in der Änderungsplanung berücksichtigten Parameter¹⁰⁸, so können die *Strafkosten* eines Planes bestimmt werden durch:

- die Bewertung von Restriktionsverletzungen
- Strafkosten für Bereitstellungsprozesse zur lokalen Materialbereitstellung
- Strafkosten für Materialbeschaffungsprozesse
- Strafkosten für die Bereitstellung von Betriebsmitteln als Produktionskapazitäten

Restriktionsverletzungen können durch Strafkosten bewertet werden, um grundsätzlich die Leistungsfähigkeit bei der Auflösung dieser durch eine Aktion messen zu können. Bereitstellungsstrafkosten und Betriebsmittelstrafkosten messen die Verbesserung von Plänen bezüglich quantitativer Veränderungen durch eine lokale Aktion, z. B. die Reduzierung des Sicherheitsbestandes oder die Erhöhung des Leistungsgrades. Beschaffungsstrafkosten bewerten globale Aktionen und messen den Erfolg von Beschaffungsprozessen, um z. B. in der Änderungsplanung durch gelernte Regeln zwischen Lieferanten auswählen zu können.

2.2.5.4. Zustand, Reward und Q-Update

Für das Q-Update im Q-Learning ist entscheidend, dass die Bewertungsfunktion nach erfolgter Änderungsplanung eine klare Differenzierung zwischen Verbesserung oder Verschlechterung eines Planes ermöglicht. Ist dieses der Fall, so spiegeln die Q-Werte nach Abschluss des Trainings den potenziell zu erwartenden Erfolg oder Misserfolg einer Aktion zur Auflösung eines ungültigen Zustandes wider und bestimmen so über die Höhe des Q-Wertes die Priorität eines anzuwendenden Änderungsplanungsverfahrens. Das Regelsystem wird über die mit Q-Werten priorisierten Aktionen erzeugt.

Die Rewardberechnung erfolgt durch die Berechnung der Differenz der Kosten des Planes eines Agenten vor und nach erfolgter Änderungsplanung. Zum zustandsabhängigen Kostenvergleich wird der *vollständige* Plan eines Agenten vor und nach erfolgter Aktion kostentechnisch bewertet. Dadurch wird sichergestellt, dass nicht nur die Restriktionsverletzungen in die Bewertung einbezogen werden, sondern auch planbezogene Kosten wie die Bestandshöhe nach erfolgter Änderungsplanung bewertet werden können. Dieses ist z. B. für Änderungsplanungsverfahren erforderlich, deren Varianten unterschiedliche Strategien bei der Erhöhung oder Verringerung von Losgrößen –

¹⁰⁸Vgl. Kap. 2.1 und 2.1.3, S. 9 ff.

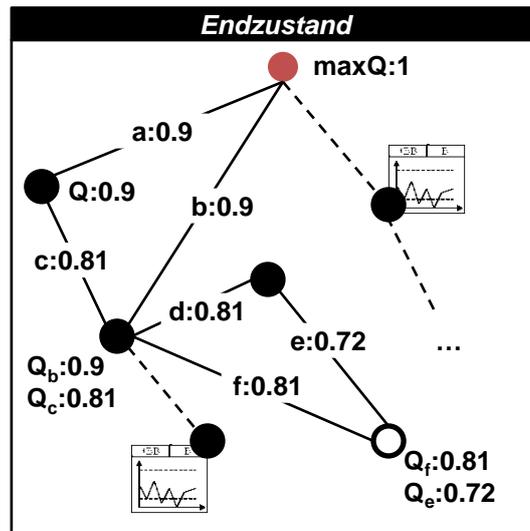


Abbildung 2.8.: Lernpfad des Q-Learnings

z. B. prozentual oder fix – verfolgen. Die isolierte Betrachtung der Restriktionsverletzungen bei der Bewertung von Zuständen bzw. Aktionen lässt diese Aspekte außer Acht und beschränkt die Bewertung von Zuständen und Aktionen folglich auf die Restriktionsverletzungen. Das Q-Update erfolgt über die Q-Formel¹⁰⁹ auf den Q-Werten des Ausgangszustandes und unter Berücksichtigung des maximalen Q-Wertes des Folgezustandes.

Beispiel

Es sei jeder Punkt in Abbildung 2.8 ein Zustand in einem Produktionsnetzwerk und die Pfade zwischen den Zuständen Aktionen, die einen ungültigen Plan in einen gültigen Plan, hier als Endzustand bezeichnet, überführen.¹¹⁰ Einerseits wird für den betrachteten ungültigen Plan als Ausgangszustand¹¹¹ deutlich, dass von hier über die Aktionspfade

$$\begin{aligned}
 1 &: e \rightarrow d \rightarrow b & (2.2) \\
 2 &: e \rightarrow d \rightarrow c \rightarrow a \\
 3 &: f \rightarrow b \\
 4 &: f \rightarrow c \rightarrow a
 \end{aligned}$$

der Endzustand am Kopf der Abbildung erreicht werden kann.

¹⁰⁹Vgl. Formel (2.1)

¹¹⁰Rückschritte sind in diesem Beispiel nicht möglich.

¹¹¹Repräsentiert durch den nicht ausgefüllten Kreis

2. Problemstellung

Zum Lernen der Pfade bzw. der Q-Werte zwischen Zuständen beginnt der Q-Learning-Algorithmus zunächst in einem zufälligen ungültigen Zustand und führt in diesem Zustand anfänglich zufällige Aktionen durch. Wird durch eine Aktion ein Endzustand erreicht, z. B. durch die Aktion b , so kann der erzielte Reward für die Aktion im Ausgangszustand verbucht werden. Dieser berechnet sich in Abhängigkeit vom maximalen Q-Wert des Folgezustandes und dem erzielten Reward der durchgeführten Aktion¹¹², diskontiert um den *Diskontfaktor* γ . Dieser hat hier den Wert 0,9. Für Aktion b errechnet sich in diesem Beispiel ein neuer Q-Wert von 0,9, da hier vereinfachend jeweils nur $0,9 \cdot \max(Q)$ als neuer Q-Wert angenommen wird.

Gleiches gilt für die Verzweigungen b und c . Für den Pfad c errechnet sich der bessere Q-Wert, da diese Aktion direkt in den Endzustand übergeht, wobei mit Pfad b über den Umweg von Pfad a der Endzustand erreicht werden kann. Aus dieser Überlegung lässt sich der optimale Pfad

$$3 : f \rightarrow b \quad (2.3)$$

bestimmen. Dieses ist der kürzeste Pfad. Praktisch werden in diesem Beispiel zwei Planungsaktionen benötigt, um in einen gültigen Plan einzuschwingen. Formel (2.3) kann als effiziente Regel für dieses Szenario verstanden werden.

Ziel des Q-Learnings ist es, viele Zustand-/Aktionspaare zu bewerten und so sukzessive ein Produktionsnetzwerk dieser Pfade aufzubauen, die in der Anwendung die Regeln zur Steuerung der Änderungsplanung repräsentieren.¹¹³

2.2.5.5. Ausgangsdaten für den automatisierten Lernprozess

Sowohl der Prozess zur Abstraktion des Zustandsraumes, als auch der Q-Learning-Prozess benötigen definierte Ausgangsdaten in Form von ungültigen Plänen. Sollen Abstraktions- und Lernverfahren automatisiert durchgeführt werden können, so müssen die Ausgangsdaten maschinell zu verarbeiten sein. Um die Qualität der gelernten Regeln für ein spezifisches Produktionsnetzwerk zu gewährleisten, sollten sie als Realdaten, z. B. aus ERP-Systemen, vorliegen. Ausgangsdaten bestimmen die Bewertung der Zustände während des Trainings, bewertete Zustände bestimmen über die Qualität der Q-Werte, und diese über die Anwendbarkeit und Effizienz der gelernten Regeln.

Um die Ausgangsdaten aus ERP-Systemen extrahieren zu können, muss eine Schnittstelle zwischen dem Lernsystem und dem ERP-System definiert werden.¹¹⁴ In dieser Schnittstelle muss festgelegt werden, welche standardmäßigen Daten aus ERP-Systemen im Lernverfahren verwendet werden. Die Beschreibung der Verwendungsmöglichkeiten der Ausgangsdaten bzw. deren Verfügbarkeit liefert den Rahmen für

¹¹²Vgl. Formel (2.1)

¹¹³Vgl. Kap. 2.2.5

¹¹⁴Z. B. aus ERP-Systemen wie SAP (<http://www.sap.com>).

den Trainingsprozess des Verfahrens. Dieser muss so ausgestaltet werden, dass mit den Ausgangsdaten aus ERP-Systemen ein Regelsystem gelernt werden kann.

2.3. Zusammenfassung der Problembereiche

Q-Learning wird verwendet, um die Regeln zur Steuerung der Änderungsplanung in Produktionsnetzwerken zu lernen.¹¹⁵ Der Q-Wert je Zustand/Aktionspaar repräsentiert das Attribut `<priorität>`¹¹⁶ der Regeln eines Zustandes, wobei die Anzahl der Aktionen die Anzahl möglicher Regeln für einen Zustand bestimmt.¹¹⁷ Zur Umsetzung des Lernverfahrens sind die in den nächsten Kapitel zusammengefassten Forschungsfragen zu adressieren.

2.3.1. Effektive Abstraktion des Zustandsraumes

Ein wesentliches Problem fast aller Anwendungsbereiche für das Q-Learning-Verfahren ist der große Zustandsraum des Untersuchungsgegenstandes. Dieses trifft auf Produktionsnetzwerke der Serienfertigung zu.¹¹⁸ Die große Anzahl der Zustände im Produktionsnetzwerk verursacht eine lange Rechenzeit des Lernverfahrens.

Effektive Abstraktion bedeutet hier, dass durch die Nutzung charakteristischer Merkmale der Zustände des Produktionsnetzwerkes bei der Abstraktion die Nachvollziehbarkeit der Lernergebnisse für den Planer ermöglicht wird. Der Bezug zwischen den Ursprungszuständen und deren Abstraktion soll bestehen bleiben in dem der Zustandsraum des Produktionsnetzwerkes auf charakteristische Zustände verdichtet wird.¹¹⁹ Es ist ein effektives und effizientes Abstraktionsverfahren zur Verkleinerung des Zustandsraumes zu entwickeln. Effektiv bedeutet hier, dass

- die charakteristischen Merkmale der Zustände des Produktionsnetzwerkes erhalten bleiben,
- die abstrahierten ungültigen Zustände als Ausgangszustände nach der Zustandsreduktion weiterhin betriebswirtschaftlich interpretierbar sind, wie die Zustände des ursprünglichen Zustandsraumes,
- die Zustandsreduktion so skalierbar ist, dass je nach Komplexität des Produktionsnetzwerkes eine Anpassung des Zustandsraumes für das Q-Learning möglich ist und

¹¹⁵Vgl. Kap. 2.2

¹¹⁶Vgl. Tab. 2.4, S. 24

¹¹⁷Vgl. Tab. 2.4

¹¹⁸Siehe Kap. 2.2.4, S. 32

¹¹⁹Vgl. 2.2.3.2, S. 28

2. Problemstellung

- die Abbildung zwischen den ursprünglichen Zuständen $x \in X$ und den Zuständen des reduzierten Zustandsraumes $y \in Y$ surjektiv ist: $f(X) = X \rightarrow Y, \forall y \in Y \exists x \in X : f(x) = y$.

Effizienz bedeutet, dass die Laufzeit des Abstraktionsverfahrens so effektiv ist, dass sich trotz durchgeführtem Abstraktionsprozess Laufzeitvorteile für den Lernprozess auf dem abstrahierten Zustandsraum ergeben.

Dieser Problembereich wird mit folgender Forschungsfrage zusammengefasst:

Forschungsfrage A:

Es muss eine effektive Abstraktionsfunktion für ein Abstraktionsverfahren entwickelt werden, welches den Zustandsraum effektiv reduziert und so das Lernverfahren für diesen in endlicher Zeit konvergiert.

2.3.2. Effektive Lernfunktion

Eine streng problemorientierte Definition der Lernfunktion ist der wesentliche Erfolgsfaktor bei der Konzeption eines effektiven Q-Learning-Verfahrens. Formel (2.1)¹²⁰ unterstreicht dieses. Bei jedem Q-Update wird als Lernfunktion die Rewardfunktion zur Rewardberechnung verwendet. Sie beeinflusst so die Q-Werte und das Ergebnis des gesamten Lernprozesses.

Eine problemspezifische Bewertung der Änderungsplanungsprozesse zur Auflösung ungültiger Zustände kann durch die Rewardfunktion des Lernverfahrens pragmatisch mit den Strafkostenarten

- Strafkosten von Restriktionsverletzungen
- Bereitstellungsstrafkosten
- Beschaffungsstrafkosten
- Betriebsmittelstrafkosten

durchgeführt werden. Um Verzögerungen im Lernprozess zu verhindern, muss die Rewardfunktion so ausgestaltet sein, dass ein Q-Update in linearer Zeit durchgeführt werden kann.

Eine effektive Rewardfunktion bewertet die Zustände so, wie ein Planer diese bewerten würde, wobei z. B. Pläne mit Restriktionsverletzungen höhere Strafkosten verursachen können als Pläne ohne Restriktionsverletzungen. Die Höhe des Rewards bewertet die Verbesserung oder Verschlechterung eines Planes nach erfolgter Änderungsplanung.

¹²⁰Vgl. Kap. 2.1, S. 35

Zur vollständigen Bewertung der Strafkosten einer Aktion müssen alle erforderlichen Zustandsinformationen wie Restriktionsgrenzen und Planverlauf eines Objektknotens vorliegen. Für lokale Änderungsplanungen ist dies kein Problem, da zu der Bewertung für jeden Objektknoten alle lokalen Informationen vorliegen.¹²¹ Zur vollständigen Bewertung globaler Aktionen müsste zur Ermittlung der Gesamtlage der Plan des Partizipanten an der Änderungsplanung mit in die lokale Strafkostenbewertung einbezogen werden. Da der Initiator und der Partizipant in einem Produktionsnetzwerk eine eher kompetitive denn kooperative Zusammenarbeit pflegen, ist dieses nur eingeschränkt möglich.¹²² Die Strafkostenfunktion für die Rewardberechnung müssen für globale Aktionen mit unvollständigen Informationen adäquate Ergebnisse im Lernprozess erzielen können.¹²³

Bei einer globalen Koordination werden anzupassende Kapazitäten für alle Erzeugnisse der Fertigungsstufe gleich behandelt, wobei der verfügbare Leistungsgrad eines KOK unabhängig von der Produktionsreihenfolge angefragt werden kann. Die erforderliche Anpassung der Kapazität bei globaler Koordination muss bei der Konzeption der Lernfunktion berücksichtigt werden.

Effizientes Lernen von Steuerungsregeln in Produktionsnetzwerken bedingt apriori einen durch Abstraktion reduzierten, handhabbaren Zustandsraum.¹²⁴ Die Rewardfunktion und das Q-Update müssen so gestaltet werden, dass trotz Abstraktion eine effektive Bewertung von Zuständen durchgeführt und auf diesen effizient gelernt werden kann.

Dieser Problembereich wird mit folgender Forschungsfrage zusammengefasst:

Forschungsfrage B:

Zur Rewardberechnung sind Strafkostenfunktionen zu entwickeln, die eine effektive Bewertung von Planungsaktionen im Rahmen einer Rewardfunktion ermöglichen, sodass für das Lernverfahren ein Lernfortschritt über die Q-Updates herbeigeführt werden kann. Die Rewardfunktion soll trotz unvollständiger Informationslage und abstrahiertem Zustandsraum ein effizientes Q-Learning von Regeln zur Steuerung der Änderungsplanung ermöglichen.

¹²¹Vgl. Kap. 2.2.2.1, S. 25

¹²²Diese Zusammenarbeit wird insbesondere in der Automobilindustrie in Zukunft potenziell ausgebaut. Siehe die FAST2015-Studie von Mercer Management Consultants und der Fraunhofer Gesellschaft [FM04] bzw. die Bestrebungen des EU-Projektes AC/DC z. B. in [DDKT07].

¹²³Vgl. Kap. 2.2.2.2, S. 26

¹²⁴Vgl. Kap. 2.3.1

2.3.3. Automatisierte Regelgenerierung durch effizientes Training

Um den Lernprozess effizient zu gestalten, muss ein automatisiertes Verfahren zur Abstraktion des Zustandsraumes konzipiert werden. Ein automatisierter Prozess ist ein unüberwachter Input/Output-Prozess, der Ausgangsdaten mit einem festgelegten Algorithmus und unter Berücksichtigung definierter Zielgrößen sowie Restriktionen in endlicher Zeit in Ergebnisdaten transformiert. Er wird hier als Training bezeichnet. Der Ablauf des Trainings ist über Parameter konfigurierbar. Als Ausgangsdaten für das Training werden ungültige Pläne des Produktionsnetzwerkes verwendet. Das Training ist effizient, wenn bezogen auf eine manuelle Erstellung des Regelsystems mit gegebenen Ausgangsdaten in adäquater Zeit anwendbare Regeln zur Steuerung der Änderungsplanung erzeugt werden können.

Sowohl für einen effektiven Trainingsprozess des Zustandsabstraktionsprozesses als auch für den Lernprozess müssen adäquate Ausgangsdaten bereitgestellt werden. Diese sind Realdaten, z. B. aus einem ERP-System. Es ist zu analysieren, welche Daten hierzu vorhanden sein müssen und wie eine Schnittstelle zwischen Lernsystem und ERP-System konzipiert sein sollte.

Die Überwachung des Trainingsprozesses und der Fortschritte des Lernprozesses erfordert, sowohl für das Abstraktionsverfahren als auch für das Lernverfahren *Lernepisoden* zu definieren. Ein Trainingsprozess besteht aus einer endlichen Anzahl davon. In den Lernepisoden des Abstraktionsverfahrens wird für jeden ungültigen Zustand der Ausgangsdaten je ein Abstraktionsschritt durchgeführt. In den Lernepisoden des Lernverfahrens wird die Rewardbewertung auf den abstrahierten Zuständen ausgeführt.

Nach Abschluss des Trainings wird das Regelsystem durch einen Algorithmus erzeugt, der in dieser Arbeit spezifiziert werden muss. Der Algorithmus verwendet zur Erzeugung der Regeln die gelernten Q-Werte aus dem Trainingsprozess des Lernverfahrens. Umgekehrt muss ein Algorithmus spezifiziert werden, der in der Anwendung bei Eingabe eines ungültigen Zustandes eine Regel zur Steuerung der Änderungsplanung auswählt.¹²⁵

Dieser Problembereich wird mit folgender Forschungsfrage zusammengefasst:

Forschungsfrage C:

Es muss ein effizienter Trainingsprozess sowohl für das Abstraktionsverfahren als auch für das Q-Learning-Verfahren konzipiert werden. Der Ablauf der Lernepisoden ist zu spezifizieren. Es sind die benötigten Ausgangsdaten, deren Struktur und Quelle zu bestimmen. Zur Quelle muss ei-

¹²⁵Vgl. Abb. 2.5, S. 31

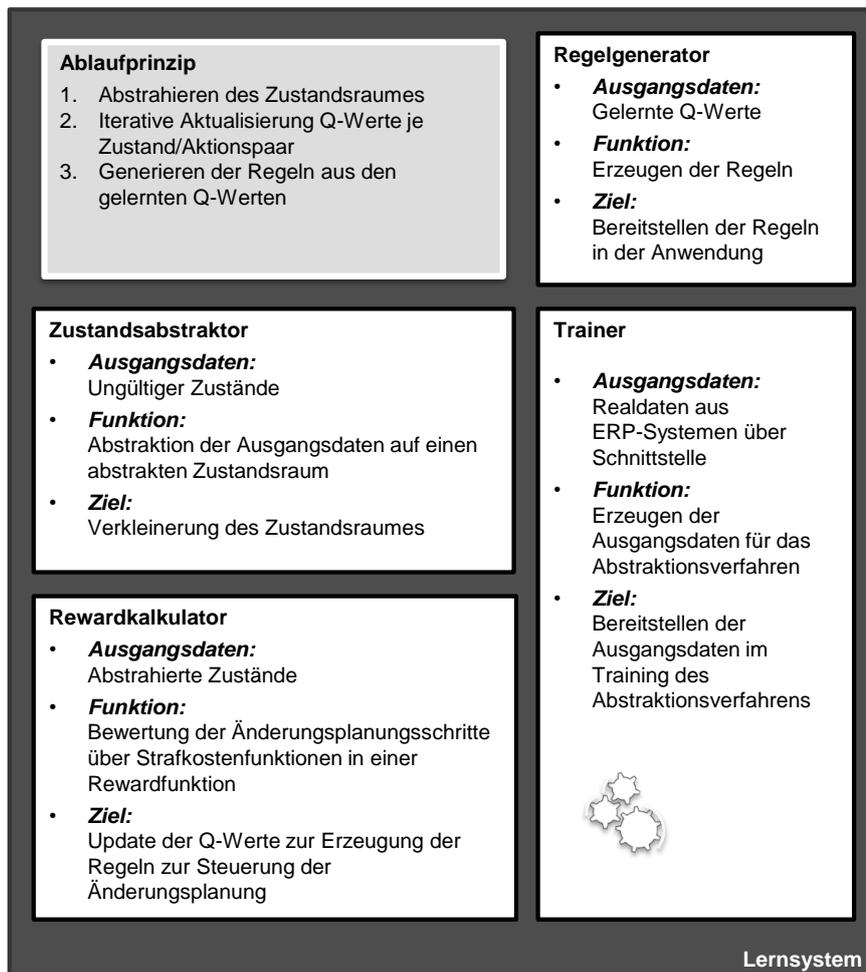


Abbildung 2.9.: Die Architektur des Lernsystems wird in Teilmodule dekomponiert und greift dabei die Struktur der Forschungsfragen A-C auf.

ne Schnittstelle definiert werden. Der Ablauf und das Ende des Trainings sollen mit Hilfe von Parametern steuerbar sein.

2.3.4. Systemarchitektur und Zusammenfassung

Das automatisierte Lernen von Regeln zur Steuerung der Änderungsplanung in Produktionsnetzwerken stellt Anforderungen an verschiedene Problembereiche aus der Betriebswirtschaftslehre und der künstlichen Intelligenz, die durch ein Softwareprogramm umgesetzt werden sollen. Die Architektur dieses Programms besteht aus Teilmodulen, die jeden Problembereich dieser Arbeit abdecken. Abbildung 2.9 zeigt eine grafische Darstellung der Architektur und fasst stichwortartig die einzelnen Funktionen der Teilmodule zusammen.

2. Problemstellung

Einerseits wird Q-Learning in der Regel durch die Verwendung von situierten Agenten realisiert. Andererseits werden die Objekte in Produktionsnetzwerken in dieser Arbeit als dezentral organisierte und über Nachrichten verknüpfte Objektknoten modelliert. Aus diesem Grund bietet sich zur Umsetzung des Lernsystems ein Multiagentensystem (MAS) an. Dafür ist zwischen zweckgebundenem MAS, wie z. B. MASCOPP von Heidenreich¹²⁶ und MAS-Programmbibliotheken wie JADE¹²⁷ zu unterscheiden.

Erste setzen bereits konkrete Probleme in einem nutzbaren MAS um, während Zweite durch die Bereitstellung von Objekten, Methoden und Protokollen in sogenannten APIs¹²⁸ eine theoretisch beliebige anwendungsbezogene Implementation eines MAS ermöglichen. Es bleibt im Weiteren zu analysieren, ob hier zur Umsetzung des MAS eine API verwendet werden soll, oder ob ein bereits bestehendes MAS verwendet werden kann, in welches das Lernsystem integriert werden kann.

Für die Umsetzung des Zustandsabstraktors, des Rewardkalkulators, des Trainers und des Regelgenerators ist zu überprüfen, ob bereits geeignete Konzepte in der Literatur existieren, die eventuell in der Konzeption und Umsetzung des Lernsystems angewendet werden können. Dieses wird im folgenden Kapitel untersucht.

¹²⁶Ebd.

¹²⁷[Til]

¹²⁸Engl. *Application Programming Interface*

3. Stand der Forschung

Wenige wissen, wie viel man
wissen muss, um zu wissen,
wie wenig man weiß.

(William Faulkner)

Nachdem im vorherigen Kapitel die Forschungsfragen dieser Arbeit identifiziert, problematisiert und diskutiert wurden, müssen diese in den aktuellen Stand der Forschung eingeordnet werden. Kapitel 3.1 konzentriert die Analyse auf geeignete Verfahren zur Zustandsabstraktion, die sowohl für den Untersuchungsgegenstand, als auch in Kombination mit dem Q-Learning-Verfahren zielführend einsetzbar sind. Kapitel 3.2 betrachtet existierende Q-Learning-Verfahren in der Produktionsplanung und -steuerung und diskutiert deren Eignung zur Lösung der Problemstellung. In Kapitel 3.3 wird diskutiert, ob das Lernsystem in bestehende Multiagentensysteme integriert werden kann oder eine eigenständige Multiagentensystemlösung umgesetzt werden muss. Ebenso werden Methoden zur Modellierung der Schnittstelle für die Bereitstellung der notwendigen Ausgangsdaten für das Lernsystem betrachtet.

3.1. Zustandsreduktionsverfahren für Produktionsnetzwerke der Serienfertigung

Um das adressierte Problem des Q-Learnings in großen Zustandsräumen zu lösen, werden in der Literatur verschiedene Verfahren, wie z. B. Neuronale Netze, Regression Trees zur Approximation der Q-Funktion oder Clusterverfahren zur Reduzierung der Zustände des Problembereiches, beschrieben.¹ Die Verfahren lassen sich in drei Klassen unterteilen:

1. Verfahren zur Approximation der Value Funktion V
2. Verfahren zur Reduktion des Zustandsraumes durch Abstraktion mittels eines Abstraktionsverfahrens
3. Verfahren, die relationale Zustandsbeschreibungen zur Generalisierung von Wissen nutzen

¹Siehe z. B. [SB98]

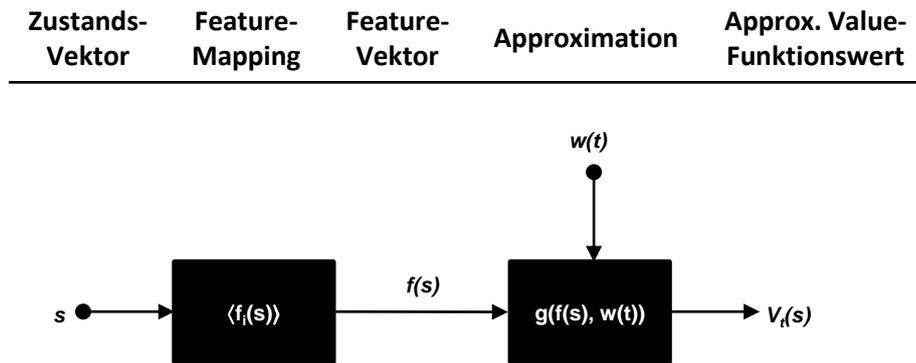


Abbildung 3.1.: Framework zur Approximation der Value-Funktion [SB98]

In den folgenden Kapiteln erfolgt eine Diskussion über die Eignung der klassifizierten Abstraktionsverfahren zur Reduktion des Zustandsraumes von Produktionsnetzwerken. Ziel ist die Auswahl eines Verfahrens zur Zustandsreduktion, das für die Problemstellung der Arbeit geeignet ist. Hierauf fußt dann die folgende Diskussion über existierende Ansätze in der Literatur.

3.1.1. Approximation der Value-Funktion

Beim Q-Learning wird für jeden bewerteten Zustand eine Value-Funktion gelernt, die den erwarteten Reward beziffert, den der lernende Agent aus diesem Zustand heraus erreichen kann. Da dieser Wert im Falle von großen Zustandsräumen nicht für jeden Zustand explizit gelernt und gespeichert werden kann, basieren alle Verfahren dieser Klasse zur Lösung des Q-Learning-Problems auf der Idee, diese Werte lediglich zu approximieren.

Ziel bei der Approximation der Value-Funktion² ist es, Zustände $s \in \mathcal{S}$ eines Zustandsraumes auf einen *Feature-Vektor* $f(s) = (f_1(s), f_2(s), \dots, f_m(s))$ abzubilden. Jeder einzelne Feature f_i ist eine Abbildung aus dem Zustandsraum \mathcal{S} in einen Raum Q_i mit möglichen Ausprägungen dieses Features, die jeweils spezielle Eigenschaften eines Zustandes widerspiegeln. Z. B. können Features eines Zustandes mit vielen numerischen Ausprägungen, z. B. der Bestand eines FOK, auf eine kleine Menge diskreter Ausprägungen, wie z. B. Lagerbestand „hoch“, „mittel“, „niedrig“, reduziert werden.

²Siehe z. B. [TR96]

3.1.1.1. Feature-Vektoren

Der Feature-Vektor kann mit einem Parameter-Vektor kombiniert werden.³ Hiermit kann

$$V_t(s) = g(f(s), w(t)) \quad (3.1)$$

direkt approximiert werden, da der Wert von $V_t(s)$ direkt aus der Abbildung der Kombination von Features und Parametern zu einem analysierten Zeitpunkt durch den Agenten ermittelt werden kann. Wird ein Feature durch die Belegung seiner Parameter repräsentiert, so können, ohne die Value-Funktion explizit zu definieren, Rückschlüsse auf den Zustand und auf die Value-Funktion des Zustandes gezogen werden. In der Literatur beziehen sich die meisten Autoren mit „Approximation der Value-Funktion“ auf den zweiten Schritt dieses Frameworks, d. h. die Wahl der Funktionsapproximation g . Die Zustandsbeschreibung als Eingabe von g ist beliebig. Eine zuvor durchgeführte Reduktion auf einen kompakteren Feature-Vektor muss nicht berücksichtigt werden.

Die Approximation der Value-Funktion kann mithilfe von Feature-Vektoren als lineare Funktion durchgeführt werden.⁴ Kann die Value-Funktion nicht linear approximiert werden, so können Techniken Neuronaler Netze, wie z. B. der Backpropagation-Algorithmus⁵ oder das Multi-Layer-Perzeptron⁶ angewendet werden. Neben den bereits vorgestellten linearen und nichtlinearen Verfahren können Regressionsbäume zur Funktionsapproximation verwendet werden.

Die Zustände von Produktionsnetzwerken sind durch Planungsperioden mit diskreten Belegungen, Restriktionen sowie Leistungsvereinbarungen definiert. Die Einteilung dieser Zustände durch Features wie

Lagerbestand = (hoch, mittel, niedrig)

scheint auf den ersten Blick eine deutliche Reduktion der Zustände des Produktionsnetzwerks zu bewirken. Dieses Vorgehen wirft jedoch verschiedene Probleme auf. Bereits die Änderung des Planes einer einzelnen Periode um den Wert 1 kann diesen Plan in einen ungültigen Zustand überführen. Die grobe Partitionierung des Zustandsraumes auf die dargestellten Features kann diese Feinheiten der Planunterschiede nicht adäquat abbilden. Die charakteristischen Merkmale eines Zustandes würden in ihrem Abstraktum so nicht erhalten bleiben, was der Prämisse dieser Arbeit widerspricht.

Um eine deutliche Reduktion der Zustände erreichen zu können, müsste die Abstraktion des Zustandsraumes durch die Verwendung von Features recht grob sein. Selbst die oben beispielhaft aufgeführte Partitionierung in (*hoch, mittel, niedrig*) führt bereits

³Siehe Abb. 3.1

⁴Siehe [SB98] oder [Wat89]

⁵Siehe [RHW86]

⁶Vgl. [Tes95], S.60

bei einem hinreichend großen Plan von 30 Perioden zu einem Zustandsraum mit 3^{30} Zuständen. Eine Approximation der Value-Funktion durch Feature-Vektoren wäre für Produktionsnetzwerke und deren Zustände, wie sie hier definiert sind, so nicht sinnvoll anwendbar.

3.1.1.2. Regressionsbäume

Regressionsbäume⁷ sind Bäume⁸, an deren inneren Knoten Tests durchgeführt werden, um einen Eingabevektor von der Wurzel an den Baum „herunterzureichen“, bis er eindeutig einem der Blätter zugeordnet werden kann. An dem resultierenden Blatt des Baumes wird eine Berechnung auf dem Eingabevektor ausgeführt, die schließlich den gewünschten Funktionswert approximiert.⁹ Im Kontext des Reinforcement-Learning soll ein Regressionsbaum einen Zustand s über die Tests an den inneren Knoten bis zu einem Blatt klassifizieren und anschließend über eine Berechnungsvorschrift an diesem Blatt den Wert $V(s)$ approximieren.

Nach dem Ansatz von Wang und Dietterich¹⁰ können Regressionsbäume bereits als eine Partitionierung des Zustandsraumes angesehen werden, bei der jede Partition alle Zustände enthält, die auf den gleichen Knoten des Baumes abgebildet werden. Regressionsbäume setzen numerische, durch Feature-Vektoren repräsentierte Attribute voraus, da Abstände zwischen einzelnen Zuständen über das Skalarprodukt der Feature-Vektoren berechnet werden.

3.1.1.3. Relationale Zustandsbeschreibungen

Der Ansatz der relationalen Zustandsbeschreibung ist eine intuitive Art, eine Menge von Zuständen durch Beziehungen zwischen Objekten auszudrücken.¹¹ Für die Anwendung muss die Problemdomäne eine entsprechende Beschreibung der Zustände zulassen. Die hier betrachteten Zustände eines Produktionsnetzwerkes lassen sich nur sehr schwer in Beziehungen zueinander setzen, da sie nicht zwingend ursächlich voneinander abhängen.

3.1.1.4. Bewertung

Die oben vorgestellten Abstraktionsverfahren erhalten die charakteristischen Merkmale der Zustände des Produktionsnetzwerkes beim Abstraktionsprozess nicht. Die

⁷Vgl. [Mit97], Kap. 3

⁸Häufig sogar binäre Bäume

⁹Siehe [ZD95]

¹⁰[WD99]

¹¹Siehe z. B. [LR02] oder [Mor03]

Approximation der Value-Funktion bildet Zustände durch Funktionen ab, die mit Verschlüsselungsfunktionen verglichen werden können. Das Ergebnis der Abstraktion ist ein Funktionswert oder Vektor, dessen Denotation keinen direkten Bezug zum Ursprungszustand hat. Ähnliches gilt für relationale Zustandsbeschreibungen. Hier wird der abstrahierte Zustandsraum als Relation zwischen den Ursprungszuständen modelliert. Diese Verfahren können als *implizite Abstraktionsverfahren* bezeichnet werden. In der Literatur werden implizite Abstraktionsverfahren in Problemdomänen eingesetzt, in denen der Mensch das Ergebnis des Lernprozesses nicht explizit nachvollziehen muss. Hierbei handelt es sich um automatisierte, sich autonom steuernde Anwendungssysteme, wie der automatische Back-Gammon-Spieler *TD-Gammon* von Tesau-ro.¹² Ein weiteres Beispiel ist das automatische System zur Signalsteuerung in Multi-media-Netzwerken von Tong und Timothy¹³.

Auch in der Forschung von Produktionssystemen wird die Automatisierung von Steuerungssystemen durch den Einsatz maschineller Lernsysteme kombiniert mit Feature-Vektoren zur Zustandsabstraktion angestrebt. Zhang et. al. betrachten Job-Scheduling als Reinforcement-Learning-Problem und suchen eine Policy V^* , die in Ausfallsituationen mit wenigen Scheduling-Aktionen ein zulässiges Re-Scheduling durchführt.¹⁴ Die als RDF bezeichnete Value-Funktion wird durch eine spezielle Metrik umgesetzt und dient zur Bewertung einzelner Schedules. Die Approximation der Value-Funktion wird durch ein künstliches neuronales Netzwerk (KNN) mithilfe eines Feature-Vektors realisiert. Als Eingabe des KNN werden die vollständigen und eindeutigen Informationen eines Zustandes benötigt, welche für die Problemstellung dieser Arbeit *nicht* vorhanden sind.¹⁵ Weiterhin ist das von Zhang et. al.¹⁶ verwendete Modell des Produktionssystems inkompatibel mit dem Modell des Untersuchungsgegenstandes. Das Ergebnis des Lernprozesses von Zhang et. al.¹⁷, die Steuerungslogik, wird implizit durch ein KNN repräsentiert und stellt keine explizite Repräsentation von Regeln dar.¹⁸

3.1.2. Approximation durch Zustandsaggregation

Bei der Approximation des Zustandsraumes durch Aggregation wird, im Gegensatz zur direkten Approximation der Value-Funktion, versucht, den Zustandsraum durch Abbildung von ähnlichen Zuständen in einen aggregierten Zustand durch Abstraktion zu minimieren. Das Q-Learning Verfahren operiert in dem abstrahierten und so verkleinerten Zustandsraum.

¹²TD-Gammon kombiniert TD-Learning, eine Spezialisierung des Q-Learnings, mit neuronalen Netzen zur Approximation der Value-Funktion (Siehe [Tes95]).

¹³Engl. *Call Admission Control and Routing* (Siehe [TB04])

¹⁴Siehe [ZD95]

¹⁵Siehe Kap. 2.2.2

¹⁶Siehe [ZD95]

¹⁷Ebd.

¹⁸Vgl. Kap. 2.2.3.2, S. 28 und die Forschungsfragen A und B in den Kap. 2.3.1 und 2.3.2, S. 41 ff.

3.1.2.1. Entscheidungsbäume

Zur sukzessiven Abstraktion des Zustandsraumes werden Entscheidungsbäume verwendet, wobei der Entscheidungsbaum die Werte der Q-Funktion kompakt repräsentiert und je nach analysierten Zuständen angepasst wird. Beispiele sind der G-Learning-Algorithmus von Chapman und Kaebeling¹⁹ sowie der U-Tree-Algorithmus von McCallum und Katchis²⁰. Verschiedene weitere Kombinationen der Value-Funktionsapproximation und Zustandsabstraktionsverfahren wurden in der Literatur untersucht.²¹

3.1.2.2. Clusterverfahren

In der Literatur wird der Einsatz von Clusteringverfahren im Bereich des Q-Learnings beschrieben. Mahadevan et. al. verwenden z. B. statisches Clustering zum Q-Learning von Robotersteuerungen, die einen Roboter befähigen soll, Boxen in einer unbekanntem Umwelt zu verschieben.²² Als Ähnlichkeitsmaß²³ zur Abstraktion des Zustandsraumes wird die Umwelt durch einen 18-Bit-Vektor repräsentiert. Durch die formale Umweltrepräsentation als Vektor kann der Roboter einen direkten Bezug zwischen Umwelt und interner Repräsentation der gelernten Regeln herstellen. Wie in dieser Arbeit muss das Lernsystem von Mahadevan et. al.²⁴ Regeln ohne Kenntnis aller verfügbaren Informationen in einem großen Zustandsraum lernen. Durch das Clustering wird dieser auf den 18-bit Vektor reduziert und ermöglicht das Lernen der Steuerungsregeln in endlicher Zeit. Der Roboter kann in einem deterministischen, aber abstrahierten Zustandsraum operieren.

Eine weitere Möglichkeit ist die Verwendung des dynamischen Clustering, während des Lernprozesses, bei dem der abstrahierte Zustandsraum nach jedem analysierten Zustand angepasst wird.²⁵ Ein einfaches und oft eingesetztes Verfahren ist z. B. der *k-means*-Clustering-Algorithmus.²⁶ Eine Anwendung von Clustering zur Abstraktion des Zustandsraumes des Untersuchungsgegenstandes „Produktionsnetzwerk“ ist nicht bekannt.

¹⁹Siehe [CK91]

²⁰Siehe [McC95]

²¹Siehe z. B. [TB04], [UV98] oder [SJJ95]

²²Siehe [MC92]

²³Hier mithilfe der Hamming-Distanz. Die Hamming-Distanz zwischen zwei binären Vektoren entspricht der Anzahl der Vektor-Elemente, die unterschiedliche Werte aufweisen.

²⁴Ebd.

²⁵Ebd.

²⁶Siehe [HW79]. Es gibt auch diverse andere Verfahren, die in [JMF99] oder [Ber02] dargestellt und hier nicht verwendet werden.

3.1.2.3. Bewertung

Das Ziel, die charakteristischen Merkmale der Zustände von Produktionsnetzwerken trotz Abstraktion zu erhalten, wird durch das Konzept des Clusterings positiv unterstützt. Beim Clustering werden gerade bei der Konzeption der Ähnlichkeitsmetrik die charakteristischen Merkmale der Zustände zur Abstraktion verwendet.²⁷ Der abstrahierte Zustandsraum ist eine surjektive Abbildung des ursprünglichen Zustandsraumes.²⁸ Die Abbildung bildet die Ursprungszustände des Produktionsnetzwerkes auf charakteristische Zustände ab und reduziert so den Zustandsraum. Die Ausgestaltung dieser Abbildung hat direkten Einfluss auf das Ergebnis des Lernverfahrens. Werden Regeln auf charakteristischen Zuständen definiert, so sind diese Regeln für einen Planer genauso nachvollziehbar wie Regeln, die auf diskreten Zuständen definiert werden, da in den charakteristischen Zuständen die charakteristischen Merkmale der Ursprungszustände erhalten bleiben. Clusterverfahren können als *explizite Abstraktionsverfahren* bezeichnet werden.

Es können Analogien zwischen dem Untersuchungsgegenstand und den oben aufgeführten Forschungsarbeiten aufgezeigt werden, welche die Eignung von Clustering für die Problemstellung unterstreichen. Die Anwendung eines Clusteringverfahrens ist vergleichbar mit der Anwendung eines Planungsverfahrens. Das Ergebnis der Anwendung der Verfahren ist bei identischen Ausgangsdaten stets gleich. Übertragen auf die Welt des in Kapitel 3.1.2.2 erwähnten Roboters bedeutet die Unsicherheit einer globalen Planung in einem Produktionsnetzwerk für diesen das Ausprobieren von Aktionen in unbekanntem Umweltzuständen. In der Roboterwelt gelten Restriktionen die helfen die Unsicherheit zu minimieren, wie z. B. physikalische Gesetze. Die Gesetzmäßigkeiten lassen sich mit typischen Verhalten der Partner in Produktionsnetzwerken vergleichen.

Die Anwendung des Clusteringverfahrens von Mahadevan et. al.²⁹ ist für die Problemstellung dieser Arbeit nicht möglich, da sowohl die Clusterfunktion als auch die Ergebnisvektoren des Clusterings keinen Bezug zu den Anforderungen dieser Arbeit aufzeigen. Dieses gilt insbesondere für die betriebswirtschaftliche Interpretierbarkeit der Clusteringergebnisse. Das Problem der in der Literatur beschriebenen Clusteringverfahren ist die problemspezifische Umsetzung der Abstraktionsfunktion. Durch die jeweilige Art der Umsetzung ist die Wiederverwendung einer spezifizierten Abstraktionsfunktion in einem anderen Problemkontext als schwierig anzusehen.

Eine Kombination von Clustering mit Entscheidungsbäumen³⁰ kann sinnvoll sein, wenn der Zustandsraum durch harte Kriterien bereits vor dem Durchlaufen der Ähn-

²⁷Vgl. Kap. 3.1.3, S. 54

²⁸Vgl. Kap. 2.3.1, S. 41

²⁹Ebd.

³⁰Vgl. Kap. 3.1.2.1, S. 52

lichkeitsmetrik des Clusterings partitioniert werden kann. In dieser Arbeit kann z. B. das Zustandsmerkmal

Anfragender Objektknoten

als hartes Unterscheidungskriterium für relevante Zustände gesehen werden, da im Q-Learning partnerspezifische Regeln gelernt werden sollen.³¹

Als Fazit kann festgehalten werden, dass Clustering als geeignete Methode zur Lösung des Abstraktionsproblems für das Lernverfahren eingestuft werden kann. Die Ergebnisse des Clusterings sind explizit nachvollziehbar und die charakteristischen Eigenschaften des Untersuchungsgegenstandes können, je nach Modellierung der Abstraktionsfunktion, erhalten bleiben. Ein mögliches Clusterverfahren, das zur Anwendung in dieser Arbeit geeignet ist, ist das *k-means*-Clusterverfahren. Diese wird im nächsten Kapitel im Kontext der Problemstellung diskutiert.

3.1.3. Anwendung von *k-means*-Clustering

Der *k-means-Algorithmus*³² zählt zu den bekanntesten Clusterverfahren. Seine Verbreitung begründet sich dadurch, dass es ein einfaches und robustes Verfahren ist, das sich in zahlreichen Anwendungen bewährt hat.³³ Der Algorithmus partitioniert eine Menge der Trainingsdaten $X = \{x_1, \dots, x_n\}$ in eine vorgegebene Anzahl von k Clustern C_1, \dots, C_k . Dabei wird jedes Cluster C_i durch einen Mittelpunkt³⁴ c_i repräsentiert, der sich als gewichteter Mittelwert aus den Elementen des Clusters ergibt. Jedes Element der Trainingsdaten und sowie die Clustermittelpunkte lassen sich als Vektor im \mathbb{R}^d darstellen.

Elementar für die Funktionsweise des Algorithmus ist die Definition einer Distanzfunktion $d : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$, die als Maß für die Ähnlichkeit zweier Datenpunkte dient. Je kleiner das Distanzmaß für zwei Punkte x_i, x_j ist, desto ähnlicher sind sich diese Punkte und desto größer ist die Wahrscheinlichkeit, dass sie demselben Cluster zugewiesen werden. Eines der am häufigsten verwendeten Ähnlichkeitsmaße ist die Euklidische-Distanz, die als

$$d_{\text{Euklidisch}}(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

³¹Vgl. Kap. 2.2.4.1, S. 30

³²Siehe [HW79]

³³Vgl. [JMF99], S. 278

³⁴Engl. *centroid*

definiert ist. Der k -means-Algorithmus verfolgt das Ziel, in jedem Cluster die quadrierte mittlere Distanz der zugehörigen Elemente zum jeweiligen Mittelpunkt zu minimieren:

$$\min \sum_{C_i} \sum_{x_j \in C_i} d(x_j, c_i)^2 \quad (3.2)$$

Zu diesem Zweck weist der Algorithmus iterativ jeden Datenpunkt gerade dem Cluster zu, dessen Mittelpunkt die geringste Distanz zu ihm aufweist. Der genaue Ablauf ist in Algorithmus 3.1 wiedergegeben.

Algorithmus 3.1 : Der k -means-Algorithmus

Eingabe : Zu clusternde Daten

Ausgabe : Geclusterte Daten

- 1 Lege Anzahl der Cluster k fest
 - 2 Wähle initiale Clustermittelpunkte c_1, \dots, c_k aus der Menge der Trainingsdaten X
 - 3 Für jedes Element $x_i \in X$ bestimmt das Cluster C_j , sodass die Distanz $d(x_i, c_j)$ minimal ist
 - 4 Für jedes Cluster C_j ($j = 1, \dots, k$) aktualisiere den Mittelpunkt c_j der einzelnen Punkte, die C_j im vorherigen Schritt zugewiesen wurden
 - 5 Wiederhole die Schritte 3 und 4, bis ein Abbruchkriterium erfüllt ist
-

Die Anzahl der zu erzeugenden Cluster muss als Parameter k festgelegt werden. Anschließend werden die Clustermittelpunkte c_1, \dots, c_k initialisiert. Dies geschieht typischerweise dadurch, dass k Elemente der Trainingsmenge als initiale Clustermittelpunkte gewählt werden. Die Mittelpunkte müssen nicht notwendigerweise in der Trainingsmenge enthalten sein, sodass eine Initialisierung mit zufällig gewählten Einträgen möglich ist.

Nach dieser Initialisierungsphase beginnen die Iterationen des Algorithmus. In jeder Iteration wird jedes Element der Trainingsmenge einem bestehenden Cluster, repräsentiert durch seinen Mittelpunkt, zugewiesen. Dieses Cluster wird so gewählt, dass die Distanz des zuzuweisenden Punktes zum Clustermittelpunkt des besten Clusters minimal ist. Nach diesem Schritt ist die Menge der Trainingsdaten in k -Cluster partitioniert, da jedes Element genau einem Cluster zugewiesen wurde.

Der darauf folgende Aktualisierungsschritt berechnet die Clustermittelpunkte c_1, \dots, c_k mit dieser Partitionierung neu. Typischerweise geschieht dies, indem der Mittelpunkt c_j als Mittelwert der Elemente des Clusters C_j berechnet wird:

$$c_j = \sum_{x_i \in C_j} x_i / |C_j| \quad (3.3)$$

Anschließend beginnt die nächste Iteration des Algorithmus unter Berücksichtigung der aktualisierten Clustermittelpunkte erneut mit der Suche nach einem passenden Cluster für jedes Element der Trainingsmenge. Der Algorithmus terminiert, wenn eines

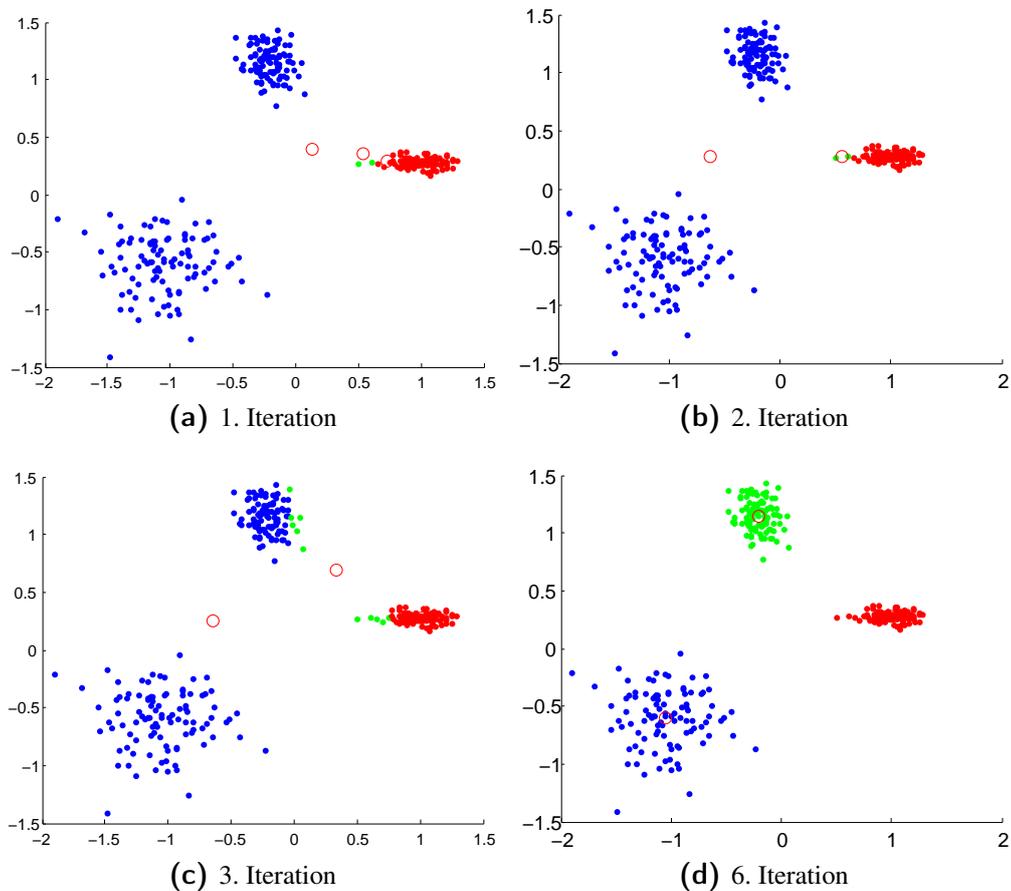


Abbildung 3.2.: Beispiel für die Anwendung des k -means-Algorithmus

von mehreren möglichen Abbruchkriterien eintritt. Einerseits kann der Algorithmus beendet werden, wenn sich die Clustermittelpunkte in einer Iteration nicht ändern. In diesem Fall sind im vorherigen Schritt alle Zuordnungen von Datenpunkten zum Cluster konstant geblieben und der Algorithmus ist konvergiert. Da diese Konvergenz nicht in jedem Fall garantiert werden kann, werden in der Regel weitere Abbruchkriterien eingeführt. So kann der Algorithmus z. B. terminieren, wenn die Summe der quadrierten Distanzen über mehrere Iterationen nicht signifikant abgenommen hat oder eine maximale Anzahl von Iterationen erreicht wurde.

Abbildung 3.2 zeigt ein Beispiel für das Clustern von Punkten im \mathbb{R}^2 mit dem k -means-Algorithmus. In Abbildung 3.2(a) erkennt man die initialen Clustermittelpunkte als offene Kreise. Die Zuweisung der Trainingsdatenpunkte nach der ersten Iteration des Algorithmus ist durch die farbliche Markierung ersichtlich. Abbildungen 3.2(b)-(d) zeigen die Clustermittelpunkte und Zuweisungen der Punkte zu den Clustern nach der 2., 3. und 6. Iteration. Nach der 6. Iteration terminiert der Algorithmus, da die Zuweisung der Punkte zu den Clustern konstant bleibt und sich folglich die Clustermittelpunkte nicht mehr ändern.

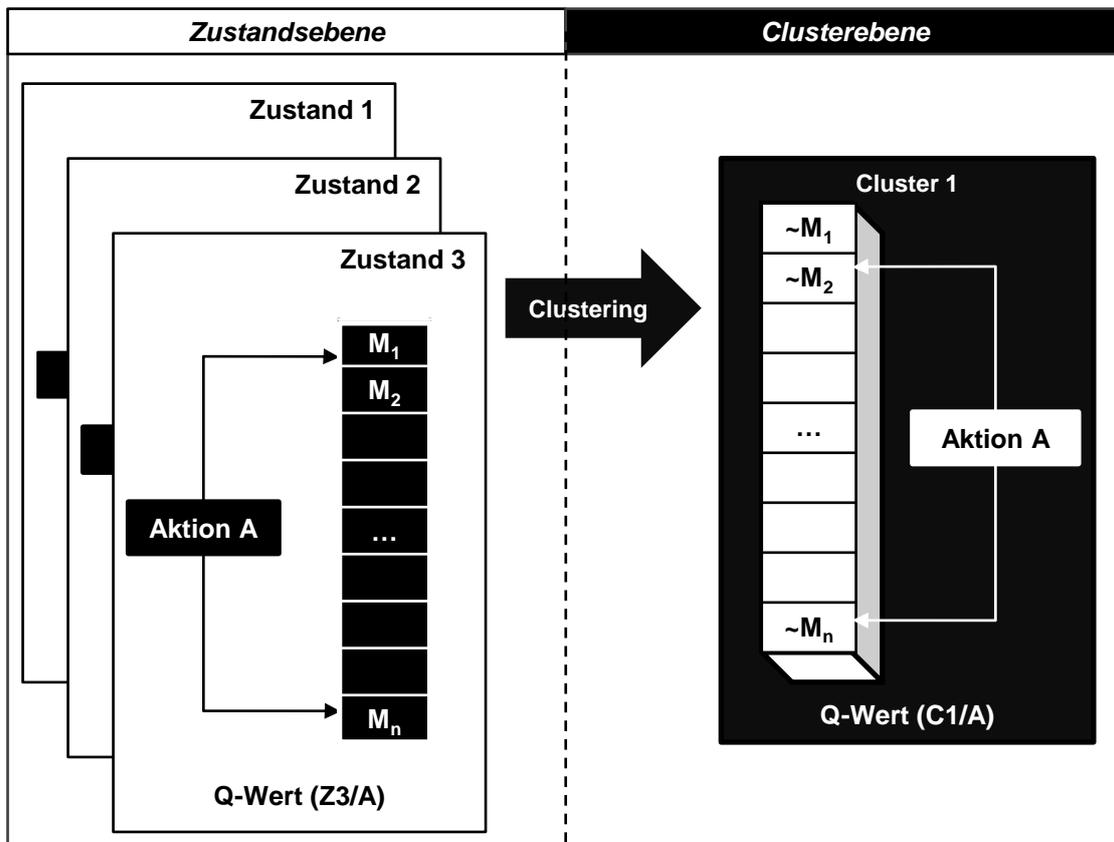


Abbildung 3.3.: Zusammenhang von Zustand, Cluster und Q-Werten – Aktion A operiert auf den Merkmalen M_i

Der Algorithmus des *k-means*-Clustering bietet inhärent wenige Möglichkeiten, vorhandenes Wissen über die zu abstrahierenden Daten bzw. die gewünschten Cluster einzubringen. Der Distanzfunktion zur Berechnung der Abstände zwischen Punkten und dem Centroid kommt so im Rahmen des Algorithmus eine besondere Bedeutung zu, da sie die Zuordnung von einzelnen Elementen zu Clustern maßgeblich beeinflusst. Es ist bei der Zuweisung im Clusteringprozess sinnvoll, als Maß zur Bewertung der Ähnlichkeit zweier Pläne Domänenwissen in die Abstraktionsfunktion des Clustering zu integrieren, um problemspezifische Cluster erzeugen zu können.

3.1.3.1. Bewertung

Das Clusteringverfahren soll für den Untersuchungsgegenstand durch die Verwendung einer Distanzfunktion ungültige Planzustände des Produktionsnetzwerkes in charakteristische ungültige Planzustände abstrahieren und so den Problemraum für das Lernverfahren verkleinern. Das Lernverfahren operiert lediglich auf dem abstrahierten Zustandsraum. Zur Abstraktion ungültiger Zustände werden deren Planverläufe über die

zu erarbeitende Distanzfunktion unter Berücksichtigung charakteristischer Merkmale beim Clusteringprozess Clustern zugewiesen, wie Abbildung 3.3 zeigt. Nach Abschluss des Clusterings repräsentiert jeder Centroid der erzeugten Cluster einen charakteristischen Planverlauf einer Menge ähnlicher diskreter ungültiger Zustände des Produktionsnetzwerkes.

In der Literatur werden zahlreiche Distanzfunktionen für das *k-means*-Clustering diskutiert.³⁵ Zu den Bekanntesten gehören z. B. die oben erwähnte Euklidische-Distanz, die Manhattan-Distanz oder die Hamming-Distanz für binäre Datenvektoren. Diese Metriken eignen sich nur bedingt für das vorliegende Problem, da die Elemente eines Clusters in dieser Arbeit Planverläufe repräsentieren. Für diese Planverläufe soll jeweils dieselbe optimal zu wählende Aktion für die Änderungsplanung durch das Lernverfahren bestimmt werden. Für die Auswahl einer Aktion im Training des Lernverfahrens und in der Anwendung der Regeln sind einige Merkmale von größerer Bedeutung, wie z. B. die Zeitpunkte der auftretenden Restriktionsverletzungen. Diese Merkmale müssen in der Distanzfunktion berücksichtigt werden, um verwertbare Cluster und aussagekräftige charakteristische Planverläufe als Centroiden der Cluster zu erhalten.

Ein Ähnlichkeitsmaß für Bestandsverläufe wird z. B. von Holthöfer³⁶ vorgestellt. Holthöfer definiert die *Ähnlichkeit* zwischen zwei Bestandsverläufen als Unterschiede sowohl in den Fortschrittszahlen, als auch in den Zu- und Abgängen der einzelnen Planungsperioden. Die Ähnlichkeitsmetrik von Holthöfer ist in dieser Arbeit nicht zur Verwendung beim Clustering geeignet, da sie „ausschließlich auf der Grundlage von Fortschrittszahlen“³⁷ beruht. Durch diese lässt sich zwar der diskrete Materialfluss zwischen den Objektknoten bestimmen, aber in dieser Arbeit sollen weitere Merkmale, wie z. B. die Unterscheidung nach Art des angewendeten Planungsverfahrens oder der Beschaffungsart, in die Ähnlichkeitsfunktion des Clusterings mit einfließen. Diese wurden von Holthöfer nicht berücksichtigt. Zusammenfassend kann festgehalten werden, dass Holthöfers Ähnlichkeitsmetrik im Wesentlichen die quantitativen Merkmale der Pläne betrachtet, während in dieser Arbeit zusätzlich strukturelle Merkmale beim Clustering abgebildet werden sollen. Der Aufbau der Clusteringfunktion für die Arbeit bleibt offen und muss in der Konzeption aufgegriffen werden.

³⁵Eine Übersicht mit z. B. dem Dice-Maß, Jaccard-Maß, Kosinus-Maß und Weiteren ist [SM83] zu entnehmen.

³⁶Siehe [Hol00]

³⁷Ebd., S. 111

3.1.4. Konvergenz von Q-Learning auf einem abstrahierten Zustandsraum

Wird der Zustandsraum durch eines der oben diskutierten Verfahren approximiert, so ist die inhärente Konvergenzeigenschaft des Q-Learnings nicht mehr sicher gewährleistet.³⁸ Werden im konkreten Fall die Q-Werte für jeden einzelnen Zustand, oder jedes Zustands-/Aktionspaares, einzeln aktualisiert, so kann dieses den Fehler bei der Schätzung der optimalen Value-Funktion nur verringern. Leider ist diese Garantie bei Funktionsapproximationen nicht mehr gegeben, da jede Anpassung des Parametervektors gleichzeitig die approximierten Funktionswerte für andere Zustände ändert und ggf. verschlechtert.

Sabes et. al. stellen für lokale und globale Fehler zwei Konzepte vor, die Ungenauigkeit von linearer Approximation zu quantifizieren.³⁹ Boyan et. al. zeigen, dass lineare Funktionsapproximationen durch Verwendung des Value-Iteration-Verfahrens nicht robust sind und zeigen anhand von Simulationsergebnissen für drei Testprobleme die Divergenz des Lernsystems.⁴⁰ Sutton betrachtet die gleichen Testprobleme und wendet erfolgreich das CMAC-Verfahren an, um die Eignung linearer Approximatoren zu belegen.⁴¹

Baird zeigt, dass für Q-Learning mit linearen Funktionsapproximationen eine stabile Konvergenz nicht garantiert werden kann.⁴² Er demonstriert dies anhand eines MDPs mit nur sieben Zuständen und einer einfachen linearen Funktionsapproximation, für die der Fehler zwischen der approximierten und der optimalen Value-Funktion unendlich anwächst.

Der Autor stellt in der Arbeit einen neuen *residual gradient* Algorithmus vor, für den zwar die Konvergenz gezeigt werden kann, diese sich aber auf lokale Optima beschränkt. Truhn und Schwartz beobachten, dass bei der Verwendung verschiedener Approximationsverfahren beim Q-Learning die einzelnen Q-Werte tendenziell überschätzt werden.⁴³ Dieses lässt sich dadurch erklären, dass jede Approximation einen gewissen Fehler (Rauschen) in die Repräsentation der aktuellen Q-Funktion einbringt. Da es in der Natur des Q-Learnings liegt, jeweils den Folgezustand mit maximalem Q-Wert für das Q-Update einzubeziehen, neigt das Verfahren zum Überschätzen des wahren Funktionswertes. Besonders für Off-Policy-Lernverfahren wie das Q-Learning sind daher die Konvergenzeigenschaften bei der Anwendung von Funktionsapproximation Thema Forschung.⁴⁴

³⁸Siehe [Mit97]

³⁹Siehe [Sab93]

⁴⁰Siehe [BM95]

⁴¹Siehe [Sut96]

⁴²Siehe [Bai95]

⁴³Siehe [TS93]

⁴⁴Siehe z. B. [PSD01]

Bewertung

Die Abstraktion des Problemraumes kann die Ergebnisse des Q-Learnings beeinflussen. Effizientes Training ist in komplexen Zustandsräumen ohne die Anwendung zustandsreduzierender Verfahren jedoch oft unmöglich. Für die Problemstellung dieser Arbeit muss eine Abstraktion des Zustandsraumes durchgeführt werden, da der Zustandsraum „Produktionsnetzwerk“ mit seinen potenziellen Ausgangsdaten für das Lernverfahren groß ist. Bei der Definition der Abstraktionsfunktion muss darauf geachtet werden, dass diese durch eine problemspezifische Konzeption das Lernergebnis gering beeinflusst und so effizientes Lernen der Regeln möglich bleibt.

3.2. Q-Learning zum Lernen von Steuerungsregeln in Produktionsnetzwerken

In der Literatur wurden erfolgreiche Anwendungen von Q-Learning zum Lernen von Steuerungsregeln in verschiedenen Problemdomänen dokumentiert.⁴⁵ Im Folgenden werden die Verfahren aus der Literatur diskutiert, die im Bereich der Produktionsplanung und -steuerung in Produktionssystemen oder -netzwerken erfolgreich angewendet werden. Es wird speziell untersucht, ob mit den bestehenden Verfahren die Problemstellung der Arbeit gelöst werden kann bzw. Lösungsansätze auf die Problemstellung übertragen werden können.

Stegherr⁴⁶ entwickelte einen Q-Learning Ansatz zur dispositiven Auftragssteuerung in der Variantenreihenproduktion. Als Umsetzungsbasis wurde ein dezentrales Multiagentensystem gewählt, bei dem jeder Agent seine Aktionen aus einer vektorbasierten Liste auswählt. Als Teilziele des Lernsystems wurden Durchlaufzeit, Bestand, Bedienstationsauslastung und Termintreue festgelegt, welche in der Rewardfunktion zusammengeführt werden. Das Q-Update findet nach erfolgter Einlastung eines Auftrages für den jeweiligen Agenten statt. In der Rewardfunktion können Gewichtungen der einzelnen Teilziele angegeben und die Umgebungsvektoren angepasst werden, um auf veränderte Umweltsituationen reagieren zu können. Die Agenten werden anhand von Durchläufen in einer simulierten Umgebung trainiert. Nach Beendigung des Trainings wird das erlernte Wissen in die Anwendungsumgebung transformiert. Die Simulationsumgebung muss exakt der Anwendungsumgebung entsprechen, um eine gute Transformation des gelernten Wissens zu gewährleisten.

⁴⁵Z. B. die parallele Steuerung von Personenaufzügen durch Regeln von Crites und Barto [CA96]. Oder die Ressourcenverteilung zum Be- und Entladen bei Shuttle-Projekten von Zhang und Ditterich [ZD95] und andere wie [HW98] und [SB97].

⁴⁶Siehe [Ste00]

Ein Vorteil des Systems liegt darin, dass die Agenten durch das Training mittels Simulation selbstständig gute Strategien zur Auftragssteuerung entwickeln, ohne eine aufwendige Analyse des Systems durchführen zu müssen. Aus diesem Grund kann das System flexibel an neue Anforderungen angepasst werden. Nach Parameteranpassungen am Lernsystem muss eine weitere Lernphase durchlaufen werden. Danach kann das System im produktiven Betrieb eingesetzt werden. In einer beispielhaften Anwendung eines Materialflusssystemes wurden im Vergleich zu optimierten konventionellen Strategien gleich gute bzw. bessere Resultate erzielt, ohne das Produktionssystem im Voraus analysieren zu müssen. Weitere Stärken des Systems liegen in der guten situativen Entscheidungsfindung und in der Geschwindigkeit, mit der Entscheidungen erzielt wurden. Diese erfüllen die Kriterien eines Echtzeitsystems.

Stegherr versuchte, für die Auftragsplanung in der Variantenreihenproduktion eine optimale Auslastungsstrategie zu lernen. Die Problemstellung von Stegherr beleuchtet den Bereich der Neuplanung, während in dieser Arbeit die Änderungsplanung adressiert wird. Stegherrs Konzept bezieht sich auf die Produktionsplanung in einer Fabrik, während hier dezentrale Produktionsnetzwerke mit partnerspezifischen Leistungsvereinbarungen untersucht werden. Die Planung bei Stegherr verläuft bedarfsorientiert, während in dieser Arbeit die angebotsorientierte Planung berücksichtigt wird.

Stockheim et. al.⁴⁷ stellen einen Ansatz zum Q-Learning im Supply Chain Management vor. Durch Agenten wird das lokale Einlasten von Aufträgen auf Fertigungslinien gelernt und ein Sekundärbedarf auf der vorgelagerten Fertigungsstufe erzeugt. In dem zugrunde liegenden Modell werden starke Vereinfachungen hinsichtlich der durchaus komplexen Anforderungen des Supply Chain Managements⁴⁸ vorgenommen. Die Auslastung einer Fertigungslinie wird über eine einfache FIFO-Warteschlange⁴⁹ abgebildet, in der die Aufträge in Klassen aufgeteilt und dann nach Klassenpriorität eingelastet werden. Die einzulastenden Aufträge werden durch eine Zufallsfunktion erzeugt.

Das Lernsystem von Stockheim et. al.⁵⁰ steuert die Fabrik durch Löschen und neues Einlasten zufälliger Aufträge mit dem FIFO-Prinzip. Zur Lösung der Problemstellung dieser Arbeit müssen darüber hinaus weitere Strategien gelernt und nach Trainingsabschluss als Regeln repräsentiert werden. Das Konzept dieser Arbeit soll so ausgelegt werden, dass beliebige Änderungsplanungsverfahren im Lernprozess verwendet werden können. Stockheim et. al. nutzen nicht die emergenten Eigenschaften von Produktionsnetzwerken.

Riedmiller und Riedmiller⁵¹ beschreiben die Kombination eines Q-Learning Verfahrens mit einem Künstlichen Neuronalen Netz (KNN), welches unter Berücksichtigung

⁴⁷Siehe [SSK03]

⁴⁸Siehe z. B. [Sch05], [DB02], [KH02] und [WA99]

⁴⁹First-In First-Out

⁵⁰Ebd.

⁵¹Siehe [RR99]

3. Stand der Forschung

globaler Planungsziele einer Fabrik lokale Belegungspläne für Fertigungslinien erzeugt. Ziel des Lernverfahrens ist die Verkürzung der Durchlaufzeit, unter Einhaltung der Liefertermine der Aufträge, durch optimierte Einlastung wartender Aufträge bei der Belegungsplanung. Die Zielfunktion des Lernverfahrens minimiert die für jeden eingelasteten Auftrag zu veranschlagenden Kosten. Kapazitäten werden nicht berücksichtigt. Das Verfahren von Riedmiller et. al. hat keinen direkten Bezug zum Untersuchungsgegenstand der kooperativen Änderungsplanung und deren Steuerung.

Cao et. al.⁵² beschreiben ein zweistufiges Produktionsverfahren, in dem Q-Learning zum Einsatz kommt. In der ersten Fertigungsstufe werden Komponenten in Serienfertigung hergestellt, für die eine Lagerhaltung existiert. Die zweite Fertigungsstufe verbaut diese Komponenten auftragsorientiert zu Endprodukten. Für diese Fertigungsstufe existiert keine Lagerhaltung. Die Fertigungszeiten der zweiten Stufe sind gering und werden vernachlässigt. Cao et. al.⁵³ konzentrieren sich primär auf die optimale Produktionsplanung der ersten Stufe mit dem Ziel der Minimierung der Produktionskosten. Das Verfahren von Cao et. al. ist auf einzelne Fertigungsstufen fokussiert.

Ein weiterer in der Produktion erfolgreich angewendeter Q-Learning-Algorithmus ist der SMART-Algorithmus von Mahadevan et. al.⁵⁴ Mit diesem modellunabhängigen Ansatz zur Lösung kontinuierlicher Probleme wurde ein System zur Wartung von Produktionssystemen entwickelt, welches gute Resultate erzielte.⁵⁵ Das Produktionssystem besteht aus einer Maschine, die fünf unterschiedliche einzulagernde Produkte herstellen kann. Der Lagerbestand reicht stets aus, um die Marktbedürfnisse zu befriedigen. Durch die Lagerstrategie wird das Lager so lange mit dem jeweils aktuell produzierten Produkt aufgefüllt, bis es entweder voll ist oder die minimale Bestandsgrenze eines anderen Produktes unterschritten wird. Wird die Bestandsgrenze eines Produktes unterschritten, wird mit der Produktion des vakanten Produktes fortgefahren. Wenn genügend Bestände je Produkt vorhanden sind, wird die Produktion angehalten. Durch Störungen der Maschinen können sich reparaturbedingte Unterbrechungen ergeben, welche die Termintreue der Lieferung gefährden können. Der SMART-Algorithmus versucht durch präventive Wartungen Reparaturen überflüssig zu machen, um termintreue Lieferungen zu gewährleisten. Die Mindestbestände im Lager müssen je Produkt eingehalten werden und die Wartungs- und Reparaturkosten der Maschinen sollen gering ausfallen.

Zur Abstraktion des Zustandsraumes wurde die Value-Funktion durch ein mehrschichtiges KNN abgebildet. Der Reward beim Q-Update berechnet sich aus dem erwarteten Erlös aus dem Produktverkauf, abzüglich der zeitlich fakturierten Wartungs- und Reparaturkosten. Der Zustandsraum wird durch einen zehndimensionalen Vektor beschrieben, der für jedes der fünf Produkte jeweils die produzierte Menge seit der letzten War-

⁵²Siehe [CS03]

⁵³Ebd.

⁵⁴Siehe [MMDG97]

⁵⁵Vgl. [Ste00], Kap. 4.2.8

tung oder Reparatur und die aktuelle Lagermenge enthält. Der SMART-Algorithmus wurde mit zwei bekannten Heuristiken⁵⁶ verglichen und erzielte ein besseres Steuerungsverhalten. Der Fokus des SMART-Algorithmus weicht vom Fokus der Problemstellung dieser Arbeit ab. Das Konzept von Mahadevan ist eher als Metamodell zu bezeichnen.

Bewertung

Die Analyse der Verfahren unterstreicht, dass Q-Learning geeignet ist, technische Systeme und insbesondere auch Planungs- und Steuerungssysteme im Produktionsumfeld durch gelernte Regeln zu steuern. Eine Kombination der vorgestellten Verfahren aus dem Stand der Forschung ist, wie auch bei Abstraktionsverfahren, schwierig, da gerade die problemspezifische Ausgestaltung der Lernfunktion der Kern der entwickelten Verfahren darstellt.⁵⁷ Existierende Verfahren berücksichtigen die Neuplanung sowie andere Probleme der PPS und decken derzeit den Fall des Lernens bei unvollständiger Informationslage wie bei der globalen Änderungsplanung nicht ab. Die Verfahren dienen zumeist zur Steuerung lokaler Systeme.

3.3. Durchführung von Training und Generierung von Ausgangsdaten

Zum effizienten Lernen der Regeln durch Q-Learning muss ein geeignetes Trainingsverfahren konzipiert werden. Dieses soll einerseits auf einem abstrahierten Zustandsraum operieren können und andererseits relativ viele Lernschritte in kurzer Zeit durchführen können, um das Lernziel⁵⁸ effizient zu erreichen.⁵⁹ Wie z. B. Stegherr⁶⁰ und andere zeigen, dienen Simulationstechniken⁶¹ als Grundlage des Trainings.

Beim Training werden in dieser Arbeit Ereignisse durch die Anwendung von Änderungsplanungsverfahren planerisch aufgelöst. Jede Lernepisode kann als Simulationslauf im ursprünglichen Sinne verstanden werden:

Definition 3.1 (Simulation) „Simulation ist das Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierfähigen Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind.“⁶²

⁵⁶Coefficient of Operational Readiness (COR) und Age Replacement (AR)

⁵⁷Vgl. Kap. 2.2.5, S. 36 ff.

⁵⁸Vgl. Definition 2.7

⁵⁹Dieses hängt maßgeblich von der Größe des Zustandsraumes ab, wie in vorherigen Kapiteln diskutiert wurde.

⁶⁰Siehe [Ste00]

⁶¹Siehe z. B. [Lar07]

⁶²VDI 3633

Der Unterschied zwischen dem Training und der Materialflusssimulation ist, dass nicht das zu erwartende Verhalten des Systems mithilfe der Simulation und unter Modellierung stochastischer Einflüsse simuliert werden muss, sondern mit Hilfe von Realdaten aus der Produktion Planungsläufe durchgeführt werden müssen. Es muss kein vollständiges Simulationsmodell vorliegen, um das Training durchführen zu können, da die hier vorgestellte Granularität des Modells der Fertigung genügt.⁶³ Es ist ausreichend, das Wissen über das Produktionsnetzwerk, die Partner und deren Leistungsvereinbarungen sowie die Restriktionen zu berücksichtigen.

Die Verwendung von Materialflusssimulationswerkzeugen⁶⁴ wie z. B. EMPlant⁶⁵, EnterpriseDynamics⁶⁶ oder d3FactInsight⁶⁷ zur Umsetzung des Trainers stellt sich als schwierig dar, da der Sourcecode nicht vorliegt oder das Werkzeug, im Falle von d3FactInsight, nur im Alphastadium der Entwicklung vorliegt.

3.3.1. Training mit einer Multiagentensystemarchitektur

Multiagentensysteme (MAS) in der Produktionsplanung und -steuerung sind in inner- und überbetriebliche Konzepte klassifizierbar.⁶⁸ Innerbetrieblich fokussierte MAS zielen auf unternehmensinterne Prozesse ab, während überbetrieblich fokussierte MAS eine unternehmensübergreifende Prozesskoordination ermöglichen sollen.

Stiefbold⁶⁹ stellt mit der *Agile Agent Control Environment* ein Konzept zum Management von Lieferketten mithilfe eines MAS vor, dessen Agenten an ein MRP-System gekoppelt werden. Das MAS übernimmt anstatt eines MRP-Systems die Funktion, für Lieferabrufe unter Kapazitäts- und Ressourcenrestriktionen eine Neuplanung verhandlungsbasiert durchzuführen.⁷⁰ Da bei einer Neuplanung der Plan vollständig neu erzeugt wird, steht das Ergebnis einer Neuplanung im Konflikt zu dem Ergebnis einer Änderungsplanung. Die Reaktion auf ungünstige Zustände, wie sie in der Änderungsplanung im Sinne der Problemstellung durchgeführt wird, wird nicht umgesetzt. Das verwendete Verhandlungsprotokoll wurde durch eine Skriptsprache implementiert. Diese ist nicht ausreichend dokumentiert, was eine einfache Adaptierung der Verhandlungssteuerung erschwert. Die Möglichkeit einer externen Integration weiterer Funktionsmodule, wie dem Lernverfahren, bleibt offen.

⁶³Siehe Kap. 2.1, S. 9

⁶⁴Vgl. Abb. 2.9, S. 45

⁶⁵<http://www.emplant.de>

⁶⁶<http://enterprisedynamics.com/>

⁶⁷Siehe [Lar07]

⁶⁸Siehe [Hei06]

⁶⁹Siehe [Sti98]

⁷⁰Klassische MRP-Systeme führen vorrangig Neuplanungen durch (Vgl. [Sch05], S. 397 ff.).

Mannmeusel⁷¹ stellt mit *Planet AS* die prototypische Umsetzung eines MAS zur dezentralen und verhandlungsbasierten Steuerung von Produktionssystemen vor. Durch *Planet AS* wurde konzeptuell ein allgemeines Vorgehensmodell zur Erstellung dezentraler Produktionssteuerungssysteme erarbeitet. Im Fokus stand die Konzeption von Koordinationsprotokollen zur bilateralen Verhandlung über Auftragsvergaben in einem Produktionsnetzwerk. Mögliche Engpässe im Netzwerk wurden zentralistisch ermittelt, wobei kurzfristige sowie angebotsseitige Planänderungen nicht beachtet wurden. Zur Prüfung der Kapazitäts- und Materialverfügbarkeit einzelner Aufträge wurden Standardverfahren aus der Produktionsprogrammplanung⁷² verwendet. Ein System zur Steuerung der Produktion, sowie Schnittstellen zur Integration von Regelsystemen, wurden nicht umgesetzt.

Dudenhausen⁷³ stellt mit *X-CITTIC* ein MAS zur kollaborativen Auftragskoordination für Prozessketten der Halbleiterindustrie vor. Globale Aufträge werden in kleine Teilaufträge gesplittet und dann an die einzelnen Unternehmen weitergegeben. Jedes Unternehmen wird durch einen Agenten repräsentiert. Die Verhandlungen über die Weitergabe der Lieferdaten erfolgen über einen Brokeragenten, an den die Produktionsagenten angeschlossen sind. Der Verhandlungsrahmen wird durch ein wohldefiniertes Protokoll geregelt. Die Integration eines Regelsystems zur Steuerung der Planungsabläufe wurde nicht vorgenommen.

*Agent.Enterprise*⁷⁴ ist ein Multi-Multi-Agentensystem zur hierarchischen Koordination einer Supply Chain. In *Agent.Enterprise* werden unternehmensspezifische Aufgaben durch spezialisierte, unternehmensinterne MAS umgesetzt und etwaige Ergebnisse durch das unternehmensübergreifende *MAS DISPO WEB*⁷⁵ an betroffene unternehmensinterne MAS kommuniziert. Bei der Produktionsplanung erstellt *MAS DISPO WEB* durch Sukzessivplanung über alle relevanten internen MAS einen initialen Plan, der für alle weiteren Berechnungen verwendet wird. In der Konsequenz arbeitet *Agent.Enterprise* auf unternehmensübergreifender Ebene bedarfsorientiert. Eine angebotsorientierte Planung ist nicht vorgesehen. Ein konfigurierbares Steuerungssystem ist nicht vorhanden und keine Schnittstelle, um gelernte Regeln zur Steuerung der Koordination einzubinden.

Pape et. al. stellen mit *COAGENS*⁷⁶ ein MAS zur Steuerung des Lieferanten- und Beschaffungsmanagements in der Serienfertigung vor. Einzelne Agenten modellieren jeweils Abnehmer, Spediteure und Lieferanten. Sie interagieren kooperativ innerhalb eines sequenziellen Protokolls zur Planung des Auftragsdurchlaufes, vom Lieferanten

⁷¹Siehe [Man97]

⁷²Siehe z. B. [Tem06]

⁷³Siehe [Dud99]

⁷⁴*Agent.Enterprise* ist das Ergebnis eines Projektes gefördert durch die DFG im Rahmen des SPP 1083 „Intelligente Softwareagenten und betriebswirtschaftliche Anwendungsszenarien“.

⁷⁵Siehe [SNSS04]

⁷⁶Siehe z. B. [DPR04] oder [Pap06]

3. Stand der Forschung

über den Spediteur bis zum Kunden. Da das Protokoll spezialisiert ist, entspricht es nicht der hier verwendeten kooperativen Änderungsplanung. Eine Anpassung würde tief greifende Änderungen in COAGENS bedingen, die für diese Arbeit wegen des hohen Aufwandes keinen zielführenden Nutzen mit sich bringen.

Heidenreich⁷⁷ stellt in seiner Arbeit mit *MASCOPP* die prototypische Umsetzung des in dieser Arbeit referenzierten kooperativen Änderungsplanungsverfahrens vor. Es werden angebots- und bedarfsseitige Koordinationen berücksichtigt. Diverse Änderungsplanungsverfahren sind in *MASCOPP* umgesetzt. Die Abfolge der Änderungsplanung ist durch ein manuell konfigurierbares zustandsbasiertes Regelsystem steuerbar. *MASCOPP* verwendet ein im Heinz Nixdorf Institut der Universität Paderborn mit der Programmiersprache „Microsoft.Net“ entwickeltes Multiagentensystem.⁷⁸ Durch die objektorientierte Programmierung ist das System theoretisch beliebig erweiterbar, wobei keine Standards wie FIPA⁷⁹ zur Kommunikation zwischen den Agenten verwendet werden. Die Erweiterung der Kommunikationsprotokolle von *MASCOPP* zur Implementation dieses Lernsystems stellt sich als schwierig dar. Um im Training Agenten übergreifende Lernepisoden ausführen zu können, müssen die durchgeführten Aktionen gespeichert werden, um eine erneute Ausführung zu verhindern. Bei *MASCOPP* fehlt die Definition eines Protokolls für die Verarbeitung der Antworten für globale Änderungsplanungen über mehrere vorgelagerte Agenten. Dessen ungeachtet erfüllt *MASCOPP* aus Sicht der Änderungsplanung die meisten Anforderungen dieser Arbeit. Die aufwendige Programmierung der Plattform und die knappe Dokumentation erschweren die Erweiterung von *MASCOPP*.

Bewertung

Die Implementation eines eigenen MAS zur Umsetzung des Lernkonzeptes sinnvoll, da die oben analysierten MAS nicht oder nur bedingt zur Umsetzung der Problemstellung geeignet sind. Problematisch sind insbesondere die Verfügbarkeit des Quellcodes, die Dokumentation und die Kompatibilität des Datenmodells, sofern bekannt, um die Anforderungen dieser Arbeit zielführend zu implementieren. *MASCOOP* ist im Prinzip geeignet, jedoch unzureichend dokumentiert und durch die verwendete Programmiersprache *Visual Basic.NET* nicht so offen zu erweitern wie eine *JAVA*-Anwendung.

Zur Umsetzung einer flexiblen Lösung, die auch für weiterführende Forschungsfragen in diesem Kontext genutzt werden kann, eignet sich die in *JAVA* implementierte *MAS-API JADE*⁸⁰. *JADE* ist eine weltweit verwendete und umfassend dokumentierte

⁷⁷Siehe [Hei06]

⁷⁸Siehe [Fra04]

⁷⁹Siehe z. B. [Fou97] oder [IA00]

⁸⁰Siehe [Til]

MAS-API, die durch ein renommiertes Konsortium, u. a. mit IBM, entwickelt wurde. JADE folgt dem FIPA-Standard⁸¹ und ist objektorientiert erweiterbar. JADE bietet eine umfassende Funktionsbibliothek⁸², durch die der Entwickler bei seiner Arbeit unterstützt wird. Die notwendigen Protokolle zur Umsetzung des Lernsystems⁸³ und der Änderungsplanung sind in JADE vorhanden und können mit der JADE-API durch eigene Nachrichtentypen erweitert werden.

3.3.2. Ausgangsdaten für das Training

Vorhandene reale Ausgangsdaten können aus unternehmensinternen ERP-Systemen⁸⁴ extrahiert werden. Der Zugriff auf ERP-Systeme kann proprietär⁸⁵ in das System selbst oder über eine Programmier- und Datenbankschnittstelle⁸⁶ erfolgen. Bei der Verwendung des proprietären Zugriffs müsste das Lernsystem programmiertechnisch in das ERP-System integriert werden, um einen Zugriff auf die Datenbank zu ermöglichen. Da das Lernsystem über ein MAS umgesetzt werden soll, ist dieses nur unter großem Aufwand möglich, da jedes ERP-System eine angepasste Implementation benötigen würde.

Der Zugriff auf die Daten kann über definierte Schnittstellen erfolgen, mit dem Vorteil, dass die Umsetzung der Funktionen des Lernsystems unabhängig von konkreten ERP-Systemen durchgeführt werden kann. Es ist vielmehr entscheidend, dass die benötigten Ausgangsdaten für das Training vollständig definiert werden. Zur Definition der Datenstrukturen der Ausgangsdaten bietet sich UML⁸⁷ als einheitliche Beschreibungssprache und Quasistandard in der Softwareindustrie an.⁸⁸ Mithilfe einer definierten Schnittstelle kann analysiert werden, ob in einem ERP-System die notwendigen Ausgangsdaten für das Lernsystem vorhanden sind.

⁸¹Siehe z. B. [Fou97] oder [IA00]

⁸²Z. B. ein Agentenmanager, Gelbe Seiten, grafische Konfigurationsoberflächen und weitere Funktionen

⁸³Im Wesentlichen Anfrage → Antwort → Auswertung der Antwort

⁸⁴ERP: Enterprise Resource Planning. Betriebliche Standardsoftware, die neben anderen Daten und Funktionen die zur Produktionsplanung und -überwachung relevanten Stamm- und Bewegungsdaten sowie Funktionen zur deren automatisierten oder manuellen Verarbeitung bereitstellt (Siehe [SW01]). Diese sind z. B. SAP (<http://www.sap.com>), SAGE (<http://www.sage.de>) oder NAV (<http://www.microsoft.com/germany/dynamics/nav/>).

⁸⁵Als proprietär bezeichnet man Hardware oder Software, die nur auf einem spezifischen System verwendbar und nicht kompatibel zu anderer Hard- oder Software ist.

⁸⁶„Die Programmierschnittstelle (Application Programming Interface, API) definiert in ihrer Syntax und Semantik die Funktionen des Betriebssystems in Form von Systemdiensten (System Services).“ ([SW01], S. 237). „Datenbankschnittstellen erlauben Applikationsprogrammen den Zugriff auf verschiedenartige Datenquellen (Datenbanken, tabellarisch strukturierte Quellen, Text).“ ([SW01], S. 502).

⁸⁷Unified Modelling Language, siehe z. B. in [Jec04] oder [SG00]

⁸⁸Vgl. [SW01], S. 339

3.4. Zusammenfassung

Keiner der analysierten Ansätze aus den Bereichen der Abstraktionsverfahren oder des Q-Learnings für das Lernen von Regeln zu Steuerung der Änderungsplanung für Produktionsnetzwerke ist vollständig bzw. ohne notwendige Anpassungen geeignet, um die Forschungsfragen A und B aufzulösen. Auch die Kombination verschiedener Ansätze ist problematisch, da sowohl für Abstraktionsverfahren als auch für das Q-Learning problemspezifische Funktionen entwickelt werden müssen.

Ein wesentliches Hemmnis der vorgestellten Abstraktionsverfahren ist, dass die charakteristischen Eigenschaften des Produktionsnetzwerkes bei der Abstraktion nicht explizit erhalten bleiben, da durch sie eine implizite Repräsentation der Value-Funktion umgesetzt wurde. Wurden die abstrahierten Zustände des Zustandsraumes explizit, z. B. durch einen Eigenschaftsvektor modelliert, so war diese Repräsentation für die Problemstellung nicht anwendbar.

Für den Bereich der Lernfunktion konnten erfolgreiche Anwendungen von Q-Learning im industriellen Umfeld und speziell zur Steuerung von Planungsprozessen in Produktionssystemen identifiziert werden. Die aufgeführten Arbeiten betrachteten zentral lös-bare Planungs- und Steuerungsprobleme, die wegen der spezialisierten Lernfunktion nicht auf diese Arbeit übertragen werden können.

Zur Umsetzung von Produktionsplanung und -steuerungssystemen durch MAS wurden in der Literatur verschiedene, einfache wie komplexe Ansätze beschrieben. Wesentliche Hemmnisse einer potenziellen Integration des Lernsystems in diese bestehenden Ansätze sind ⁸⁹

- mangelnde Datenkompatibilität,
- mangelnde Protokolle und
- unzureichend dokumentierte oder aufwendig anzuwendende Programmierschnittstellen.

JADE wurde als generisch verwendbare MAS-API zur Implementation des Lernsystems vorgeschlagen, wobei eine spätere Erweiterbarkeit des Lernsystems durch die Verwendung von Standards⁹⁰ und eine umfangreiche Dokumentation von JADE vereinfacht wird.

Reale Ausgangsdaten müssen über eine definierte Schnittstelle aus ERP-Systemen extrahiert werden. Eine Einbettung des Lernverfahrens in ein ERP-System ist nicht sinnvoll, da der Aufwand einer solchen Umsetzung zu hoch und die Erweiterungsflexibilität des Systems gering ist.

⁸⁹Siehe [Hei06]

⁹⁰Siehe z. B. [Fou97]

4. Zu leistende Arbeit

Die Neugier steht immer an erster Stelle eines Problems, das gelöst werden will.

(Galileo Galilei)

Bei der Verwendung des Q-Learning-Verfahrens ist die Komplexität des Zustandsraumes, in dem gelernt wird als anwendungsübergreifendes Problem anzusehen. Die bisherigen Anwendungen von Q-Learning zeigten, dass die Reduzierung komplexer Zustandsräume auf ein berechenbares Maß möglich, aber keines der existierenden Verfahren auf die Problemstellung direkt übertragbar ist. In dieser Arbeit muss ein spezielles Verfahren umgesetzt werden, welches die Komplexität des Zustandsraumes für den Untersuchungsgegenstand derart reduziert, dass das Q-Learning-Verfahren in akzeptabler Zeit anwendbare Regeln zur Steuerung der Änderungsplanung erzeugt. Dieses geschieht durch problemspezifische Abstraktion des Zustandsraumes, bei der die charakteristischen Merkmale von Produktionsnetzwerken berücksichtigt und genutzt werden. Es besteht die Aufgabe, durch ein Abstraktionsverfahren Ausgangsdaten für das Lernverfahren zu erzeugen, die effizientes Lernen auf einem skalierten Zustandsraum ermöglichen. Hierzu werden diskrete ungültige Zustände auf charakteristische ungültige Zustände des Produktionsnetzwerkes durch das Abstraktionsverfahren eindeutig abgebildet.

Die Analyse geeigneter Abstraktionsverfahren in Kapitel 3.1.2 zeigte die Eignung der Methode „Clustering“ zur Ausgestaltung des Abstraktionsverfahrens. Es ist kein problemspezifisches Clusteringverfahren für Produktionsnetzwerke bekannt, dessen Abstraktionsfunktion, auch als Distanzfunktion bezeichnet, unmittelbar auf die Problemstellung übertragen werden kann. Die Definition der Distanzfunktion zur problemspezifischen Abstraktion des Zustandsraumes ist daher wichtiger Bestandteil dieser Arbeit. Die während der Abstraktion durchgeführte Abbildung des Ursprungszustandsraumes auf den abstrahierten Zustandsraum soll eindeutig, vollständig, betriebswirtschaftlich interpretierbar und benutzerspezifisch skalierbar sein. Negative Effekte auf die Lernfunktion durch abstraktionsbedingten Informationsverlust müssen minimiert werden, indem in der Spezifikation der Distanzfunktion die charakteristischen Merkmale der Zustände in Produktionsnetzwerken berücksichtigt werden. Die zu entwickelnde Distanzfunktion soll für das *k-means*-Clusteringverfahren umgesetzt werden. Die zum Clustering verwendeten Ausgangsdaten müssen realitätsnah sein. Hierzu eignen sich Realdaten, z. B. aus Datenbanken von ERP-Systemen.

Als Lernverfahren wurde Q-Learning ausgewählt. Die Leistungsfähigkeit des Q-Learnings hängt von der problemorientierten Definition der Rewardfunktion und der Qualität der im Training verwendeten Ausgangsdaten ab. Die Analyse zeigte, dass eine problemspezifische Rewardfunktion durch eine Kostenfunktion repräsentiert werden kann, um die Planungsaktionen und deren Konsequenzen vergleichbar bewerten zu können. Hierzu wurden Bereitstellungskosten, Betriebsmittelkosten, Beschaffungskosten über Leistungsvereinbarungen und Restriktionsverletzungen als relevante Kostenfaktoren je Objektknoten herausgearbeitet. Der Trainingsablauf wird durch einzelne Lernepisoden bestimmt. Die Ausgangsdaten der Lernepisoden sowie deren Start- und Endzeitpunkt sind Bestandteil der problemorientierten Ausgestaltung der Lernepisoden. Es sind darüber hinaus Abbruchbedingungen für die Lernepisoden und den Trainingsprozess zu definieren, die ein definiertes Ende des Gesamten oder von Teilen des Lernprozesses ermöglichen. Es ist eine Schnittstelle zu konzipieren, die ein Konzept zur Verwendung von Realdaten aus ERP-Systemen im Lernverfahren aufzeigt. Die Regeln müssen durch ein Verfahren aus den gelernten Q-Werten erzeugt werden können. Umgekehrt muss ein Verfahren konzipiert werden, dass in der Anwendung des gelernten Regelsystems die passende Regel zur Auswahl eines Änderungsplanungsverfahrens zur Steuerung der Änderungsplanung bestimmt.

Um die Leistung des Lernverfahrens zu validieren, ist es notwendig, dieses zu implementieren und praktisch zu evaluieren. Die Implementation wird mithilfe eines Multiagentensystems unter Verwendung der JADE-API durchgeführt. Kern der Validierung ist die Analyse der Effektivität und der Effizienz der Distanzfunktion und des Lernsystems. Die Effektivität der Zustandsabstraktion kann durch den analytischen Vergleich der Lernergebnisse auf diskreter und abstrahierter Zustandsebene erfolgen. Die Effektivität des Lernverfahrens kann durch die Analyse der erzeugten Regeln hinsichtlich ihrer Anwendbarkeit zur Steuerung der Änderungsplanung validiert werden. Die Effizienz des Verfahrens wird bereits durch Aufzeigen der Effektivität des Abstraktions- und Lernverfahrens untermauert. Es bleibt zu zeigen, dass der Lernprozess gegenüber der manuellen Regelsystemerstellung Performancevorteile mit sich bringt und somit der Konfigurationsaufwand der Teilmodule des Lernsystems gerechtfertigt ist. Abschließend soll das Konvergenzverhalten des Lernsystems untersucht werden.

Abbildung 4.1 zeigt den Ablauf der zu leistenden Arbeit, nach dem die Konzeption in Kapitel 5 und die Validierung der Konzepte in Kapitel 6 gegliedert ist. ¹

¹Die Dokumentation der Implementation findet sich im Anhang C, S. 213 ff.

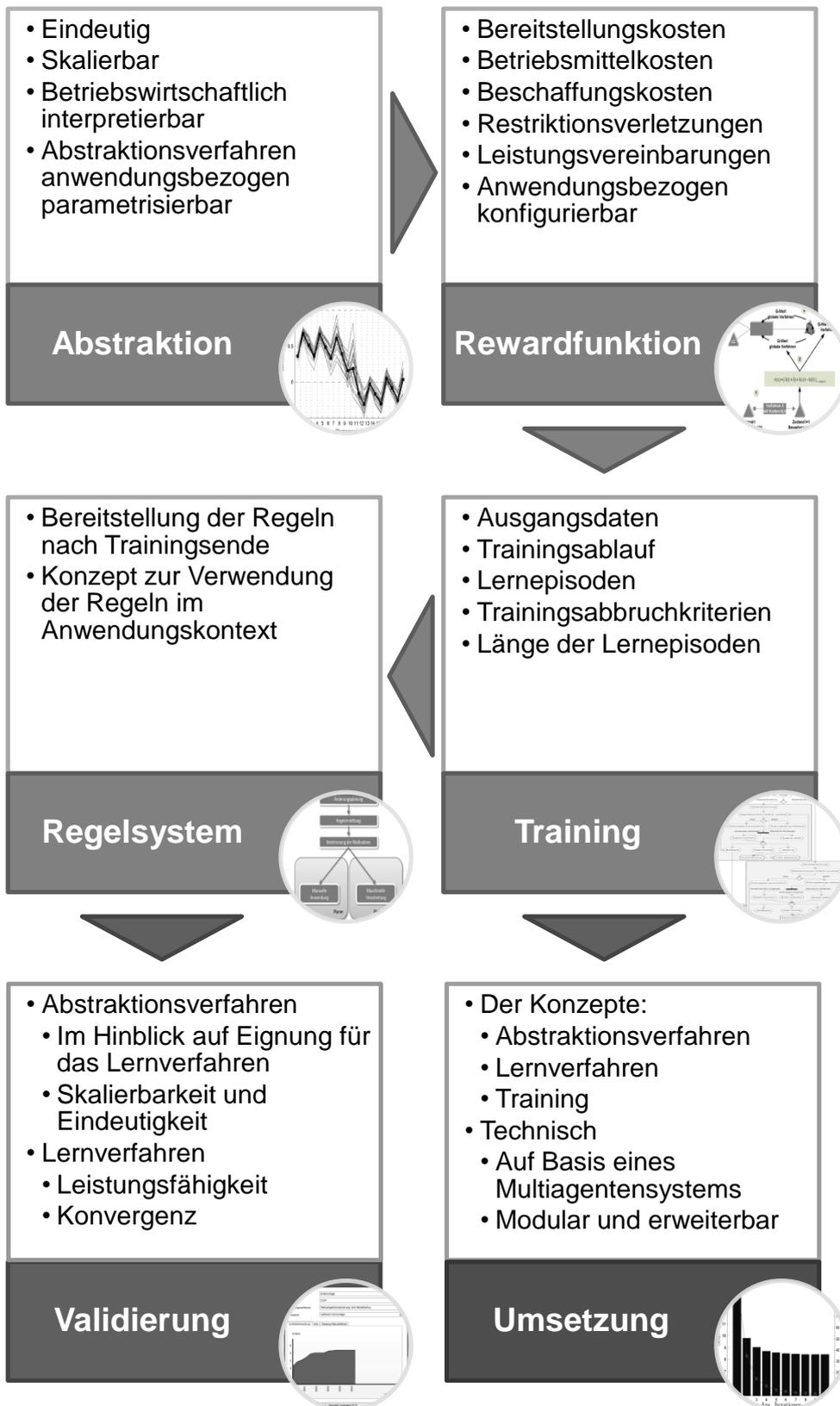


Abbildung 4.1.: Überblick über die Abfolge der zu leistenden Arbeit

4. *Zu leistende Arbeit*

5. Konzeption

We have to learn not to learn
what we learn.

(Marvin Minsky)

In diesem Kapitel werden die in Kapitel 4 identifizierten zu leistenden Arbeiten konzeptionell dargestellt. In Kapitel 5.1 wird das Abstraktionsverfahren konzipiert. Zunächst wird das grundlegende Konzept der Abstraktionsfunktion erarbeitet (Kapitel 5.1.2) und dann über die Entwicklung der Distanzfunktion und des Trainingsverfahrens das Konzept des Clusterings detailliert (Kapitel 5.1.3). Danach folgt eine Zusammenfassung des Konzeptes des Abstraktionsverfahrens (Kapitel 5.1.4).

In Kapitel 5.2 wird das Lernverfahren konzipiert. Die Arbeitsschritte gliedern sich dabei in die Basiskonzeption des Verfahrens (Kapitel 5.2.1 – 5.2.5), Bewertung von Restriktionsverletzungen in der Rewardfunktion (Kapitel 5.2.6) und Bewertung von Kosten in der Rewardfunktion je Objektknotentyp sowie unterschieden nach lokaler und globaler Änderungsplanung (Kapitel 5.2.7 – Kapitel 5.2.9). Weiterhin werden zusätzliche Anforderungen des Q-Learnings an die Konzeption, wie z. B. die Bewertung von Endzuständen, diskutiert und die Ergebnisse zusammengefasst (Kapitel 5.2.10 – Kapitel 5.2.12).

In Kapitel 5.3 wird das Training des Lernverfahrens erarbeitet. Hierzu müssen die Lernepisoden konzipiert werden (Kapitel 5.3.1) und es muss erörtert werden, wie diese in das Training integriert und der Gesamtprozess des Trainings parametrisiert werden kann (Kapitel 5.3.2). Für die Anwendung der Regeln muss das Regelsystem durch ein Verfahren erstellt und in der Anwendung passende Änderungsplanungsverfahren für einen ungültigen Zustand ermittelt werden können (Kapitel 5.3.3). Abschließend wird das Konvergenzverhalten des Lernsystems analysiert und es werden die Arbeiten in diesem Kapitel zusammengefasst (Kapitel 5.3.4 und Kapitel 5.3.5). Abschließend folgt in Kapitel 5.4 eine tabellarische Zusammenfassung und Einordnung der Ergebnisse der Konzeption des gesamten Lernsystems.

5.1. Reduktion des Zustandsraumes durch Clustering

Die Analyse existierender Abstraktionsverfahren in Kapitel 3.1 hat gezeigt, dass die meisten alternativen Approximationsverfahren für den hier betrachteten Anwendungsfall Nachteile aufweisen. Dieses liegt insbesondere daran, dass die als Ausgangsdaten für das Clustering dienenden ungültigen Zustände¹ eines Produktionsnetzwerkes durch eine Menge inhomogener und nicht numerischer Merkmale modelliert werden, für deren Abstraktion sich lineare Approximationen und neuronale Netzwerke nur begrenzt eignen. Zu diesen Merkmalen gehören z. B. die mögliche Bereitstellungsmenge und der Leistungsgrad von Plänen verschiedener Objektknoten.

5.1.1. Vorüberlegungen

Die Abstraktion von Zuständen in Produktionsnetzwerken ist als Untersuchungsgegenstand herausfordernd, da geringe Zustandsänderungen einen gültigen in einen ungültigen Plan überführen können und ein Produktionsnetzwerk dadurch viele ungültige Zustände annehmen kann, die sinnvoll in eine abstrakte Repräsentation überführt werden müssen. Im Gegensatz zur großen Anzahl möglicher Zustände in einem Produktionsnetzwerk ist festzustellen, dass die Anwendbarkeit von Änderungsplanungsverfahren auf bestimmte Arten von Zuständen beschränkt ist.² Zum Beispiel erzeugt eine Planänderung von einer oder zwei eingeplanten Materialeinheiten im Plan eines FOK zwei verschiedene neue Zustände. Die Entscheidung über die Anwendung eines Planungsverfahrens würde bei solchen Änderungen eines Zustandes dennoch sehr wahrscheinlich gleich ausfallen.

5.1.1.1. Ausreißer

Der oben beschriebene Sachverhalt kann so interpretiert werden, dass gering differenzierbare Zustände vom Lernverfahren nicht separat betrachtet werden müssen. Zustände mit marginalen Unterschieden können zu einem Zustand, dem sogenannten *Cluster*, abstrahiert werden. Die Abstraktionsfunktion muss folglich so konzipiert werden, dass die zugewiesenen Zustände eindeutig einem Cluster zuweisbar sind.³ Dennoch sind Ausgangszustände zu erwarten, deren Zuordnung zu einem Cluster als „Ausreißer“ im Sinne des charakteristischen Planes eines Clusters interpretiert werden können. Aus

¹Das gilt ebenso für gültige Zustände, die grundsätzlich auch mit dem Clusterverfahren abstrahiert werden können.

²Siehe Kap. 2.1.3

³Vgl. auch Kap. 2.3.1

Sicht des Clusters liegen diese Ausgangsdaten nach ihrer Zuweisung zu einem Cluster am Rand dieses Clusters, also maximal entfernt vom Centroiden des Clusters. Dieses Problem kann nicht grundsätzlich behoben werden, da durch die Abstraktion Ungenauigkeiten zugunsten eines effizienten Lernprozesses auf einem reduzierten Zustandsraum erkaufte werden müssen. Dennoch kann dieses durch eine problemspezifische Abstraktionsfunktion abgemildert werden. Weiterhin ist es sinnvoll, die Anzahl zu erzeugender Cluster während des Abstraktionsprozesses parametrisierbar zu gestalten. Durch eine höhere Anzahl der Cluster können „Ausreißer“ über alle Cluster besser verteilt und somit Extremwerte geglättet werden.⁴

5.1.1.2. Prinzipablauf und Nutzen des Clustering

Das Clusterverfahren partitioniert den gesamten Zustandsraum der Problemdomäne in disjunkte Zustandscluster. Es kann als eine Abbildung $f: \mathcal{S} \mapsto \{S_1, \dots, S_m\}$ aus dem Zustandsraum \mathcal{S} in die Menge der betrachteten Zustandscluster $\{S_1, \dots, S_m\}$ aufgefasst werden, die jeden Zustand genau einem Cluster zuordnet.⁵ Diese bedeutet für die resultierenden Cluster, dass sie jeweils disjunkte Teilmengen des Zustandsraumes sind und ihre Vereinigung gerade wieder den gesamten Zustandsraum beschreibt, was formal als $S_i \cap S_j = \emptyset$ für alle $i \neq j$ und $S_1 \cup S_2 \dots \cup S_m = \mathcal{S}$ ausgedrückt werden kann. Das Ziel bei diesem Vorgehen ist, eine signifikante Reduktion des von dem Lernverfahren betrachteten Zustandsraumes $m \ll |\mathcal{S}|$ zu erreichen, sodass die resultierenden Cluster als einzelne Einträge einer Datenbanktabelle im Q-Learning verwaltet werden können.⁶

Der Vorteil der Abstraktion von Zuständen in Produktionsnetzwerken zu Clustern liegt darin, dass beim Design der Cluster und deren Merkmalen genau diejenigen Merkmale der Zustände explizit berücksichtigt werden können, die für das Lernverfahren notwendig sind. Zustände eines Produktionsnetzwerkes sind aus einer Menge von Merkmalen und deren Belegungen eindeutig beschreibbar.⁷ Durch die problemspezifische Abstraktion der Ausgangszustände kann das Lernverfahren fortan auf den im Clustering erzeugten charakteristischen Zuständen lernen. Die im Q-Learning approximierten Q-Werte können daher auf Clusterebene als *Cluster-/Aktionspaare* gespeichert werden. Da Pläne für FOK und KOK aus Sicht der Datenstruktur analog definiert wurden, wird im Clustering nicht zwischen FOK und KOK unterschieden.⁸

⁴Dieses wird u. a. in der Validierung in Kap. 6.1.2.1 untersucht.

⁵Abb. 3.3, S. 57 zeigte dieses bereits.

⁶Siehe hierzu Forschungsfrage A in Kap. 2.3.1.

⁷Vgl. [Sch99a]

⁸Vgl. Kap. 2.1.2, S. 14. Die Verläufe beider Planarten können im Abstraktionsverfahren gleichartig behandelt werden, da durch die Abstraktion die Verläufe selbst abstrahiert werden und nicht die Semantik der Pläne, wie z. B. Bestandsmenge oder eingeladete Kapazität, betrachtet wird. Im Folgenden wird daher allgemein von Plänen gesprochen.

5.1.2. Aufbau der Abstraktionsfunktion

Um die Merkmale einzelner ungültiger Zustände⁹ eines Produktionsnetzwerkes zu abstrahieren, muss analysiert werden, welche Merkmale dieser Zustände wie zusammengefasst werden können. Jeder in der Änderungsplanung und so auch im Lernverfahren zu verarbeitende Ausgangszustand resultiert aus einer im Plan eines Objektknotens eingerechneten Änderungsanfrage. Die Arten der Änderungen und die dadurch entstehenden Unterschiede in den resultierenden Zuständen werden als sogenannte *weiche Kriterien* in der Abstraktionsfunktion des Clusterings repräsentiert.

Nach Heidenreich¹⁰ sind die Varianten der anwendbaren Änderungsplanungsverfahren auf bestimmte Arten von Zuständen einschränkbar. Ein Merkmal für diese Unterscheidung ist dabei der anfragende Objektknoten. Ein weiteres ist die Art der Planänderung. Diese beiden Kriterien der Zustandsbeschreibung¹¹ werden als sogenannte *harte Kriterien* in der Abstraktionsfunktion des Clusterings definiert. Die Differenzierung dieser und weiterer Merkmale zur Spezifikation der Abstraktionsfunktion sind Diskussionsgegenstand des folgenden Kapitels. Dabei werden bei der Detaillierung der Abstraktionsfunktion in Kapitel 5.1.3 verwendeten Konzepte vorbereitend hergeleitet und erläutert.

5.1.2.1. Auswahl relevanter Merkmale für die Zustandsbeschreibung

Als erster Schritt zur Reduktion der Komplexität des Zustandsraumes muss eine Auswahl der relevanten Merkmale zur Beschreibung eines Zustandes vorgenommen werden. Die eingeführte Definition 2.4¹² für Zustände eines Produktionsnetzwerkes wird als Grundlage der Auswahl relevanter Merkmale zur Zustandsabstraktion verwendet.

Es werden nur die Merkmale in der Abstraktionsfunktion berücksichtigt, die das Lernverfahren benötigt, um die zielführenden Planungsverfahren während des Lernprozesses auswählen zu können. Die Menge anwendbarer Aktionen wird von statischen, objektknotenspezifischen Attributen beeinflusst.¹³ Diese Art von Merkmalen kann an bestimmten Objektknoten zu nicht anwendbaren Aktionen führen. Wenn ein FOK, z. B. bei einem vorgelagerten Objektknoten, Bedarfe nur nach dem Bestellzyklusverfahren anmeldet, fallen alle Nettobedarfsänderungen nach dem Bestellpunktverfahren aus der Menge der anwendbaren Aktionen heraus. Die als unzulässig ausgeschlossenen Aktionen stehen während des Lernprozesses nicht zur Verfügung.

⁹Zur besseren Lesbarkeit ist mit Zustand im Folgenden stets ein ungültiger Zustand als Ausgangsdatum für das Clustering gemeint. Ist dieses nicht der Fall, so wird eine explizite Differenzierung des Begriffs vorgenommen.

¹⁰[Hei06]

¹¹Vgl. Definition 2.4, S. 16

¹²Siehe Kap. 2.4, S. 16

¹³Siehe Darstellung der Planungsstrategien in Abb. 2.4 und Kap. 2.1.3, S. 18 ff.

Für die Clusterabbildung reicht es aus, sich für die Bestimmung anwendbarer Aktionen auf die Merkmale zu beschränken, durch deren Änderung die Anzahl der Zustände eines Objektknotens beeinflusst wird. Diese sind die Merkmale:

1. Art der Planänderung und
2. anfragender Objektknoten.¹⁴

Zur Auswahl der möglichen Aktionen ist theoretisch auch die Menge der während einer Koordinationsphase ausgeführten Aktionen relevant. Anhand dieser Menge kann sichergestellt werden, dass eine Aktion nicht mehrfach ausgeführt wird. Die Unterscheidung aller Teilmengen von möglichen Aktionen würde den Zustandsraum deutlich vergrößern. Es kann darauf verzichtet werden, da dieses durch die zustandsspezifischen Regeln des Lernverfahrens selbst gesteuert werden kann.

Zur Bewertung einzelner Zustände im Lernprozess sind Informationen über die zugehörigen Pläne notwendig. Ein Plan gibt für jedes Planungsintervall den zugehörigen Bestand im Fall von Fertigungsobjektknoten bzw. die zugewiesene Kapazität im Fall von Kapazitätsobjektknoten an. Die Pläne sind mit Abstand die bedeutendste Ursache für die Größe des Zustandsraumes, da die kombinatorischen Möglichkeiten, die aus der Angabe von Beständen bzw. Kapazitäten für jedes Planungsintervall resultieren, die Menge von möglichen Zuständen unendlich werden lassen können.

Selbst wenn diese Angaben auf begrenzte Intervalle mit diskreten Werten beschränkt werden, wächst die Anzahl möglicher Pläne exponentiell mit dem Planungshorizont.¹⁵ Weiterhin muss die Repräsentation von Plänen im Clustering für alle lernenden Objektknoten im Produktionsnetzwerk anwendbar sein. Diese sind die Fertigungs- und Kapazitätsobjektknoten, die je nach Belegung Pläne mit unterschiedlichen Planwerten und Konfigurationen aufweisen. So können die Bestandshöhen und Restriktionsgrenzen an den FOK unterschiedlich ausfallen, da die Höhe der Lagerbestände stark von der Art des gelagerten Materials und der definierten Leistungsvereinbarungen mit den Kunden und Lieferanten abhängt.

Da die Abstraktion für das Lernverfahren gleichermaßen für Fertigungs- und Kapazitätsobjektknoten anwendbar sein muss, darf sie keine Annahmen über die zugrunde liegenden Pläne machen. Ebenso müssen die Pläne für die Umsetzung der Ausgangsdaten eines effizienten Clusteringverfahrens vergleichbar und analog durch die Clusterfunktion verarbeitbar sein. Daher werden die diskreten Pläne der Ausgangsdaten für das Clustering im ersten Schritt auf das Intervall $[0, 1]$ normiert. Eine einheitliche Normierung der Pläne ist zulässig, da unabhängig von der Semantik des betrachteten Objektknotens im Model der Fertigung die Definition für den jeweiligen Plan gleich ist.¹⁶ Eine Normierung unter Berücksichtigung der einzelnen Planungsperioden $p(k)$

¹⁴Für Bedarfsänderungen ist dies ein nachgelagerter Objektknoten, für Angebotsänderungen hingegen ein vorgelagerter Objektknoten.

¹⁵Siehe Kap. 3.1.1.1, S. 49

¹⁶Siehe Definition 2.3, S. 14

5. Konzeption

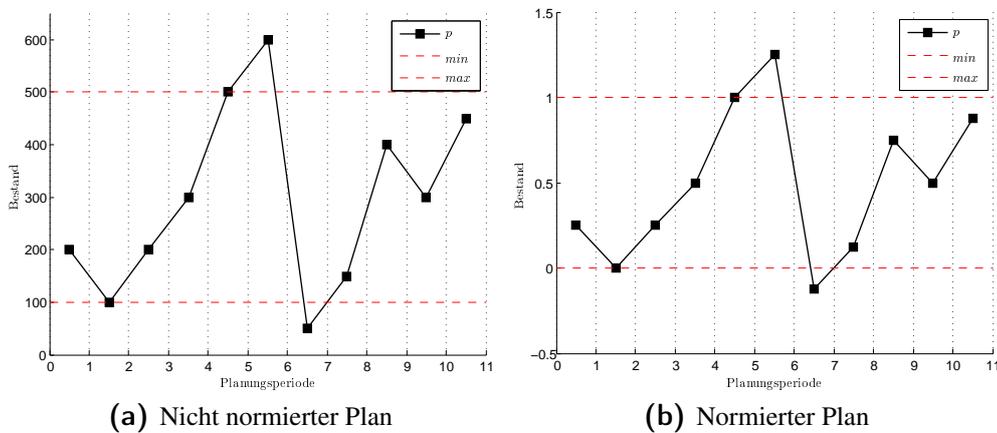


Abbildung 5.1.: Beispielhafte Darstellung einer Plannormierung

ist notwendig, da der Fall abgesenkter oder erhöhter Restriktionen einzelner Perioden $p(k)$ als spezifisches charakteristisches Merkmal des Planes eines Objektknotens in der Normierung berücksichtigt werden soll.

Diskrete Pläne werden im Clustering als Vektoren p mit Bestands- bzw. Kapazitätsangaben $p(k)$ für jedes Planungsintervall $k \in \{1, \dots, PH\}$ beschrieben.¹⁷ Der normierte eines diskreten Planes wird dabei durch einen Vektor $p^* \in \mathbb{R}^{PHZ}$ repräsentiert, der für einen Zeitpunkt $k \in \{1, \dots, PH\}$ genau einen Wert $p^*(k) \in [0, 1]$ aufweist, wenn sich der Bestand bzw. die Kapazität zu diesem Zeitpunkt innerhalb der vorgegebenen Restriktionsgrenzen des jeweiligen Objektknotens bewegt.

Die Normierung der Pläne kann an den Objektknoten dadurch erreicht werden, dass für jede Planungsperiode $k = 1, \dots, PH$ der Wert des ursprünglichen Planes $p^*(k)$ durch

$$p^*(k) = \frac{1}{p(k)_{max} - p(k)_{min}} (p(k) - p(k)_{min}) \quad (5.1)$$

ersetzt wird.¹⁸ Eine Restriktionsverletzung in Plan P zum Zeitpunkt k ist allgemein an $p^*(k) < 0$ oder $p^*(k) > 1$ zu erkennen. Abbildung 5.1 zeigt den Zusammenhang. Diese Normierung der Pläne wird sowohl für FOK wie auch für KOK durchgeführt und erlaubt eine einheitliche Betrachtung von Bestands- und Kapazitätsverläufen.

¹⁷Siehe Kap. 2.6, S. 32

¹⁸Die Normierung der Planwerte erfolgt unabhängig vom Objektknotentypen, da im Clustering alle Objektknoten aufgrund der analogen Definition von Plänen für FOK und KOK gleich behandelt werden können. (Vgl. Kap. 2.1.2, S. 17). Durch Verwendung normierter Pläne abstrahiert das hier vorgestellte Konzept den dedizierten Bezug beim Clustering zu einem spezifischen FOK oder KOK und kann so für beide Objektknotentypen angewandt werden.

5.1.2.2. Unterscheidungskriterien zur Zustandsabstraktion

Das Lernverfahren bewertet mithilfe einzelner Systemzustände die Ausführung möglicher Aktionen in der Änderungsplanung. Wenn mehrere Zustände zu einem Cluster zusammengefasst, und fortan als einzelner, abstrahierter Zustand betrachtet werden sollen, verliert diese Bewertung an Aussagekraft, falls nicht für alle Zustände des Clusters die Menge von zulässigen Aktionen übereinstimmt. Es muss also gelten

$$A(s_i) = A(s_j) \quad (\forall s_i, s_j \in S_k) \quad (5.2)$$

Andernfalls könnte das Lernverfahren eine Aktion für einen Zustand als zielführend lernen, die für diesen Zustand gar nicht anwendbar ist. Um derartige Situationen auszuschließen, muss es harte Unterscheidungsmerkmale zwischen den Clustern geben, die dafür sorgen, dass Zustände mit unterschiedlichen Mengen von ausführbaren Aktionen in unterschiedliche Cluster abgebildet werden.

In Kapitel 5.1.2.1 wurden die Art der Planänderung sowie der anfragende Objektknoten als Merkmale zur Festlegung anwendbarer Aktionen herausgestellt. Sie dienen im Rahmen der Clusterabbildung als *harte* Unterscheidungsmerkmale zwischen den Clustern. Die Anwendung harter Kriterien sichert ab, dass alle Zustände innerhalb eines Clusters aus einer gleichartigen Planänderung gleicher Objektknoten resultieren. Die Separation von Zuständen nach der Art der auslösenden Planänderung bewirkt, dass sich in den resultierenden Clustern nur Pläne mit gleichartiger Restriktionsverletzung befinden. Eine einzelne Planänderung kann in einem gültigen Plan eine Restriktionsverletzung hervorrufen. Diese sind Verletzungen der Minimal- oder Maximalrestriktionen eines Planes. Diese Einsicht ist wichtig, da sie die Herleitung der Cluster an mehreren Stellen vereinfacht.

Die Unterscheidung von Plänen durch die Art der auftretenden Planänderung und den anfragenden Objektknoten ermöglicht eine grobe Partitionierung des Zustandsraumes. In dieser Partitionierung werden die Zustände abstrahiert, auf denen gleiche Aktionen anwendbar sind. Diese Partitionierung ist zu grob für die Verwendung im Lernverfahren, da für die Auswahl einer auszuführenden Aktion zur Änderungsplanung insbesondere Informationen über den Plan am FOK bzw. KOK benötigt werden. Da aber aufgrund der diskutierten Komplexität¹⁹ nicht jeder mögliche Plan einzeln betrachtet werden kann, sollen genau diejenigen Zustände zusammengefasst werden, deren Pläne in charakteristischer Hinsicht „ähnlich“ anzusehen sind und die deshalb erwarten lassen, dass sie für gleiche Aktionen einen ähnlichen Reward erzielen. Die Ähnlichkeit von Zuständen in Clustern wird in dieser Arbeit neben der Angabe der Art der Planänderung und des anfragenden Objektknotens insbesondere durch den Planverlauf der jeweiligen Zustände ausgedrückt. Alle Pläne eines Clusters werden im Centroiden

¹⁹Siehe Kapitel 2.2.4

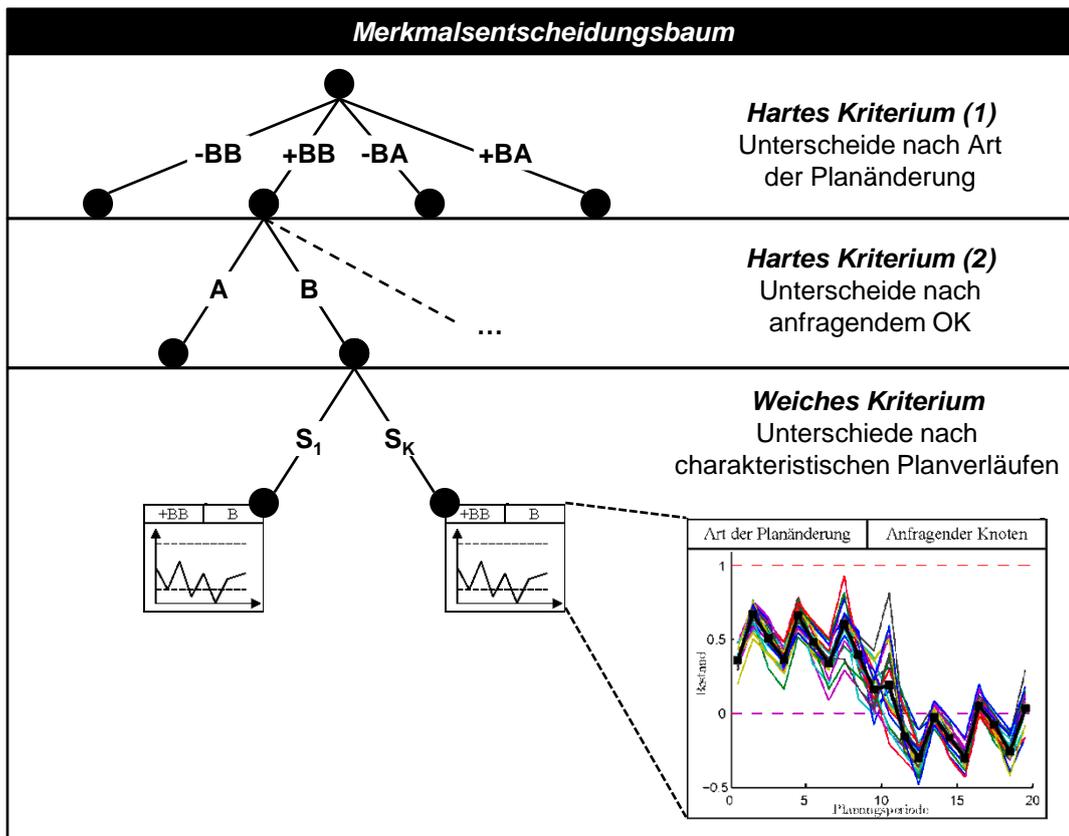


Abbildung 5.2.: Entscheidungsbaum zur Herleitung charakteristischer Pläne im Clustering

des Clusters zu einem charakteristischen Plan²⁰ für alle Zustände des Clusters abstrahiert.

Definition 5.1 (Charakteristischer Plan) *Ein charakteristischer Planverlauf repräsentiert die für das Lernverfahren relevanten Merkmale einer Menge ihm ähnlicher Pläne. Die einzelnen Pläne, die zu einem charakteristischen Plan abstrahiert werden, können im Lernverfahren gleich behandelt werden. Charakteristische Pläne werden durch eine Abstraktionsfunktion aus einer Menge von Plänen berechnet. Der Centroid eines Cluster repräsentiert einen charakteristischen Plan.*

Abbildung 5.2 verdeutlicht diese Herleitung der charakteristischen Pläne in Form eines Entscheidungsbaumes über die oben erarbeiteten relevanten Zustandsmerkmale: die harten und weichen Kriterien zur Zustandsbeschreibung. Auf oberster Ebene wird der Zustandsraum danach unterteilt, aus welcher Art von Planänderung die jeweiligen Zustände hervorgegangen sind. Diese Unterscheidung der insgesamt vier möglichen Typen von Planänderungen ist für alle Objektknoten im Netzwerk gleich. An

²⁰Siehe Kap. 2.2.2.2, S. 26

den Kanten der folgenden Ebene werden die Planänderungen danach unterschieden, von welchem vor- oder nachgelagerten Objektknoten eines PK die Anfrage zur Planänderung ausgeht. Die Art der Planänderung und die spezifischen Leistungsvereinbarungen bestimmen die bei der Koordination auftretenden charakteristischen Zustände des Produktionsnetzwerkes. Die Verzweigungen auf dieser Ebene des Baumes sind objektknotenspezifisch, da hier je eine abgehende Kante für jeden vor- oder nachgelagerten Objektknoten existieren muss.²¹ Nach diesen beiden Unterscheidungen ist für alle Zustände in den Objektknoten der dritten Ebene die gleiche Menge an Aktionen anwendbar. Bis zu dieser Ebene ist die Clusterabbildung nicht parametrisierbar und unterscheidet sich für unterschiedliche Objektknoten lediglich durch die variablen Mengen von vor- und nachgelagerten Objektknoten.

Im letzten Schritt folgt mit einer weiteren Unterteilung die Bildung der eigentlichen Zustandscluster, repräsentiert durch die Blätter des Entscheidungsbaumes. Jeder Zustand wird dem Cluster zugewiesen, dessen charakteristischer Plan dem des zuzuweisenden Zustandes am ähnlichsten ist. Die Menge der existierenden Cluster bzw. Blätter dieses Entscheidungsbaumes kann durch die Menge zu erzeugender charakteristischer Pläne bzw. Cluster vorgegeben werden.

Eine große Menge von Clustern erhöht die Wahrscheinlichkeit, dass im Clustering zu einem Zustand ein passender Cluster gefunden werden kann. Je niedriger die Anzahl der Cluster ausfällt, desto höher ist die Diversität der Pläne unter den Zuständen des Clusters. Dies ist ein Abwägungsproblem, da eine größere Menge von Clustern wiederum den Zustandsraum für das Lernverfahren vergrößert und zu erhöhtem Lern- und Rechenaufwand im Clustering selbst und ebenso im Lernverfahren führt.²²

5.1.2.3. Auswahl der charakteristischen Pläne

Zur vollständigen Beschreibung der Abstraktionsfunktion bleibt festzulegen, wie die Auswahl der charakteristischen Pläne erfolgt. Es wird untersucht, wie eine Menge von Plänen auf einen charakteristischen Plan reduzierbar ist und welche Merkmale der Pläne sich darin widerspiegeln müssen.

Ein Planer, der zu einem gegebenen Plan ein geeignetes Planungsverfahren auswählen soll, sucht nach bestimmten Mustern der Pläne²³, die Aufschluss über die Eignung zur Anwendung einzelner Änderungsplanungsverfahren geben. Dabei spielen bestimmte Merkmale in den Plänen eine deutlich größere Rolle, als der exakte Verlauf in jeder einzelnen Planungsperiode. Dieses Vorgehen des Planers wird bei der Zustandsabstraktion für die charakteristischen Pläne berücksichtigt. Ein solcher Plan als Repräsentant abstrahierter Zustände muss die charakteristischen Merkmale der Pläne aller

²¹Je nachdem, ob die zugehörige Art der Planänderung am Elternknoten angebots- oder bedarfseitig ist

²²Vgl. Kap. 5.1.1.1, S. 74

²³[Erl07]

5. Konzeption

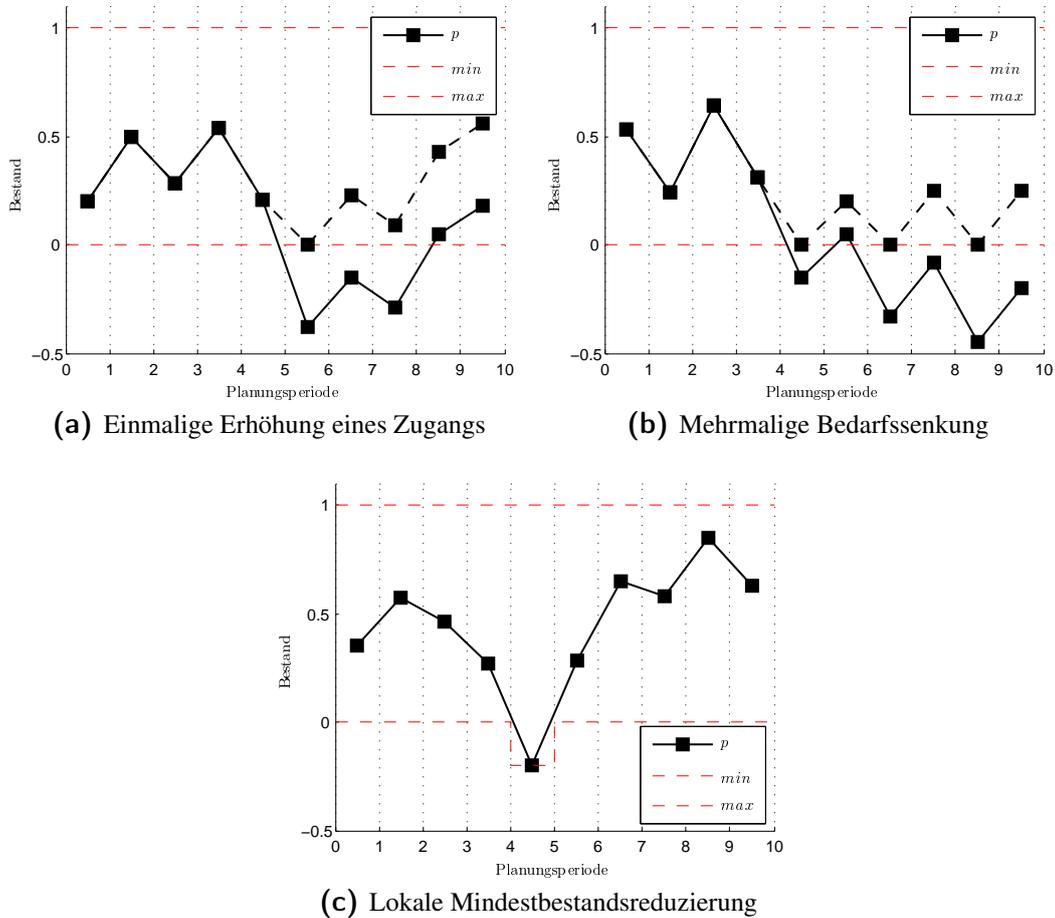


Abbildung 5.3.: Charakteristische Pläne und Planungsverfahren

zugehörigen Zustände wiedergeben. Anders formuliert müssen die Pläne aller Zustände eines abstrahierten Zustandes dieselben Planmuster aufweisen, die für die Auswahl eines Planungsverfahrens relevant sind.

Charakteristische Pläne beinhalten die für die Auswahl eines Planungsverfahrens wichtigsten Informationen. Diese beziehen sich insbesondere auf die Restriktionsverletzungen eines Planes, die sich durch die Zeitpunkte ihres Auftretens sowie ihre Höhe beschreiben lassen. Pläne können Muster aufweisen, die einen deutlichen Aufschluss über die geeigneten Planungsverfahren geben. Beispielsweise lassen sich zahlreiche aufeinanderfolgende Restriktionsverletzungen nur schwer durch lokale Aktionen wie die Änderung der Restriktionsgrenzen ausgleichen, sodass eine Weitergabe der geänderten Menge vorteilhafter ist. Vereinzelt Restriktionsverletzungen geringer Höhe lassen sich hingegen lokal ausgleichen.

Charakteristische Pläne können Aufschluss über die Art der Weitergabe von geänderten Bedarfen oder Angeboten geben. Einmalige oder in ihrer Höhe gleichbleibende

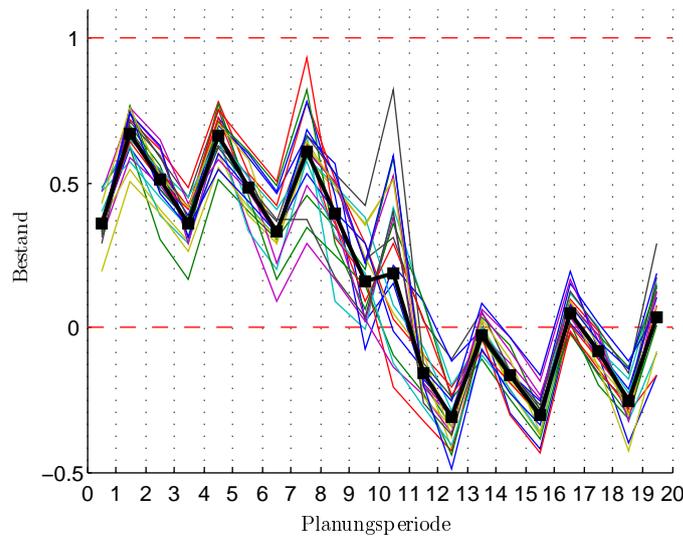


Abbildung 5.4.: Beispiel für ähnliche Pläne und deren charakteristischen Plan

Restriktionsverletzungen können z. B. durch die einmalige Änderung eines Zu- oder Abganges behoben werden, während periodisch auftretende und sich aufschaukelnde Restriktionsverletzungen eine Neuplanung der Losgrößen erfordern. Abbildung 5.3 zeigt einige Beispiele für Pläne mit charakteristischen Merkmalen, die möglichen Aufschluss über die Anwendung eines geeigneten Planungsverfahrens geben.

Der Vorteil dieser Art von Abstraktion ist, dass das Lernverfahren nicht mehr eine einzelne Planungsstrategie für jeden möglichen Plan lernt, sondern für ganze Klassen von Plänen. Zu den in Abbildung 5.3 dargestellten Plänen gibt es zahlreiche „ähnliche“ Verläufe, für die sich die gleichen Aussagen bezüglich der geeigneten Planungsverfahren treffen lassen. Durch die Zustandsabstraktion werden derartige Zustände zusammengefasst, sodass die Aussagen über diese Klassen von Zuständen verallgemeinert werden können. Dadurch wird nicht nur die Anzahl der zu verwaltenden Q-Werte im Lernverfahren reduziert, sondern eine Generalisierung des erlernten Wissens erreicht. Anstatt die Bewertungen von Planungsverfahren für alle Pläne unabhängig voneinander vorzunehmen, werden Bewertungen für alle Pläne vorgenommen, die gleiche Planmuster aufweisen.

Abbildung 5.4 zeigt beispielhaft, wie eine Menge von Plänen mit dem zugehörigen charakteristischen Plan aussehen kann. Anstatt für jeden einzelnen der als dünne Linie eingezeichneten Pläne die Eignung der verschiedenen Planungsverfahren im Lernverfahren zu bewerten, kann das Lernverfahren durch die Zustandsabstraktion die Bewertungen für alle Pläne, die dem mit dicker Linie eingezeichneten charakteristischen Plan hinreichend ähnlich sind, gleichzeitig vornehmen.

5.1.3. Erlernen charakteristischer Pläne mit k -means-Clustering

In diesem Kapitel wird das Zustandsabstraktionsverfahren dieser Arbeit detailliert konzipiert. Es wird die Methode k -means-Clustering angewendet. Mit dieser sollen unter Verwendung der harten und weichen Kriterien zur Zustandsabstraktion automatisch aus einer Menge von Trainingsdaten charakteristische Pläne erzeugt werden.²⁴

5.1.3.1. Anforderungen an die Distanzfunktion

Im Folgenden wird eine spezielle Distanzfunktion d für das vorgestellte k -means-Clustering definiert. Diese muss allen Anforderungen genügen, die an Distanzfunktionen gestellt werden²⁵:

1. $d(x, y) \geq 0$ und $d(x, y) = 0$ genau wenn $x = y$
2. Symmetrie: $d(x, y) = d(y, x)$
3. Dreiecksungleichung: $d(x, z) \leq d(x, y) + d(y, z)$

Neben diesen allgemeinen Merkmalen muss die zu definierende Distanzfunktion aber insbesondere die zusätzlichen problemspezifischen Anforderungen erfüllen:

- Die Verteilung der Restriktionsverletzungen eines Planes muss beachtet werden. Eine ähnliche Verteilung des zeitlichen Auftretens der Restriktionsverletzungen sollte eine verhältnismäßig große Ähnlichkeit (bzw. geringe Distanz) implizieren.
- Differenzen zwischen den absoluten Werten der Bestände bzw. Kapazitäten in Plänen sollten abhängig von ihrer Höhe bewertet werden. Unterschiede innerhalb der Restriktionsgrenzen sollten weniger stark gewichtet werden als Unterschiede, die zu Restriktionsverletzungen führen bzw. außerhalb dieser Grenzen liegen.

Die Distanzfunktion setzt sich aus der Kombination zweier Ähnlichkeitsmetriken zusammen. Die erste Metrik gibt die *strukturelle* Ähnlichkeit der Pläne wieder, die misst, inwiefern die Zeitpunkte der auftretenden Restriktionsverletzungen übereinstimmen.

²⁴Dabei muss beachtet werden, dass hier Clustering auf einer anderen Ebene diskutiert wird als bisher. Im Folgenden werden Pläne aus einer Trainingsmenge TR in Cluster C_1, \dots, C_k zusammengefasst. Diese Cluster enthalten eine Menge von Plänen und sind nicht mit den bisher diskutierten Zustandsclustern S_1, \dots, S_m zu verwechseln. Vielmehr dienen die Clustermittelpunkte von C_1, \dots, C_k als charakteristische Pläne, die zur Bildung der Zustandscluster zusätzlich mit Ausprägungen für die Merkmale *Art der Planänderung* und *anfragender Objektknoten* kombiniert werden.

²⁵Vgl. [Lar05], S. 99

Die zweite Metrik fasst hingegen die Differenzen in der Höhe der Bestände bzw. Kapazitäten *quantitativ* zusammen, wobei diese Differenzen zusätzlich danach gewichtet werden, ob einer der Pläne sich in der betrachteten Planungsperiode außerhalb seiner Restriktionen befindet.

5.1.3.2. Strukturelle Distanz

Liegt ein Plan mit Restriktionsverletzungen vor, so sind für die Auswahl einer adäquaten Reaktion insbesondere die Zeitpunkte relevant, zu denen die Restriktionsverletzungen auftreten. Pläne mit Restriktionsverletzungen nahe der Heutelinie können z. B. eine effiziente Änderungsplanung erschweren, da benachbarte Fertigungsstufen vorliegende Angebote oder Bedarfe nicht kurzfristig anpassen können.

Die Metrik für die strukturelle Distanz soll die Ähnlichkeit der zeitlichen Verteilung von Restriktionsverletzungen zweier Pläne wiedergeben. Es genügt hierfür, die Pläne auf binäre Vektoren zu reduzieren. In diesen Vektoren weisen Pläne mit einem Wert von 1 an der Stelle k in Planungsperiode $p^*(k)$ eine Restriktionsverletzung auf. Zu einem Plan P ist dieser binäre Vektor $\hat{P} \in \{0, 1\}^{PH}$ definiert durch:²⁶

$$\hat{P}(k) = \begin{cases} 1 & \text{falls } p^*(k) < 0 \vee p^*(k) > 1 \\ 0 & \text{sonst} \end{cases} \quad \text{für } k = 1, \dots, PH \quad (5.3)$$

Um die strukturelle Distanz zwischen zwei Plänen P_i und P_j zu bestimmen, sind nur die Elemente von \hat{P} relevant, die einen Wert von 1 aufweisen, d. h. die Planungsintervalle mit Restriktionsverletzungen. In der Literatur werden eine Anzahl von Metriken zur Messung der Ähnlichkeit zweier binärer Vektoren dokumentiert.²⁷ All diesen Metriken ist gemein, dass sie die Elemente der Vektoren paarweise vergleichen, d. h. auf Übereinstimmungen der Werte an der i -ten Stelle prüfen. Bei der Verteilung der Restriktionsverletzungen über den Planungshorizont ist es aber auch von Bedeutung, wie weit die Restriktionsverletzungen in den zwei Plänen voneinander entfernt sind.

In Abbildung 5.5(a) und 5.5(b) sind jeweils zwei binäre Vektoren dargestellt, aus denen die Zeitpunkte der Restriktionsverletzungen für zwei Pläne P_i und P_j zu entnehmen sind. Im Fall Abbildung 5.5(a) liegen diese Zeitpunkte weit auseinander: P_i weist eine Restriktionsverletzung in Periode $p(2)$ auf, P_j hingegen in Periode $p(6)$. Die Verteilung der Restriktionsverletzungen ist unterschiedlich und sollte für die Bewertung durch die Metrik eine große Distanz implizieren. Die Pläne in Abbildung 5.5(b) hingegen sind sich in dieser Hinsicht deutlich ähnlicher, da die Restriktionsverletzung in

²⁶An dieser Stelle sei angemerkt, dass die hier betrachteten Pläne nur gleichartige Restriktionsverletzungen aufweisen, sodass in \hat{P} nicht zwischen Unter- oder Überschreiten von Restriktionsgrenzen unterschieden werden muss (Vgl. Kap. 5.1.2.1, S. 79). Dieses ist kein Problem, da die Charakteristika der Zustände der Cluster widergespiegelt werden.

²⁷Eine Übersicht findet man z. B. in [CYT05]. Eine Diskussion findet sich in Kap. 3.1.3, S. 54 ff.

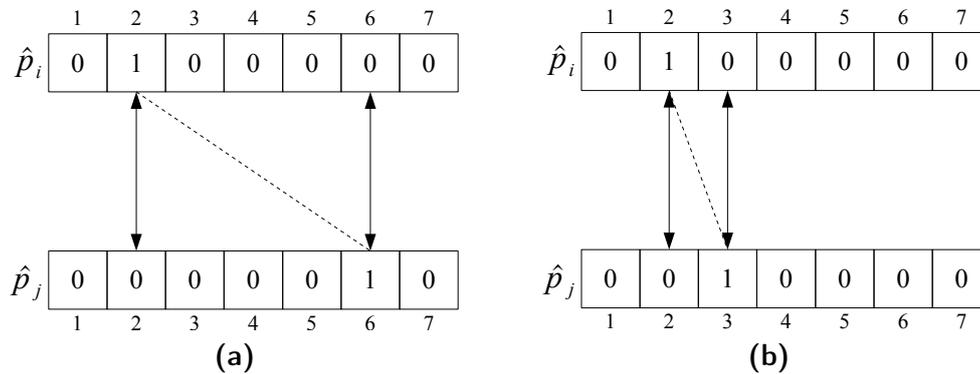


Abbildung 5.5.: Problematik bestehender Ähnlichkeitsmetriken für binäre Vektoren

Periode $p(3)$ auftritt und zeitlich nahe an der Restriktionsverletzung aus P_i , nämlich an $p(2)$, liegt.

Gängige Metriken für die Ähnlichkeit binärer Vektoren würden diesen beiden Beispielen die gleiche Distanz zuordnen, da sie alle auf einem paarweisen Vergleich der Elemente beruhen und lediglich die Anzahl der übereinstimmenden bzw. unterschiedlichen Elemente betrachten. Die Grundidee bei der Berechnung der strukturellen Distanz ist, Abstände zwischen den Zeitpunkten der Restriktionsverletzungen in P_i und P_j zu bewerten.

Im Folgenden bezeichnet $arv_P = \|\hat{P}\|$ die Anzahl der Restriktionsverletzungen in Plan P . O. B. d. A wird angenommen, dass P_i nicht weniger Restriktionsverletzungen aufweist als P_j , d. h. $arv_{P_i} \geq arv_{P_j}$. Es seien $RV_P = \{rv_1^P, \dots, rv_{arv_P}^P\}$ die Indizes der Planungsintervalle, in denen Plan P Restriktionsverletzungen aufweist, d. h. $\hat{p}(k) = 1$ für $k \in RV_P$. Algorithmus 5.1 stellt die Berechnung der strukturellen Distanz D_S als Pseudocode dar.

Zunächst werden die Spezialfälle betrachtet, in denen keiner der beiden Pläne oder nur P_i eine Restriktionsverletzung aufweist. Im ersten Fall ist die strukturelle Distanz als 0 definiert, da die Verteilung der Restriktionsverletzungen für beide Pläne identisch ist. Im zweiten Fall ist die strukturelle Distanz als 1 definiert, da nur einer der beiden Pläne überhaupt eine Restriktionsverletzung aufweist. Handlungsbedarf besteht bei einem der Pläne. Für den Normalfall, dass beide Pläne Restriktionsverletzungen aufweisen, wird für jede Restriktionsverletzung $rv_k^{P_i}$ in P_i diejenige Restriktionsverletzung in P_j ermittelt, die zeitlich am nächsten an $rv_k^{P_i}$ liegt. Dieser zeitliche Abstand wird noch ins Verhältnis zur Länge des Planungshorizontes gesetzt, um eine relative Bewertung zu erreichen. Diese Berechnung wird in Zeile 10 für jede Restriktionsverletzung in P_i durchgeführt. Die strukturelle Distanz berechnet sich schließlich als Mittelwert all dieser Distanzen ²⁸. Diese Berechnung impliziert, dass Pläne mit einer identischen

²⁸Siehe Algorithmus 5.1 Zeile 12

Algorithmus 5.1 : Berechnung der strukturellen Distanz

Eingabe : Indizes der Restriktionsverletzungen RV_{P_i}, RV_{P_j}

Ausgabe : $D_S(P_i, P_j)$ Strukturelle Distanz zwischen P_i und P_j

```

1 if  $arv_{P_i} = arv_{P_j} = 0$  then
2    $D_S = 0$ 
3   return  $D_S$ 
4 end
5 if  $arv_{P_i} > 0 \wedge arv_{P_j} = 0$  then
6    $D_S = 1$ 
7   return  $D_S$ 
8 end
9 for  $k = 1, \dots, arv_{P_i}$  do
10   $dist(k) \leftarrow \frac{1}{PH} \min_{rv_{P_j}} (|rv_k^{P_i} - rv^{P_j}|)$ 
11 end
12  $D_S = \frac{1}{arv_{P_i}} \sum dist(k)$ 
13 return  $D_S$ 

```

Verteilung der Restriktionsverletzungen eine strukturelle Distanz von 0 aufweisen, da für jede Restriktionsverletzung aus P_i der Abstand zur zeitlich nächsten Restriktionsverletzung in P_j 0 beträgt. Nach oben ist der Wert für die strukturelle Distanz durch 1 begrenzt. Dies wird deutlich, wenn man den Grenzfall betrachtet, bei dem jeweils eine Restriktionsverletzung in P_i und P_j auftritt, die an den unterschiedlichen Extremen, d. h. im ersten und letzten Planungsintervall, liegt. In diesem Fall nähert sich die strukturelle Distanz mit wachsendem Planungshorizont dem Wert 1 beliebig nahe an. Abbildung 5.6 veranschaulicht diese Berechnung anhand von zwei Plänen.

Aus den in Abbildung 5.6(a) dargestellten Plänen ergeben sich für die Zeitpunkte der

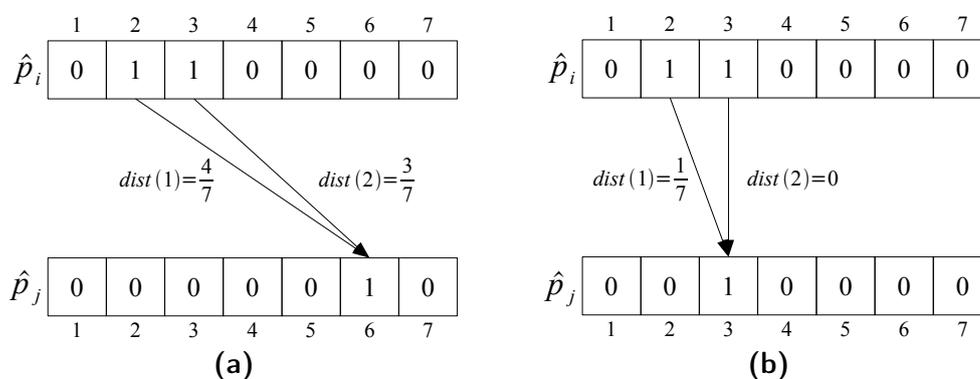


Abbildung 5.6.: Beispiel zur Berechnung der strukturellen Distanz

5. Konzeption

Restriktionsverletzungen die Werte $RV_{P_i} = \{2, 3\}$ und $RV_{P_j} = \{6\}$. Es gilt $arv_{P_i} = 2$ und $arv_{P_j} = 1$. Die Distanz der Restriktionsverletzung in $p_i(2)$ zur zeitlich nächsten Restriktionsverletzung in $p_j(6)$ beträgt vier Planungsperioden, die von $p_i(3)$ entsprechend 3 Planungsperioden. Diese Distanzen werden ins Verhältnis zum Planungshorizont von 7 gesetzt, sodass man die Werte $dist(1) = \frac{4}{7}$ und $dist(2) = \frac{3}{7}$ erhält. Die strukturelle Distanz $D_S(P_i, P_j)$ als deren Mittelwert beträgt also

$$\frac{1}{2} \left(\frac{4}{7} + \frac{3}{7} \right) \approx 0,51.$$

Analog ergibt sich für das Beispiel in Abbildung 5.6(b) eine strukturelle Distanz von

$$\frac{1}{2} \left(\frac{1}{7} + 0 \right) \approx 0,07.$$

Dieser Unterschied der Ergebnisse bei abweichenden Restriktionsverletzungen verdeutlicht die Bewertungssystematik der Metrik zur Messung der strukturellen Distanz zweier Pläne.

5.1.3.3. Quantitative Distanz

Neben der Verteilung der Restriktionsverletzungen stellen die absoluten Differenzen der Planwerte zwischen den Plänen die zweite Komponente der Distanzfunktion dar. Die Berechnung erfolgt durch den Mittelwert der Differenzen der Bestände bzw. Kapazitäten in den einzelnen Planungsperioden. Dabei ist es sinnvoll, Differenzen in Perioden mit Restriktionsverletzungen stärker zu gewichten als diejenigen, bei denen beide Pläne innerhalb der Restriktionsgrenzen verlaufen. Die Ausprägung einer Restriktionsverletzung hat bei ungültigen Zuständen einen stärkeren Einfluss auf die Auswahl eines geeigneten Änderungsplanungsverfahrens, als bei Plänen mit Differenzen der Periodenbelegungen innerhalb der Restriktionsgrenzen. Im Rahmen dieser Gewichtung werden Differenzen in Planungsintervallen, in denen mindestens einer der Pläne eine Restriktionsverletzung aufweist, mit dem durch den Parameter w_{RV}^d vorgegebenen Wert gewichtet.

Die maximale Differenz pro Planungsperiode muss auf 1 begrenzt werden, um den Wertebereich für die Distanzmessung auf das Intervall $[0, 1]$ zu beschränken. Dies ist praktisch betrachtet keine große Einschränkung, da für realistische w_{RV}^d die gewichtete Differenz nur selten einen Wert von 1 übersteigt. Algorithmus 5.2 stellt die Berechnung dieser quantitativen Distanz D_Q als Pseudocode dar.

5.1.3.4. Kombinierte Distanzfunktion und Beispiel

Die beiden Komponenten der strukturellen und quantitativen Distanz müssen so kombiniert werden, dass aus ihnen ein einzelner Wert für die Distanz zwischen zwei Plänen

Algorithmus 5.2 : Quantitative Distanz

Eingabe : Pläne P_i, P_j

Ausgabe : $D_Q(P_i, P_j)$ Quantitative Distanz zwischen P_i und P_j

```

1 for  $k = 1, \dots, PH$  do
2    $diff(k) = |p_i^*(k) - p_j^*(k)|$ 
3   if  $\hat{P}_i(k) + \hat{P}_j(k) \geq 1$  then
4      $diff(k) = \max(diff(k) \cdot w_{RV}^d, 1)$ 
5   end
6 end
7  $D_Q = \frac{1}{PH} \sum diff(k)$ 

```

errechnet werden kann. Da die beiden einzelnen Distanzwerte auf das Intervall $[0, 1]$ normiert sind, lässt sich die gemeinsame Distanzfunktion als gewichtete Summe dieser beiden Komponenten darstellen. Diese Gewichtung kann vom Benutzer über die Parameter w_S und w_Q festgelegt werden, wobei $w_S + w_Q = 1$ gelten muss, um den Wertebereich für die Distanzfunktion auf das Intervall $[0, 1]$ zu begrenzen. Die Distanzfunktion $d : \mathbb{R}^{PHZ} \times \mathbb{R}^{PHZ} \mapsto \mathbb{R}$ ist definiert als

$$d(P_i, P_j) = w_S D_S(P_i, P_j) + w_Q D_Q(P_i, P_j) \quad (5.4)$$

Die Wahl für die Gewichtungen w_S und w_Q beeinflusst maßgeblich die resultierenden Cluster. Je größer w_S gewählt wird, desto stärker ähneln sich die Pläne innerhalb eines Clusters im Hinblick auf die Verteilung der Restriktionsverletzungen über den Planungshorizont, während Abweichungen der Pläne in der Höhe eher toleriert werden. Eine Erhöhung von w_Q führt zu Abweichungen, in denen die Verteilung der Restriktionsverletzungen eher toleriert wird, aber Abweichungen in der Höhe der Pläne stärker bestraft werden.

In Abbildung 5.7 sind zwei Pläne P_i und P_j dargestellt, für die beispielhaft die Distanz $d(P_i, P_j)$ berechnet werden soll. Der erste Plan P_i weist insgesamt 3 Restriktionsverletzungen in den Planungsperioden $p_i(3, 7, 9)$ auf:

$$RV_{P_i} = \{3, 7, 9\} \quad (5.5)$$

Der unten dargestellte Plan P_j zeigt Restriktionsverletzungen hingegen in den Perioden $p_j(2)$ und $p_j(7)$:

$$RV_{P_j} = \{2, 7\} \quad (5.6)$$

Aus den eingezeichneten Werten für die einzelnen Distanzen $dist(1, 2, 3)$ ergibt sich eine strukturelle Distanz von $\frac{1}{3}(\frac{1}{10} + 0 + \frac{1}{5}) = 0,1$. Die quantitative Distanz errechnet sich aus den gewichteten Differenzen zwischen den Plänen, summiert über alle Planungsperioden.

Die Gewichtung ergibt sich aus der Unterscheidung, ob in der betrachteten Planungsperiode einer der Pläne eine Restriktionsverletzung offenbart. Ist dies der Fall, so wird

5. Konzeption

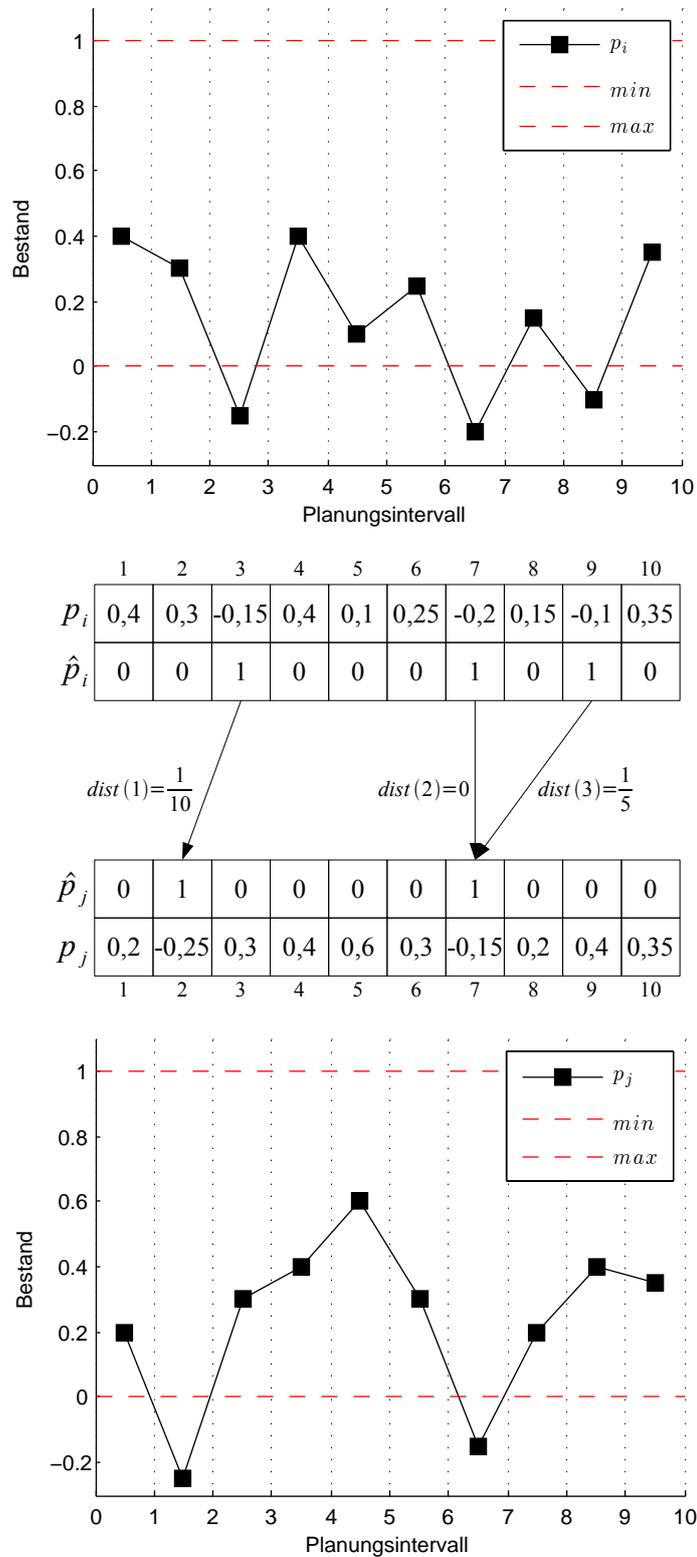


Abbildung 5.7.: Beispiel zur Berechnung der Gesamtdistanzfunktion

Tabelle 5.1.: Distanzfunktion für verschiedene w_S und w_Q

$w_S = 0.7$	$w_S = 0.5$	$w_S = 0.3$
$w_Q = 0.3$	$w_Q = 0.5$	$w_Q = 0.7$
0.1536	0.1893	0.2250

die Differenz mit dem Faktor w_{RV}^d multipliziert, ansonsten einfach gewichtet. Eine Gewichtung erfolgt für dieses Beispiel also in den Perioden 2,3,7 und 9. Es folgt für $w_{RV}^d = 2$ der Wert

$$D_Q = \frac{1}{10}(0,2 + 2 \cdot 0,55 + 2 \cdot 0,45 + 0 + 0,5 + 0,05 + 2 \cdot 0,05 + 0,05 + 2 \cdot 0,5 + 0) = 0,2786.$$

Aus der gewichteten Summe von D_S und D_Q errechnet sich schließlich die Distanz $d(P_i, P_j)$. Tabelle 5.1 zeigt die resultierende Distanz der Pläne für $w_{RV}^d = 2$ für verschiedene Gewichte w_S und w_Q .

5.1.3.5. Einfluss der Gewichtungsfaktoren

Eine höhere Gewichtung der strukturellen Distanz weist Clustern eher Pläne mit gleich verteilten Restriktionsverletzungen zu. Dieses könnte für Unternehmen relevant sein, deren Beschaffungsprozesse z. B. mit dem Bestellpunktverfahren gesteuert werden, da dort eher ungleichförmige Restriktionsverletzungen auftreten, die so zu problemspezifischen charakteristischen Plänen abstrahiert werden können.

Die verstärkte Gewichtung der quantitativen Distanz führt im Clustering eher Pläne mit ähnlichen Planverlaufswerten zu charakteristischen Plänen zusammen. Dieses kann für Unternehmen relevant sein, deren Beschaffungsprozesse bei interner Lagerfertigung mit Sicherheitsbeständen z. B. mit dem Bestellzyklusverfahren gesteuert werden. In diesem Fall kann zur Steuerung der Änderungsplanung beispielsweise die Betrachtung der mittleren Summe der quantitativen Abweichungen von Beständen interessant sein und nicht primär die Anzahl der Restriktionsverletzungen der Pläne.

5.1.3.6. Aktualisierung der Clustermittelpunkte

Am Ende jeder Iteration des k -means-Algorithmus müssen die Clustermittelpunkte neu bestimmt werden, um den charakteristischen Plan je Cluster zu erhalten. Üblicherweise wird dieser Mittelpunkt aus den Mittelwerten der Vektorelemente aller Clustermittglieder berechnet. Dabei werden implizit die Distanzen aller Clustermittglieder

zum Centroid minimiert. Vor dem Hintergrund der in Kapitel 5.1.3 vorgestellten Distanzfunktionen tritt dieser Effekt in diesem Fall nicht ein, da eine einfache Mittelwertbildung nicht zu charakteristischen Plänen führt, die typische Restriktionsverletzungen der zugeordneten Clusterpläne aufweisen. Charakteristische Pläne, die durch periodenbasierte Mittelung der normalisierten Clusterpläne berechnet werden, nehmen wegen der glättenden Wirkung der Mittelwertbildung selten Extremwerte an. Beim Clustering werden vorherige Extremwerte abgeflacht, wodurch der charakteristische Plan nicht mehr zwingend die charakteristischen Restriktionsverletzungen seiner zugeordneten Pläne repräsentiert. Solche charakteristischen Pläne sind zur Verwendung als Ausgangsdaten für das Lernverfahren nicht geeignet.

Idealerweise müsste der Clustermittelpunkt so gewählt werden, dass die Summe der Distanzfunktionswerte über alle Clustermittglieder minimiert wird. Dieses stellt ein nicht-lineares Optimierungsproblem²⁹ dar, dessen Lösung zwar möglich ist, aber mehr Rechenaufwand mit sich bringt als ein in linearer Zeit zu berechnender einfacher Mittelwert. Da die Berechnung der Centroiden während des Clusterings wiederholt durchgeführt werden muss, soll dieser zusätzliche Berechnungsaufwand im Sinne eines effizienten Verfahrens vermieden werden. Deshalb wird der Ansatz verfolgt, eine Näherung dieses optimalen Mittelpunkts zu erreichen, indem Werte, die sich außerhalb der Restriktionsgrenzen befinden, in der Berechnung des Mittelwertes deutlich stärker gewichtet werden.

Diese Gewichtung wird über den Parameter $w_{RV}^c \geq 1$ vorgenommen. Dieser Parameter hat Einfluss darauf, wie ausgeprägt die Restriktionsverletzungen in den charakteristischen Plänen ausfallen. Allgemein resultieren aus kleinen Werten für w_{RV}^c „gemäßigte“ charakteristische Pläne mit geringeren Schwankungen, während für größere Werte die Restriktionsverletzungen deutlich ausgeprägter sind. Formal berechnet sich der Plan des Clustermittelpunktes zu jeder Periode $p^*(k)$ als:

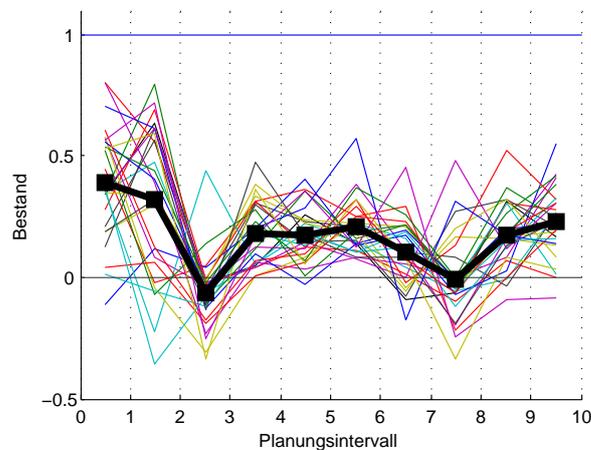
$$c_i(k) = \frac{1}{|C_i| + (w_{RV}^c - 1) \sum_{C_i} \hat{p}(k)} \sum_{C_i} p^*(k) \cdot \max(1, w_{RV}^c \hat{p}(k)) \quad (5.7)$$

Im Nenner des Bruchs wird zu der Anzahl der Pläne im Cluster C_i noch die Anzahl der Elemente mit Restriktionsverletzungen zum Zeitpunkt k , multipliziert mit $w_{RV}^c - 1$, addiert.³⁰ Dies ist erforderlich, um eine korrekte Berechnung des Mittelwertes zu erreichen. In der Summe werden alle Elemente des Plans entweder mit 1 für $\hat{p}(k) = 0$ oder mit w_{RV}^c für $\hat{p}(k) = 1$ gewichtet.

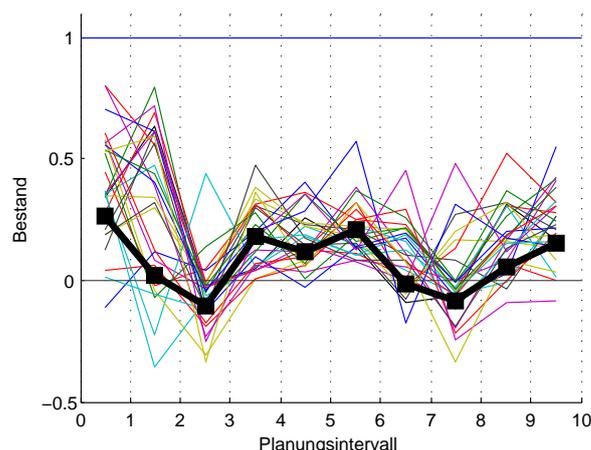
Abbildung 5.8 zeigt jeweils einen Cluster von Plänen, zusammen mit dem zugehörigen Centroid. In Abbildung 5.8(a) ist die Berechnung mit $w_{RV}^c = 1$ vorgenommen worden, d. h. Pläne außerhalb der Restriktionsgrenzen werden nicht gesondert gewichtet. Der

²⁹Z. B. [SM06]

³⁰Die Multiplikation mit $(w_{RV}^c - 1)$ resultiert aus der Tatsache, dass auch alle Elemente des Clusters mit Restriktionsverletzungen zum Zeitpunkt k einmal in $|C_i|$ enthalten sind.



(a)



(b)

Abbildung 5.8.: Aktualisierung der Clustermittelpunkte für unterschiedliche w_{RV}^c

in Abbildung 5.8(b) dargestellte Clustermittelpunkt wurde hingegen mit $w_{RV}^c = 10$ berechnet. Wie zu sehen ist, werden die charakteristischen Restriktionsverletzungen der Pläne des Clusters in Abbildung 5.8(b) durch den Clustermittelpunkt deutlicher repräsentiert. Das Beispiel zeigt die sensible Steuerung des Clusterverfahrens durch die Gewichte. Eine geringe Erhöhung der Gewichtung führt nur zu geringen Veränderungen des Centroids.

5.1.3.7. Auswahl der initialen Clustermittelpunkte

Bevor die erste Iteration mit der Zuweisung der Trainingsdaten zu Clustern beginnen kann, muss der k -means-Algorithmus n initiale Clustermittelpunkte festlegen. Die Auswahl der initialen Centroiden beeinflusst die Geschwindigkeit der Konvergenz des Algorithmus. Ein verbreitetes Vorgehen ist es, n zufällig gewählte Elemente der Ausgangsdaten als initiale Clustermittelpunkte festzulegen. Dies hat den Vorteil, dass kein Cluster zu Beginn leer bleibt, da jeder Mittelpunkt mit einem der Trainingsdatenpunkte identisch ist, zu ihm eine Distanz von Null aufweist und dem Cluster sicher zugeordnet werden kann. Dieses Vorgehen kann in der Konsequenz jedoch zu Clustern führen, die nur einen Plan, nämlich den Initialen, enthalten, weil keine weiteren ähnlichen Trainingsdaten vorhanden sind. Diese Pläne sind kein Repräsentant eines charakteristischen Planes im Sinne der Problemstellung. Dieser Effekt muss vermieden werden, damit die Verwendung solcher pseudocharakteristischer Pläne im Lernverfahren ausgeschlossen ist.

Aus diesem Grund werden im vorgestellten Verfahren als initiale Clustermittelpunkte keine Elemente aus den Trainingsdaten, sondern zufällig erzeugte Pläne gewählt. Die charakteristischen Pläne aller leeren Cluster werden nach einer Iteration neu initialisiert, um zu vermeiden, dass Cluster dauerhaft leer bleiben. Durch dieses zufallsbasierte Verfahren und die ständige Neuinitialisierung der leeren Cluster soll gewährleistet werden, dass nach Beendigung des Clusterprozesses alle Mittelpunkte nur Planmuster abbilden, die in einer hinreichend großen Menge der Trainingsdaten vorhanden sind.³¹

5.1.3.8. Trainingsdaten für das Clustering

Das Clustering muss für jeden Objektknoten durchgeführt werden. Aus der Definition des Untersuchungsgegenstandes geht hervor, dass die einzelnen Objektknoten eines Produktionsnetzwerkes unter Verwendung unterschiedlicher Strategien beplant werden. Diese hängt von der initialen Konfiguration der einzelnen Objektknoten ab.

Beispielsweise könnte die Motorenfertigung eines Automobilherstellers in den einzelnen Fertigungsstufen, repräsentiert durch einzelne Objektknoten, unterschiedlich konfiguriert sein und so unterschiedliche Ausgangsdaten je Objektknoten aufweisen.³² So hat etwa eine FST-Rumpfmontage eine variable Durchlaufzeit und eine FST-Versand eine fixe Durchlaufzeit. Bei einer zusätzlichen Unterscheidung durch die Verwendung von Bestellpunkt- und Bestellzyklusverfahren würden die charakteristischen Pläne dieser beiden Fertigungsstufen unterschiedliche Ausprägungen besitzen.

³¹Tabelle B.1 im Anhang führt die Parameter zur Konfiguration des Generators für initiale Belegungen der Cluster auf.

³²Siehe Beispiel in Kap. 2.1.1.

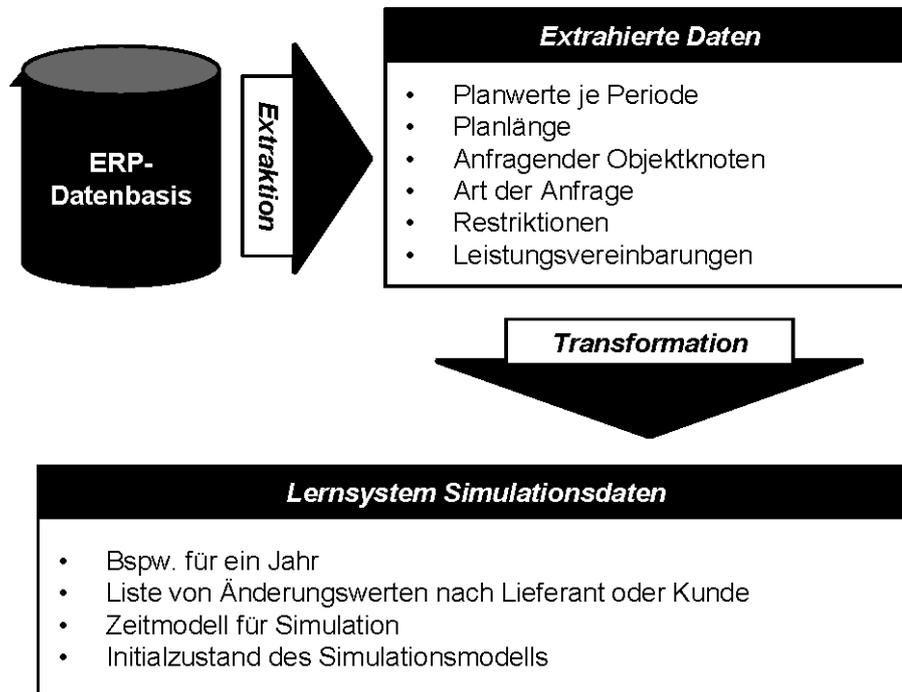


Abbildung 5.9.: Extraktion und Art der realen Ausgangsdaten für eine Lernepisode

Beim Bestellpunktverfahren könnten z. B. Schwankungen im Bestand der FOK oder Kapazitätsauslastung im KOK, verteilt über den Plan, auftreten und wären abhängig vom Bestellverhalten des Kunden. Bei einem Bestellzyklusverfahren würden die Schwankungen an den Bestellpunkten auftreten. Eine variable Durchlaufzeit führt, wiederum abhängig vom Kundenverhalten, zu weniger Spitzen im Bestand und in der Kapazitätsauslastung, während feste Durchlaufzeiten mehr Spitzen hervorrufen können. Das Verhältnis der Restriktionsgrenzen des Planes zu den Werten einer Planungsperiode ist ein weiteres entscheidendes Kriterium zur Bewertung der strukturellen und quantitativen Ähnlichkeiten der Merkmale der zu vergleichenden Pläne.

Um für ein solches Produktionsnetzwerk charakteristische Pläne erzeugen zu können, müssen für das Clustering entsprechende Ausgangsdaten aus ERP-Systemen bereitgestellt werden. Entweder liegen die erforderlichen Daten z. B. als Bestandsverläufe direkt in den ERP-Systemen vor, oder sie müssen aus den vorhandenen Daten des ERP-Systems extrahiert werden. Z. B. können sie als diskrete Mengen in Verbindung mit der jeweiligen Planungsperiode unter Verrechnung etwaiger Vorlaufzeiten gespeichert und so den betreffenden Fertigungsstufen zugeordnet werden. In ERP-Systemen wie z. B. SAP³³ werden in der Regel alle Transaktionen mit der Datenbank, wie z. B. das Buchen eines Planwertes, mit Buchungsbelegen im System dokumentiert. Diese Belege können verwendet werden, wenn die erforderlichen Daten nicht vorliegen. So

³³Siehe <http://www.sap.com>

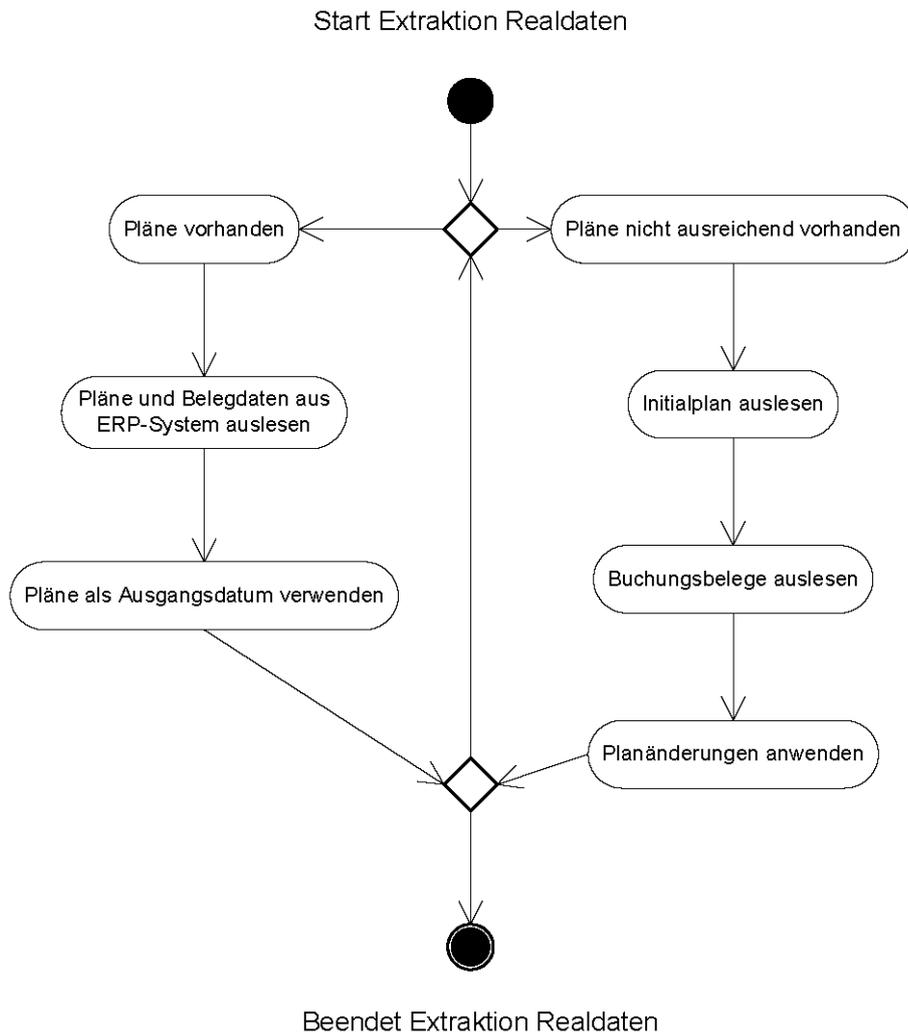


Abbildung 5.10.: Aktivität: Extraktion Realdaten

werden z. B. alte Pläne bzw. Änderungsanfragen nicht zwingend in einem Änderungsprotokoll gespeichert. Durch die Verwendung von Buchungsbelegen ist es möglich, historische Pläne mit Änderungsanfragen und deren Verursachern durch eine Planbestandsrechnung zu verknüpfen und so ursprüngliche Pläne für die Ausgangsdaten zu rekonstruieren. Abbildung 5.10 illustriert dieses.

Zum Clustering³⁴ werden die in Abbildung 5.11 dargestellten Ausgangsdaten benötigt. Die Schnittstelle kann variabel gehalten werden, da sie für spezifische ERP-Systeme angepasst umgesetzt werden muss.³⁵ Die einzelnen Datenelemente der Ausgangsdaten besitzen die in Tabelle 5.2 dargestellten Attribute.

³⁴Dieses wird auch für das Lernverfahren benötigt.

³⁵In dieser Arbeit wurde in der Umsetzung eine XML-Schnittstelle verwendet.

Tabelle 5.2.: Attribute der Ausgangsdaten

Attribut	Beschreibung	Quelle ERP
PlanwertJePeriode	Beinhaltet für jede Periode des Planungshorizontes einen Planwert	Archivdaten, Buchungsbelege
Planlänge	Bezeichnet die Anzahl der zur Verfügung stehenden Planungsperioden <i>PHZ</i>	Stammdaten
AnfrageVonObjekt	Bezeichnet den Partner und die bestätigte Höhe einer durchgeführten Anfrage eines Initiators an eines Partizipanten	Archivdaten, Buchungsbelege
ArtDerAnfrage	Bedarfs- oder Angebotsänderung	Buchungsbelege, Netzwerkmodell
StdMaxRestriktion	Maximalrestriktion in einem Objektknoten	Stammdaten
StdMinRestriktion	Minimalrestriktion in einem Objektknoten	Stammdaten
MaxRestriktionJePeriode	Angepasste Maximalrestriktion in einem Objektknoten nach erfolgter Änderungsplanung je Periode	Archivdaten, Buchungsbelege
MinRestriktionJePeriode	Angepasste Minimalrestriktion in einem Objektknoten nach erfolgter Änderungsplanung je Periode	Archivdaten, Buchungsbelege
MaxLeistungsvereinbarungen	Maximalrestriktion für Leistungsvereinbarungen zwischen zwei Objektknoten	Archivdaten, Buchungsbelege
MinLeistungsvereinbarungen	Minimalrestriktion für Leistungsvereinbarungen zwischen zwei Objektknoten	Archivdaten, Buchungsbelege

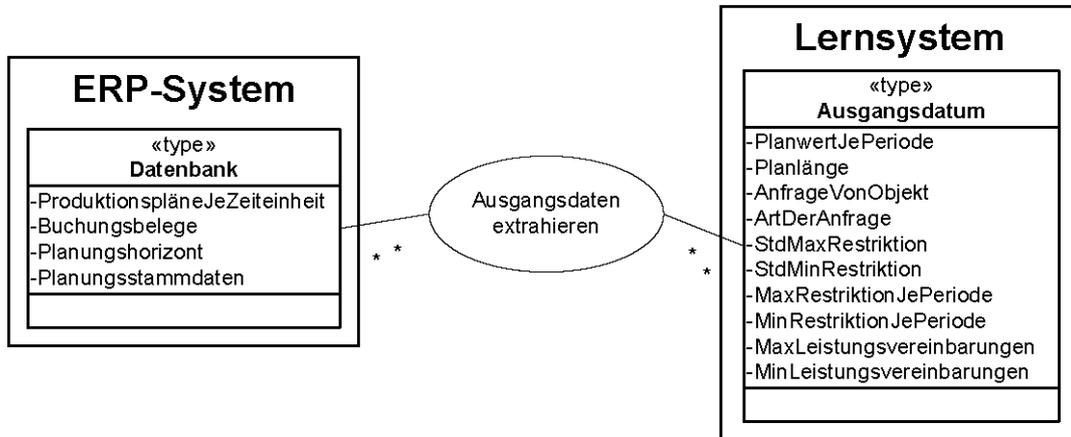


Abbildung 5.11.: Merkmale der Ausgangsdaten

Ein Problem beim Lernen auf realen Daten besteht darin, dass alle Unternehmen im Produktionsnetzwerk die erforderlichen Daten bereitstellen müssen. Handelt es sich bei dem Produktionsnetzwerk um das Produktionssystem eines Unternehmensverbundes oder eines Werkes, so können die Ausgangsdaten in der Regel aus einem zentralen Server ausgeleitet werden. Andernfalls müssen entsprechende Vereinbarungen über die Bereitstellung von Daten über kollaborative Ansätze im Supply Chain Management festgeschrieben werden.³⁶

5.1.3.9. Terminierung des Clusterverfahrens

Der k -means-Algorithmus konvergiert, wenn in einer Iteration alle Zuweisungen von Elementen der Trainingsmenge zu ihren jeweiligen Clustern unverändert bleiben. In diesem Fall bleiben die Clustermittelpunkte konstant, da sie über die gleichen Clusterelemente berechnet werden wie in der vorhergehenden Iteration. Da der Algorithmus keine Garantie für die Konvergenz des Verfahrens bietet, ist es sinnvoll, weitere Abbruchkriterien zu definieren.

Ein einfaches Abbruchkriterium ist die Vorgabe einer maximalen Anzahl von Iterationen. Hat der Algorithmus diese Anzahl von Iterationen erreicht, bricht dieser ab und liefert das zuletzt durchgeführte Clustering der Trainingsdaten als Ergebnis. Alternativ kann die Forderung, dass sich die Zuweisung von *keinem* der Elemente ändern darf, relaxiert werden. Dieses geschieht durch die Angabe eines prozentualen Schwellwertes T , für den sich die Elemente der Trainingsdaten nach einer Zuweisung zum Cluster nicht mehr ändern. Bei der Kombination der Kriterien sollte die maximale Anzahl der Iterationen im Clustering so gewählt werden, dass eine mögliche Terminierung des

³⁶Konzepte hierzu werden zum Beispiel in AC/DC untersucht. Siehe EU-Projekt AC/DC (FP6 Integrated Project, Fördernummer 031520, Laufzeit 10/2006- 09/2010) oder z. B. [DDKT07].

Verfahrens über das Konvergenzkriterium in der Regel nicht vorzeitig verhindert wird. Tritt dieses nicht ein, so verhindern die definierten Abbruchkriterien eine unendliche Iteration des Clusterings.

5.1.4. Zusammenfassung

Zur Abstraktion des Zustandsraumes wurde eine problemspezifische Abstraktionsfunktion zur Verwendung im k -means-Clusteringverfahren eingeführt. Zusätzlich zu den grob partitionierenden Zustandsmerkmalen:

- Art der Planänderung und
- anfragender Objektknoten

berücksichtigt diese bei der Zustandsabstraktion die strukturellen und quantitativen Merkmale der originären Planzustände.

Als strukturelle Merkmale werden hier die Restriktionsverletzungen der Pläne verwendet. Die eingeführte strukturelle Distanzfunktion bewertet diese hinsichtlich ihres Auftretens im Plan. Die quantitativen Merkmale bewerten über die quantitative Distanzfunktion periodenweise die Differenz von Plänen. Durch einen Gewichtungsfaktor können Perioden mit Restriktionsverletzungen bei der Bewertung besonders hervorgehoben werden. Die Gesamtdistanz zweier Pläne errechnet sich aus der gewichteten Summe der strukturellen und quantitativen Distanz zweier Pläne. Die Zuordnung zu einem Cluster erfolgt über die Distanz zwischen einem zuzuweisenden Plan und den Centroiden der zur Verfügung stehenden Cluster.

Die Berechnung des Centroiden wird nach erfolgter Iteration des Clusterings durch eine Heuristik durchgeführt. Diese errechnet durch die periodenweise Mittelwertbildung der im Cluster zugeordneten Pläne und deren Belegungen den charakteristischen Plan. Um charakteristische Merkmale der originären Pläne zu erhalten, kann der berechnete Mittelwert durch einen Gewichtungsfaktor akzentuiert werden. Das Ziel dieser Heuristik ist einerseits, die Centroiden effizient berechnen zu können, und andererseits, die charakteristischen Merkmale des Centroiden im Clustering zu unterstreichen.

Die Erzeugung von Ausgangsdaten für das Clustering wurde analysiert und eine generisch angelegte Schnittstelle zur Extraktion von Realdaten aus ERP-Systemen vorgeschlagen.

5.2. Konzeption der Lernfunktion für das Q-Learning

In diesem Kapitel wird die Lernfunktion für das Lernverfahren konzipiert. Diese wird im Sinne des Q-Learnings im Weiteren als *Rewardfunktion* bezeichnet. Aufgabe der Rewardfunktion ist die problemspezifische und automatisierte Analyse durchgeführter Änderungsplanungsprozesse während des Trainings des Lernverfahrens. Ziel dieser Analyse ist die quantitative Bewertung des Resultats dieser Änderungsplanungen mithilfe einer Strafkostenfunktion. Diese Strafkostenfunktion nutzt die identifizierten Kostenparameter³⁷ und führt die einzelnen Parameter in objektknotenspezifischen Strafkostenfunktionen zusammen. In diesen Strafkostenfunktionen werden sowohl lokale Restriktionen als auch globale Leistungsvereinbarungen berücksichtigt. Die Berechnung der Strafkosten erfolgt auf den charakteristischen Zuständen des Produktionsnetzwerkes. Die Kombinationen der einzelnen konzipierten Strafkostenfunktionen bildet die Rewardfunktion des Lernverfahrens. Um das Lernverfahren flexibel zu gestalten, sind die Rewardfunktion bzw. die einzelnen Strafkostenfunktionen umfassend parametrisierbar. Die Integration der Rewardfunktion in den Q-Learning-Algorithmus wird skizziert.

5.2.1. Planungsverfahren und Varianten im Lernsystem

Im Lernsystem führen die Objektknoten intern oder untereinander alle notwendigen Planungsprozesse zur kooperativen Änderungsplanung durch. Dabei hat jeweils ein Objektknoten die Rolle des *Initiators* und die anderen beteiligten Objektknoten die Rollen der *Partizipanten*. Die jeweilige Rolle ergibt sich aus der Art der Koordination und der Koordinationsrichtung. Bei der Anwendung eines lokalen Planungsverfahrens hat der jeweilige Objektknoten sowohl die Rolle des Initiators, als auch die Rolle eines Partizipanten.

Ziel der Arbeit ist es, Regeln zu erzeugen, die Empfehlungen geben, welches Planungsverfahren³⁸ zur Beseitigung eines ungünstigen Zustandes die wahrscheinlich besten Planungsergebnisse erzielen wird. Dabei gilt, dass für bestimmte Arten von Zuständen nur bestimmte Änderungsplanungsverfahren zugelassen sind.³⁹ Änderungsplanungsverfahren lassen sich in Verfahren mit lokaler Strategie und Verfahren mit globaler Strategie unterscheiden.⁴⁰

³⁷Vgl. Kapitel 2.2.5.3, S. 37

³⁸Hier werden beispielhaft die Änderungsplanungsverfahren von Heidenreich verwendet (siehe [Hei06]).

³⁹Vgl. Kap. 5.1.2.2, S. 79 ff.

⁴⁰Vgl. Kap. 2.1.3, S. 18

Bei lokalen Planungsverfahren wird versucht, ein planerisches Defizit oder einen Überschuss durch ein lokal begrenztes Änderungsplanungsverfahren innerhalb eines Objektknotens zu beseitigen.⁴¹ Hierzu wird entweder der betroffenen Periode die Restriktionsgrenze verschoben oder durch lokale Umplanungen der Lose ein Ausgleich des Überschusses oder Defizits erreicht. Die Anzahl der Varianten ist über die Anzahl bereitgestellter Algorithmen begrenzt, da die Anwendung der Planungsverfahren auf einen Objektknoten beschränkt ist.

Globale Änderungsplanungsverfahren bestimmen die Änderungsmenge, die als Erhöhung oder Senkung von Angebot oder Bedarf an Lieferanten oder Kunden, je nach Koordinationsrichtung, weitergegeben wird. Diese Menge wird durch das Planungsverfahren, z. B. vorwärts- oder rückwärtsterminiert, berechnet. Im Falle eines einzelnen Lieferanten wird die berechnete Menge an den angrenzenden Objektknoten weitergegeben. Bei alternativen Zu- oder Abgängen kann entweder die gesamte Bedarfs- oder Angebotsmenge an einen einzelnen Kunden oder Lieferanten weitergegeben werden, oder sie wird auf alternative Objektknoten aufgeteilt. Sollen Verteilungen von Änderungsanfragen zwischen Partizipanten am Änderungsplanungsprozess berücksichtigt werden, müssen diese Verteilungen als Variante eines globalen Änderungsplanungsverfahrens im Lernverfahren vorgegeben sein. Eine mögliche Regel wäre z. B. eine Gleichverteilung von Bedarfen über alle angrenzenden Objektknoten. In diesem Fall sind M_i die Teilmengen, die an die angrenzenden Objektknoten O_i für $i = 1, \dots, n$ Objektknoten weitergegeben werden:

$$M_i(O_i) = \frac{1}{n} \Delta \quad (5.8)$$

Aus der Unterscheidung lokaler und globaler Planungsverfahren ergibt sich die Notwendigkeit, diese bei der Konzeption der Rewardfunktion differenziert zu behandeln. Insbesondere dem Aspekt der unvollständigen Informationslage muss bei der Konzeption der Rewardfunktion für globale Verfahren Rechnung getragen werden.

5.2.2. Rewardbewertung auf Clusterebene

Die durch das Clustering erzeugten charakteristischen Zustände repräsentieren die Menge spezifisch auftretender Probleme innerhalb eines Produktionsnetzwerkes.⁴² Für jeden erzeugten charakteristischen Zustand ist die Art der anwendbaren Änderungsplanungsverfahren eingeschränkt.⁴³ Der Zustandsraum ist durch dessen Abstraktion auf charakteristische Zustände auf ein repräsentatives Maß reduziert. Die Bewertung von Zuständen während des Lernprozesses kann folglich auf diesen repräsentativen

⁴¹Enthalten in den Tab. A.2-A.3 im Anhang, S. 208 ff.

⁴²Vgl. Kap. 2.2.2.2, S. 26

⁴³Vgl. Kap. 5.1.2.1, S. 76 ff.

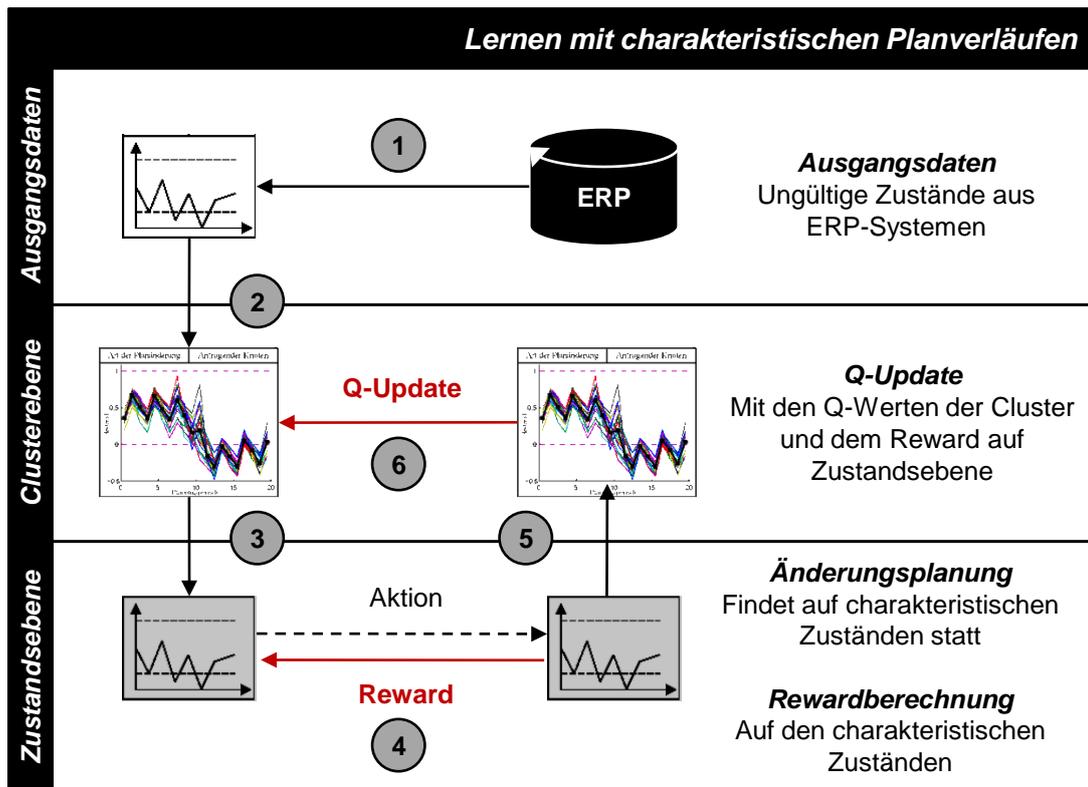


Abbildung 5.12.: Berechnung der Q-Werte im abstrahierten Zustandsraum

charakteristischen Zuständen durchgeführt werden. Durch ihre Verwendung als Ausgangsdaten für das Lernverfahren wird ein effizienter Lernprozess ermöglicht. Abbildung 5.12 verdeutlicht die Durchführung des Q-Updates unter Verwendung der charakteristischen Planverläufe.

Da die bereitgestellten Ausgangsdaten für das Training des Lernverfahrens aus ERP-Systemen stammen (1), muss während des Lernprozesses eine „Übersetzung“ dieser Daten in den charakteristischen Zustandsraum erfolgen. Für diese Übersetzung wird die Abstraktionsfunktion des Lernverfahrens verwendet (2). Wurde ein entsprechender Cluster identifiziert (3), findet auf dessen charakteristischem Plan⁴⁴ ein Änderungsplanungsprozess statt, der durch die Rewardfunktion bewertet wird (4). Zur Ermittlung des maximalen Q-Wertes des Folgezustandes wird für diesen erneut ein zugehöriger Cluster ermittelt (5). Das Q-Update wird abschließend mit dem Q-Wert des Clusters des Ausgangszustandes und dem maximalen Q-Wert des Clusters des Folgezustandes durchgeführt (6). Abbildung 5.12 veranschaulicht das im Weiteren zu vertiefende Prinzip der Rewardberechnung und des Q-Updates.

⁴⁴Es handelt sich bekanntlich um einen ungültigen Zustand (Vgl. Kapitel 2.2.5)

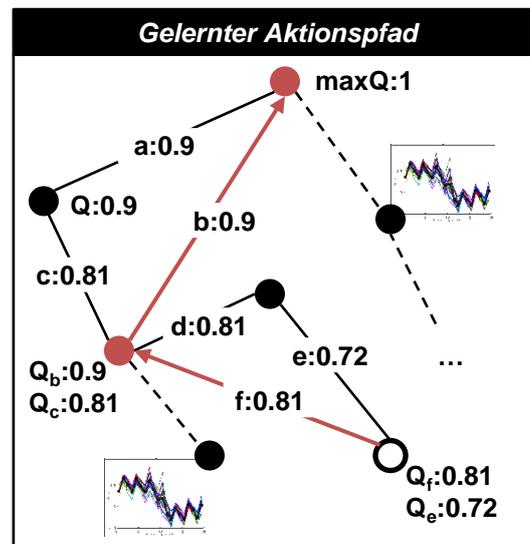


Abbildung 5.13.: Aktionspfad Q-Learning auf Clusterebene

Für die Q-Update-Regel aus Formel (2.1)⁴⁵ des Q-Algorithmus folgt für die Aktualisierung auf Clusterebene für den charakteristischen Plan C_s :

$$Q(C_s, a_i) \leftarrow Q(C_s, a_i) + \alpha \left[r_{C_s} + \gamma \max_{a_j} Q(C_{s+1}, a_j) - Q(C_s, a_i) \right] \quad (5.9)$$

Für das bereits bekannte Beispiel aus Abbildung 2.8 ergibt sich nach Abschluss des Lernprozesses für die Ausführung gelernter Regel ein Aktionspfad auf Clusterebene, wie er in Abbildung 5.13 skizziert ist.

Die Rewardfunktion für das Lernverfahren wird im Folgenden o. B. d. A. auf Zustandsebene definiert, da bei der Bewertung der charakteristischen Planungsverläufe im Training des Lernverfahrens diskrete Zustands-/Aktionspaare verwendet werden. Die Rewardfunktion soll das mögliche Entscheidungsverhalten eines Planers über mathematische Bewertungen von Plänen nachbilden können.⁴⁶ Dieses wird über die Bewertung von Aktionen durch Kostenfunktionen realisiert.

5.2.3. Strafkostenarten in der Rewardfunktion

Es konnten vier Strafkostenarten für den Untersuchungsgegenstand identifiziert werden:⁴⁷

⁴⁵Siehe Kap. 2.1, S. 35

⁴⁶Vgl. Kap. 2.2, S. 21 ff.

⁴⁷Siehe Kap. 2.1, S. 2.1 ff.

Bereitstellungsstrafkosten: Im FOK wird die Bereitstellung von Materialien geplant. Bereitstellungsstrafkosten werden aus den zeitlich entstehenden Kosten für Materialien in einem Lager hergeleitet. Diese sind z. B. Verlustwerte des Lagergutes und dessen Betriebskosten.⁴⁸ Die Bereitstellung von Material sowie die Relation bereitgestellter Materialien zu den Restriktionsgrenzen der Pläne kann über Strafkosten je Periode eines Planes bewertet werden.

Betriebsmittelstrafkosten: Die Leistung des Produktionsprozesses ist abhängig von den bereitgestellten Kapazitäten des KOK. Die bereitgestellte und verwendete Kapazität ist durch Strafkosten zwischen zwei Plänen bewertbar. Strafkosten entstehen in Relation der Auslastung des KOK zu dessen Restriktionsgrenzen und auch dann, wenn die bereitgestellte Kapazität nicht oder nicht in vollem Maße ausgenutzt wurde.⁴⁹ Die Betriebsmittelstrafkosten werden periodenweise berücksichtigt.

Beschaffungsstrafkosten: Müssen Materialien im Rahmen einer globalen Änderungsplanung beschafft werden, so ist das Ergebnis des Beschaffungsprozesses durch Strafkosten bewertbar. Die Strafkosten hängen von der Art der Leistungsvereinbarungen zwischen Kunden und Lieferanten und den Strafkosten für die Beschaffungsmenge des Materials ab.⁵⁰ Beschaffungskosten und Bereitstellungsstrafkosten sind gemeinsam zu betrachten, da eine Beschaffung vor Fertigungsbeginn in der Regel Änderungen der Bereitstellungsstrafkosten am KOK verursacht. Umgekehrt muss beschafft werden, wenn der Lagerbestand ein Minimum erreicht hat. Gleiches gilt vice versa für die Angebotskoordination. Strafkosten fallen in den Perioden an, in denen geplanter Materialfluss zwischen Kunde und Lieferant stattfindet.

Restriktionsverletzungen: Restriktionsverletzungen werden als charakteristische Merkmale von Plänen betrachtet. Sie bestimmen allgemein in Abhängigkeit ihrer Ausprägung die Verbesserungspotenziale eines Planes. Sie sind durch Strafkosten zu bewerten und periodenweise in der Rewardfunktion zu berücksichtigen.

Ein Zustand, der eine Änderungsplanung erfordert, ist jeder Zustand eines Objektknotens, der Restriktionsverletzungen in einer oder mehreren Perioden aufweist und so ungültig ist. Die Berechnung des Rewards eines Planes erfolgt auf den kumulierten Strafkosten je Planungsperiode je relevanter Strafkostenart.

⁴⁸Siehe z. B. [Gud04]

⁴⁹Z. B. in [GK98]

⁵⁰Ebd.

5.2.4. Grundprinzip bei der Rewardberechnung

Im Q-Learning ist es erforderlich, den Reward so zu berechnen, dass er die gewünschten Lernziele positiv oder negativ verstärkt. Dieses kann durch Belohnung oder Bestrafung einer durchgeführten Aktion umgesetzt werden. In der Änderungsplanung können über die durch einen Plan oder durch die Planung verursachten Kosten als Strafkosten abgebildet werden, womit dann der Reward einer Planungsaktion im Rahmen der Änderungsplanung ermittelt werden kann. Je geringer die Strafkosten nach erfolgter Änderungsplanung, desto effektiver kann dieses Planungsverfahren zur Steuerung für ein spezifisches Ereignis angewandt werden.

Der Reward einer Änderungsplanung berechnet sich aus der relativen Verbesserung des geänderten Zustandes, bezogen auf dem Ausgangszustand. Die Verbesserung oder Verschlechterung des Planes wird über berechnete Strafkosten gemessen. Die Strafkosten eines Planes sind die Summe der Strafkosten⁵¹ PC der einzelnen Perioden $p(k)$ im Planungshorizont PHZ eines Objektknotenplanes⁵² P_s^O :

$$PC(P^O) = \sum_{k=1}^{PH} PC(p(k)) \text{ mit } p(k) \in P_s^O \quad (5.10)$$

Um die Differenz der Strafkosten zwischen zwei Plänen zu ermitteln, muss die Differenz $DPC(P_z^O, P_{z+1}^O)$ ⁵³ zwischen den jeweiligen Strafkosten der Pläne berechnet werden.

$$DPC(P_s^O, P_{s+1}^O) = PC(P_{s+1}^O) - PC(P_s^O) \quad (5.11)$$

Die Verbesserung eines Planes durch eine Änderungsplanung wird erzielt, wenn die Strafkosten vom Ausgangszustand zum Folgezustand verringert werden können.

$$DPC(P_s^O, P_{s+1}^O) > 0 \Rightarrow PC(P_{s+1}^O) < PC(P_s^O) \text{ Verbesserung} \quad (5.12)$$

$$DPC(P_s^O, P_{s+1}^O) = 0 \Rightarrow PC(P_{s+1}^O) = PC(P_s^O) \text{ Keine Änderung} \quad (5.13)$$

$$DPC(P_s^O, P_{s+1}^O) < 0 \Rightarrow PC(P_{s+1}^O) > PC(P_s^O) \text{ Verschlechterung} \quad (5.14)$$

Durch die Berechnung des Rewards mithilfe der Strafkostenfunktion können positive oder negative Planungseffekte eines angewendeten Planungsverfahrens objektiviert bewertet werden. Die Erhöhung oder Verringerung der Strafkosten nach erfolgter Planung bewertet den erzielten Planungseffekt nach Durchführung eines Planungsverfahrens.

⁵¹Engl. *penalty costs*

⁵²Hier wird P^O verwendet, da die folgenden Aussagen allgemein für alle Objektknoten gelten.

⁵³ DPC engl. *difference penalty costs*

Es sei PA ⁵⁴ die Menge aller zugelassenen Planungsverfahren⁵⁵ mit $a_i \in PA$. Der Reward $R(P_s^O, a_i)$ eines Zustandsüberganges $P_s^O \rightarrow P_{s+1}^O$ des Plans P_s^O eines Objektknotens O berechnet sich aus der Differenz der Strafkosten eines Planes zum Zeitpunkt P_s vor und P_{s+1} nach Anwendung eines Planungsverfahrens $a_i \in PA$:

$$R(P_s^O, a_i) = PC(P_s^O, a_i) - PC(P_{s+1}^O) \quad \text{mit} \quad (5.15)$$

$$R(P_s^O, a_i) > 0 \quad \text{Verbesserung} \quad (5.16)$$

$$R(P_s^O, a_i) = 0 \quad \text{Keine Änderung} \quad (5.17)$$

$$R(P_s^O, a_i) < 0 \quad \text{Verschlechterung} \quad (5.18)$$

5.2.5. Vorlaufzeiten von Planungsprozessen in der Rewardfunktion

Restriktionsverletzungen eines Planes, z. B. fehlendes Material zur Befriedigung eines Kundenbedarfes, sind in der Regel bei längerer Vorlaufzeit in der Planung besser zu beseitigen als bei kürzeren. Für die Änderungsplanung bedeutet dieses, dass ungültige Zustände, deren Restriktionsverletzungen näher an der Heutelinie im Planungshorizont liegen, schwerer zu beseitigen sind, da weniger Spielraum für die Beseitigung, z. B. durch Vorziehen von Bedarfen, zur Verfügung steht. Kurzfristige Beseitigung von Restriktionsverletzungen geht oftmals mit erhöhten Kosten einher. Um dieses in der Rewardfunktion zu berücksichtigen, werden in der Zukunft auftretende Restriktionsverletzungen und dadurch entstehende Strafkosten zunehmend schwächer in der Rewardfunktion bewertet.

Dies kann durch Gewichtung der Strafkosten der Perioden des Planungshorizontes eines Planungsverlaufes mit einem Diskontfaktor⁵⁶ erreicht werden. Der Diskontfaktor DF wird je Planungsperiode $p(k)$ wie folgt verrechnet:

$$DF(p(k)) = \left(\frac{1}{(1+discount)} \right)^k \quad (5.19)$$

$$\text{mit } k = 1, \dots, PH \text{ und } p(k) \in P \text{ und } discount \in [0, 1]$$

Abbildung 5.14 skizziert dieses an einem Beispiel. Es wird sichergestellt, dass DF mit fortschreitenden Planungsperioden kleiner wird. Wird DF bei der Berechnung der Strafkosten des Planes angewandt, so wird jede Periode bei der Strafkostenberechnung mit dem jeweiligen Diskontfaktor berechnet. Der Reward erhöht sich bei steigenden und verringert sich bei sinkenden Strafkosten. Für die Summe der Strafkosten eines

⁵⁴Engl. *planning algorithm*

⁵⁵Z. B. die Planungsverfahren zur Änderungsplanung aus [Hei06]

⁵⁶Engl. *discount factor*

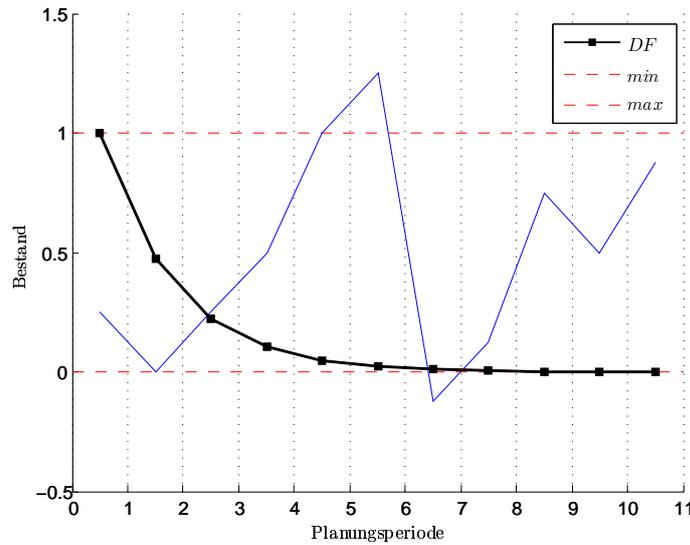


Abbildung 5.14.: Beispiel für die Abzinsung der Strafkosten durch den Diskontfaktor DF in Bezug zum Plan eines FOK

Planes P_s ergibt sich:

$$PC(P_s) = \sum_{k=1}^{PH} \left(\frac{1}{(1 + discount)^k} \right) PC(p(k)) \quad (5.20)$$

5.2.6. Bewertung von Restriktionsverletzungen

Die Beseitigung von Restriktionsverletzungen durch die Änderungsplanung kann abhängig von deren Anzahl⁵⁷ in einem Plan erschwert werden. Daher soll die Anzahl der Restriktionsverletzungen⁵⁸ eines Planes im Verhältnis zur Länge des Planungshorizontes über Strafkosten bewertet werden. Das Ergebnis ist eine Kennzahl, die die Anzahl von Restriktionsverletzungen relativ zum Planungshorizont misst. Hierzu werden beide Größen in ein Verhältnis gesetzt:

$$PC_{RV}(P_s) = \frac{RV}{PH} \quad (5.21)$$

Die Strafkosten der Restriktionsverletzungen werden *einmalig* auf die sonstigen Strafkosten der Objektknoten addiert.⁵⁹

⁵⁷Dieses gilt auch für die Höhe von Restriktionsverletzungen. Diese werden in einer separaten Kostenfunktion bewertet.

⁵⁸Engl. *restriction violation*.

⁵⁹Eine Anwendung siehe weiter unten in Formel (5.39), S. 120

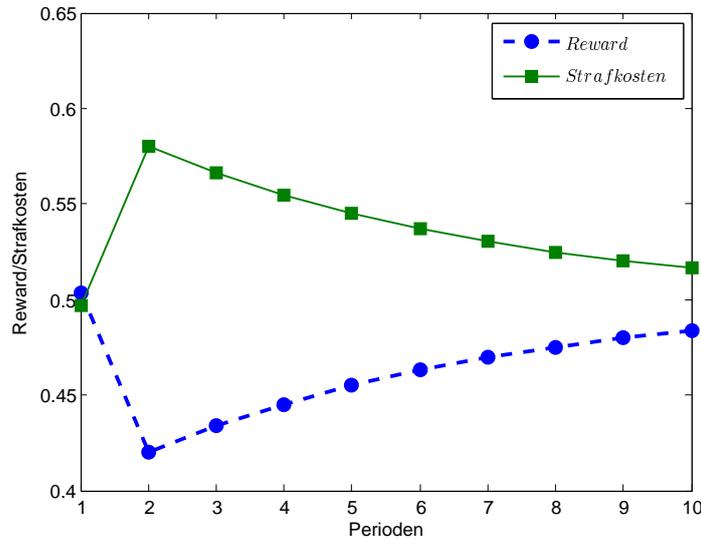


Abbildung 5.15.: Wirkungsweise der Abzinsung

5.2.7. Bereitstellungstrafkosten am Fertigungsobjektknoten

Für die FOK müssen Bereitstellungskosten als Bewertungsgrundlage der Rewardfunktion veranschlagt werden. Wie in Kapitel 2.1 beschrieben, sind die für das Lernsystem relevanten Restriktionen über die Bereitstellungsmenge⁶⁰ in Einheiten je Materialart, z. B. Stück ST , definiert. Neben den dort⁶¹ eingeführten Restriktionsgrenzen $p(k)_{min}^{sup}$, $p(k)_{max}^{sup}$ und $p(k)^{sup}$ werden zusätzliche Parameter benötigt, um die Restriktionsgrenzen für die Strafkostenberechnung in Relation zu setzen.

5.2.7.1. Parameter der Kostenfunktion

Durch $p(k)_{minlog}^{sup}$ und $p(k)_{absmax}^{sup}$ werden logische Grenzen für den Bestand definiert. Diese Überlegung ist auch in der Praxis wiederzufinden, da trotz einer *definierten* maximalen Lagerkapazität $p(k)_{max}^{sup}$ für die Planung durchaus in Notfällen weitere Kapazitäten bis zu einer physischen⁶² Maximalgrenze $p(k)_{absmax}^{sup}$ verwendet werden können. Gleiches gilt für den Sicherheitsbestand, der physisch bei $p(k)_{minlog}^{sup}$ erschöpft ist. Planerisch können bei $p(k)_{minlog}^{sup}$ negative Werte zugelassen werden, wenn ein Bedarf zu einem Zeitpunkt mehr Material erfordert, als im Bestand vorhanden ist.⁶³ Die Än-

⁶⁰Engl. *supply*

⁶¹Siehe Kap. 2.1.2, S. 17

⁶²Dieses gilt, wenn das Lager physisch mit Material gefüllt ist.

⁶³[Erl07]

derungsplanung hat die Aufgabe, diesen Wert planerisch zu einem Wert größer oder gleich Null auszugleichen.

Lokal durchgeführte Planungsverfahren operieren direkt auf den lokalen Planwerten der FOK und ändern dabei z. B. geplante Lose oder Restriktionsgrenzen. Hierbei werden Änderungen für den gesamten Plan, z. B. Änderungen am standardmäßigen Sicherheitsbestand, und temporäre Änderungen, z. B. Änderung des Maximalbestandes, für einen bestimmten Zeitraum unterschieden. Durch die Bewertung entstehender Bereitstellungskosten wird eine lokale Bewertung der Verbesserung oder Verschlechterung eines Planes ermöglicht.

Folgende Parameter orientiert an den Restriktionsgrenzen des FOK, werden in der Strafkostenfunktion für Bereitstellungskosten je Periode verwendet:⁶⁴

$p(k)_{min}^{sup}$	Festgelegte Soll-Bereitstellungsmenge ⁶⁵
$p(k)_{minlog}^{sup}$	Zur Strafkostenberechnung benötigter logischer minimaler Wert für Bereitstellungsmengen
$p(k)_{mintemp}^{sup}$	Temporäre Änderung der minimalen Bereitstellungsmenge durch ein Änderungsplanungsverfahren
$p(k)_{max}^{sup}$	Maximale Bereitstellungsmenge
$p(k)_{maxtemp}^{sup}$	Temporäre Änderung der maximalen Bereitstellungsmenge durch ein Änderungsplanungsverfahren
$p(k)_{absmax}^{sup}$	Absolutes Maximum der Bereitstellungsmenge ⁶⁶
$p(k)^{sup}$	Bestand der Periode

Es gilt

$$p(k)_{minlog}^{sup} < p(k)_{mintemp}^{sup} < p(k)_{min}^{sup} < p(k)_{max}^{sup} < p(k)_{maxtemp}^{sup} < p(k)_{absmax}^{sup} \quad (5.22)$$

stets für alle Restriktionsgrenzen und

$$p(k)_{minlog}^{sup} \leq p(k)^{sup} \leq p(k)_{absmax}^{sup} \quad (5.23)$$

für die Werte eines Planes.

5.2.7.2. Vergleichbarkeit der Strafkosten

Bei der Bewertung von Plänen durch Kosten ergibt sich zwischen verschiedenen Objektknoten⁶⁷ das Problem der Vergleichbarkeit der Strafkosten, wie folgendes Beispiel

⁶⁴Die für alle Perioden geltenden Standardgrenzen für Restriktionsverletzungen P_{min} und P_{max} werden hier zur konsistenten Formalisierung der Strafkostenfunktionen für jede Periode der Objektknoten einzeln angegeben.

⁶⁵Z. B. der Sicherheitsbestand in Stück

⁶⁶Z. B. physische Lagergrenze des modellierten FOK in ST

⁶⁷Dieses gilt sowohl für FOK als auch für KOK.

skizziert:

Beispiel 5.1 Ein Material A kostet 10 Geldeinheiten (GE) je ST in einer Lagerungsperiode. Material B kostet wiederum 100 GE je Lagerungsperiode. Beide Materialien können zu gleichen Teilen in das Lager eingelagert werden. In Plan-1 werden ausschließlich Materialien vom Typ A für ein Lager modelliert und in Plan-2 ausschließlich Materialien vom Typ B. Bei beiden Plänen wird für eine Periode die Menge 10 ST bis zum absoluten Maximum erhöht. Bei einer direkten Bewertung der Restriktionsgrenzenüberschreitung entstehen für Plan-1 Strafkosten von 100 GE und bei Plan-2 von 1000 GE.

Die ermittelten Strafkosten sind nicht unmittelbar miteinander vergleichbar. Die Strafkosten sind bei Plan-2 trotz gleicher Mengen um 900 GE höher als bei Plan-1. Die Strafkosten sollen aber für unterschiedliche Basiskosten vergleichbar bleiben.

Um dieses zu erreichen, wird im Lernverfahren zunächst von den realen Kosten eines Planes abstrahiert und vielmehr die für die Auswahl und Bewertung von Änderungsplanungsverfahren relevanten Perioden bewertet. Dieses sind die Perioden, in denen eine Restriktionsverletzung vorliegt. Für FOK⁶⁸ wird daher eine Bewertung der Grenzbereiche eines Planes vorgeschlagen. Es soll differenziert bewertet werden:

$PC_AON_{min}^{loc}(P_s)$ Bewertet, wie viel Material aus dem Sicherheitsbestand zur Auflösung eines ungültigen Planes entnommen wurde, sowie in welcher Höhe Restriktionsgrenzen verschoben wurden

$PC_AON_{max}^{loc}(P_s)$ Bewertet, wie viel Material über den Maximalbestand hinaus eingelagert werden muss, um den vollständigen Zugang vorhalten zu können, sowie in welcher Höhe Restriktionsgrenzen verschoben wurden

$PC_AON_{cost}^{loc}(P_s)$ Bewertet die Grundkosten für die Lagerung eines FOK

Da nicht mit direkten Kosten gerechnet wird, soll für die Berücksichtigung der Grundkosten eine Kennzahl ähnlich der Kennzahl zur Berücksichtigung von Restriktionsverletzungen in der Strafkostenfunktion verwendet werden.⁶⁹ Die Kombination der einzelnen Kostenparameter in einer Summe bildet die gesamte Strafkostenfunktion am FOK.

Weiterhin werden die Strafkosten zur Vergleichbarkeit normiert. Die Normierung der oben dargestellten Kostenparameter auf das Intervall $[0, 1]$ ermöglicht die einheitliche Betrachtung der Strafkosten in der Rewardfunktion und so auch bei der Q-Wertberechnung. Wie Abbildung 5.16 zeigt, werden die Strafkosten der restriktiven Grenzen eines Planes bzw. einer Planungsperiode mit 0 bewertet, sofern keine Restriktionsverletzung

⁶⁸Engl. *assembly object node* (AON)

⁶⁹Details folgen in Kap. 5.2.7.4

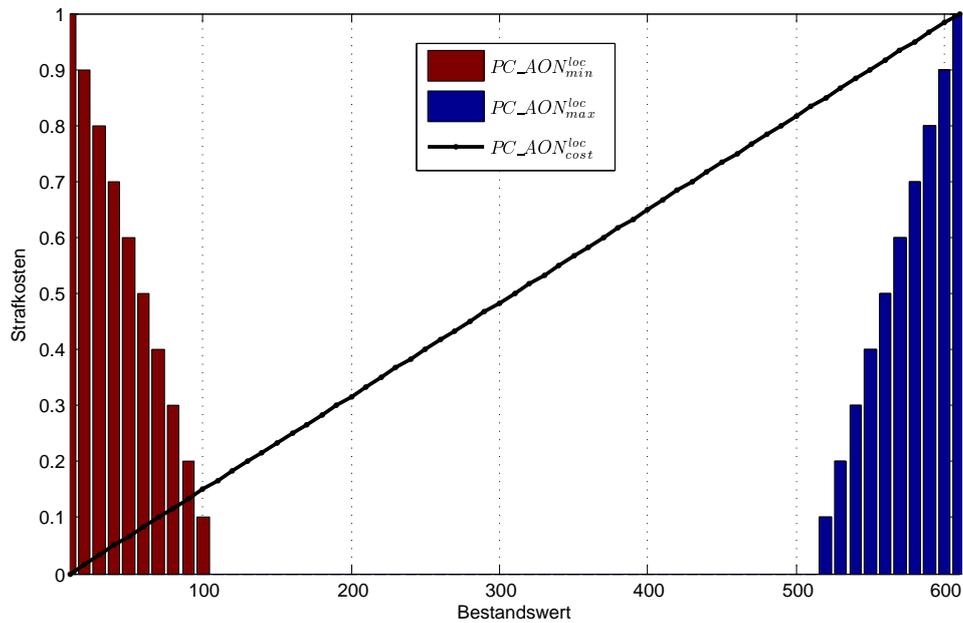


Abbildung 5.16.: Skizzierung der Wirkungsweise der Strafkostenfunktion für Bestandswerte ohne Betrachtung verschobener Restriktionsgrenzen

vorliegt, und mit einem Wert größer 0, wenn eine Restriktionsverletzung vorliegt. Ändern sich die Anzahl der betrachteten Perioden und die Restriktionsgrenzen nicht, so bleiben die maximalen Strafkosten je Plan gleich. Sowohl die Strafkosten, als auch der Reward und der Q-Wert sind so zwischen verschiedenen Zuständen vergleichbar.

5.2.7.3. Periodenweise Strafkostenfunktion für lokale Planänderungen

Um die Kosten von $PC_AON^{loc}_{min}(P_s)$ zu bestimmen, wird ein Bestand von $p(k)^{sup}_{minlog}$ mit den Strafkosten von 1 und ein Bestand von $p(k)^{sup}_{min}$ mit 0 bewertet. Da jede Periode eines Planes durch differierende Restriktionen definiert sein kann, werden die Strafkosten des Planes eines FOK $p(k)^{sup} \in P_s$ einer Periode durch $PC_AON^{loc}_{min}(p(k))$ berechnet, wobei die aktuellen Werte von $p(k)^{sup}_{min}$ und $p(k)^{sup}_{max}$ zur Berechnung verwendet werden.⁷⁰

Wird durch eine Änderungsplanung eine Restriktionsgrenze verschoben, z. B. durch die Senkung eines Sicherheitsbestandes für eine Periode, so kann diese Verschiebung durch Strafkosten bewertet werden. Wäre dieses nicht der Fall, so wäre die Verschiebung einer Restriktionsgrenze stets der kostengünstigste Fall, da dadurch ein ungültiger Plan einfach wieder in einen gültigen Plan überführt werden könnte. Globale

⁷⁰ $p(k)^{sup}_{min}$ und $p(k)^{sup}_{max}$ können von den Standardwerten abweichen, wenn z. B. der Sicherheitsbestand $p(k)^{sup}_{min}$ einer Periode $p(k)$ durch ein Planungsverfahren in vorherigen Planungsläufen reduziert oder erhöht wurde.

Aktionen würden dann im Regelsystem nicht priorisiert werden. Dadurch kann ein Ausgleich zwischen der Bestrafung lokaler und globaler Aktionen erreicht werden, da die Verschiebung einer Restriktionsgrenze selbst wieder Strafkosten verursacht.

Um die temporären Änderungen von Restriktionsgrenzen zu bestrafen, wird je Periode das Verhältnis zwischen der Höhe der temporären Restriktionsgrenze $p(k)_{mintemp}^{sup}$ zum Standardwert $p(k)_{min}^{sup}$ bewertet. Wurde diese nicht verschoben, so ergeben sich Strafkosten von 0. Wurde der vollständige Sicherheitsbestand benötigt, betragen die Strafkosten 1.⁷¹ Durch diese Verschiebung können Ergebnisse von Änderungsplanungsverfahren, die lokale Restriktionsgrenzen verändern, bewertet werden. Sofern keine Restriktionsgrenze verschoben wurde, gilt $p(k)_{min}^{sup} = p(k)_{mintemp}^{sup}$. Anderweitig wird $p(k)_{mintemp}^{sup}$ zur Berechnung der Strafkosten verwendet.⁷²

$$\begin{aligned}
 PC_AON_{min}^{loc}(p(k)) &= \omega_{min}^{loc} \left(\min \left(1; \left[\frac{\max(0; p(k)_{mintemp}^{sup} - p(k)_{minlog}^{sup})}{(p(k)_{mintemp}^{sup} - p(k)_{minlog}^{sup})} \right] \right) \right) \quad (5.24) \\
 &+ \omega_{mintemp}^{loc} \left(1 - \min \left(1; \left[\frac{(p(k)_{mintemp}^{sup} - p(k)_{minlog}^{sup})}{(p(k)_{min}^{sup} - p(k)_{minlog}^{sup})} \right] \right) \right) \\
 &\text{mit } \omega_{min}^{loc}, \omega_{mintemp}^{loc} \in [0, 1]
 \end{aligned}$$

Für $p(k)_{minlog}^{sup} = p(k)_{min}^{sup}$ wird der erste Quotient 1 und das Gesamtergebnis des Terms bleibt durch $\min(1; 1) = 1$. Ist $p(k)_{minlog}^{sup} < p(k)_{min}^{sup} < p(k)_{min}^{sup}$ nimmt der Quotient einen Wert zwischen $[0, 1]$ als Strafkosten an. Bei $p(k)_{min}^{sup} \leq p(k)_{minlog}^{sup} < p(k)_{absmax}^{sup}$ wird der Zähler durch $\min(0; -x) = 0$ und dadurch der Gesamtquotient 0. Es werden nur Werte von $p(k)_{minlog}^{sup}$ bestraft, die zwischen den Minimalgrenzen $p(k)_{minlog}^{sup}$ und $p(k)_{min}^{sup}$ einer Periode liegen.

Über das Gewicht ω_{min}^{loc} kann der Planer festlegen, welche Bedeutung die Unterschreitung des Sicherheitsbestandes bei der Strafkostenbewertung einnimmt. Je größer ω_{min}^{loc} gewählt wird, desto höher werden die zu verrechnenden Strafkosten im Rahmen der Rewardberechnung berücksichtigt und desto größer ist der Effekt auf die referenzierten Q-Werte. $PC_AON_{min}^{loc}(p(k))$ wird auf das Intervall $[0, 1]$ gedeckelt, da alle Strafkosten außerhalb dieses Intervalls auf $p(k)_{minlog}^{sup} < p(k)_{min}^{sup}$ oder $p(k)_{min}^{sup} > p(k)_{absmax}^{sup}$ zurückzuführen und mit der normierten Strafe 1 zu legen sind.

Durch den zweiten Quotienten von Formel (5.24) werden die Strafkosten bei der Verschiebung von Restriktionsgrenzen einzelner Perioden bestraft. Als Relation wird die

⁷¹Für Verschiebung von $mintemp$ sind Werte von $mintemp < min$ zugelassen. Die Erhöhung des Sicherheitsbestandes ist im Rahmen der Änderungsplanung zur Auflösung ungültiger Zustände als nicht relevant einzustufen.

⁷²In der folgenden Beschreibung wird der allgemeine Fall mit $p(k)_{min}^{sup}$ dargestellt.

Standardrestriktionsgrenze $p(k)_{min}^{sup}$ verwendet. Je höher die Abweichung der neuen Restriktionsgrenzen $p(k)_{mintemp}^{sup}$ von $p(k)_{min}^{sup}$, desto höher die Strafkosten. Ist $p(k)_{min}^{sup} = p(k)_{mintemp}^{sup}$ sind die Strafkosten 0, da keine Verschiebung stattgefunden hat. Bei $p(k)_{mintemp}^{sup} = p(k)_{minlog}^{sup}$ werden die Strafkosten mit 1 maximal. Die Normierung der Strafkosten wird eingehalten, da entweder die Bestandsänderung *oder* eine Änderung von Restriktionsverletzungen bestraft wird.

Die Strafkosten bei der Überschreitung des $PC_AON_{max}^{loc}(p(k))$ werden analog festgelegt. Hier wird eine Periode mit 0 bewertet, wenn der Bestand unterhalb von $p(k)_{max}^{sup}$ liegt, und mit 1, wenn der Bestand $p(k)_{absmax}^{sup}$ oder höher ist.

$$\begin{aligned}
 PC_AON_{max}^{loc}(p(k)) &= \omega_{max}^{loc} \min \left(1; \left[\frac{\max(0; p(k)^{sup} - p(k)_{maxtemp}^{sup})}{(p(k)_{absmax}^{sup} - p(k)_{maxtemp}^{sup})} \right] \right) \\
 &+ \omega_{maxtemp}^{loc} \min \left(1; \left[\frac{(p(k)_{maxtemp}^{sup} - p(k)_{max}^{sup})}{(p(k)_{absmax}^{sup} - p(k)_{max}^{sup})} \right] \right) \\
 &\text{mit } \omega_{max}^{loc}, \omega_{maxtemp}^{loc} \in [0, 1]
 \end{aligned}
 \tag{5.25}$$

Sofern temporär keine Restriktionsgrenze verschoben⁷³ wurde, gilt $p(k)_{max}^{sup} = p(k)_{maxtemp}^{sup}$. Anderweitig wird $p(k)_{maxtemp}^{sup}$ zur Berechnung der Strafkosten verwendet.⁷⁴ Hier wird der umgekehrte Effekt wie in Formel (5.24) erzielt, in dem für Werte im Bereich $p(k)_{minlog}^{sup} < p(k)^{sup} \leq p(k)_{max}^{sup}$ keine Strafkosten anfallen, für $p(k)^{sup} = p(k)_{absmax}^{sup}$ die Strafkosten 1 betragen und für $p(k)_{max}^{sup} < p(k)^{sup} < p(k)_{absmax}^{sup}$ Strafkosten zwischen $[0, 1]$ anfallen. Es werden nur Werte zwischen den Grenzen $p(k)_{max}^{sup}$ und $p(k)_{absmax}^{sup}$ bestraft. ω_{max}^{loc} ist der Gewichtungsfaktor zur Steuerung des Effektes der Strafkosten in der Rewardberechnung und bei der Aktualisierung der Q-Werte.

5.2.7.4. Periodenweise Strafkosten am FOK

Da der Bereitstellungsprozess von Materialien aus dem Lager an die Produktionslinie Kosten verursacht, werden diese Kosten ebenfalls bewertet. Die Verwendung der maximal vorgehaltenen Bereitstellungsmenge mit $p(k)^{sup} = p(k)_{max}^{sup}$ verursacht höhere Strafkosten als eine mit $p(k)^{sup} = p(k)_{min}^{sup}$ minimal gewählte Menge. Die Integration von Bereitstellungskosten in die Strafkostenfunktion bewirkt, dass durch das Lernverfahren auf Dauer Änderungsplanungsverfahren bevorzugt werden, die diese Kosten

⁷³Für die Verschiebung von *maxtemp* sind Werte von $maxtemp < absmax$ zugelassen, da eine Verschiebung über die maximale Lagergrenze nicht möglich ist.

⁷⁴In der folgenden Beschreibung wird der allgemeine Fall mit $p(k)_{max}^{sup}$ dargestellt.

minimieren können. Sollen Bereitstellungskosten beim Lernen nicht berücksichtigt werden, können diese durch Zuweisung $\omega_{cost}^{loc} = 0$ aus der Rewardkostenberechnung ausgeklammert werden.

Die Kosten für Bereitstellungsprozesse $PC_AON_{cost}^{loc}(p(k))$ werden in dieser Arbeit als linear angenommen.⁷⁵ Dieses kann auf viele Fälle so oder ähnlich zutreffen, da die Bereitstellungskosten mit zunehmendem Füllgrad des Lagers steigen können.⁷⁶

$$PC_AON_{cost}^{loc}(p(k)) = \omega_{cost}^{loc} \min\left(1; \frac{p(k)^{sup}}{p(k)_{absmax}}\right) \text{ mit } \omega_{cost}^{loc} \in [0, 1] \quad (5.26)$$

5.2.7.5. Kumulierte Strafkosten am FOK

Die lokalen Strafkosten einer Periode eines FOK $PC_AON^{loc}(p(k))$ werden als Summe der oben definierten Strafkostenparameter definiert.

$$\begin{aligned} PC_AON^{loc}(p(k)) &= PC_AON_{min}^{loc}(p(k)) \\ &+ PC_AON_{cost}^{loc}(p(k)) \\ &+ PC_AON_{max}^{loc}(p(k)) \end{aligned} \quad (5.27)$$

Die Summe der enthaltenen Gewichte muss 1 betragen, um einen normierten Strafkostenwert für die lokalen Strafkosten zu erhalten.

$$\omega_{max}^{loc} + \omega_{maxtemp}^{loc} + \omega_{min}^{loc} + \omega_{mintemp}^{loc} + \omega_{cost}^{loc} = 1 \quad (5.28)$$

Die gesamten an einem FOK lokal anfallenden Strafkosten $PC_AON^{loc}(P_s)$ berechnen sich aus der Summe der diskontierten Strafkosten je Periode eines Planes, summiert mit den Strafkosten für Restriktionsverletzungen dieses Planes:

$$\begin{aligned} PC_AON^{loc}(P_s) &= \omega_{rv}^{aon} \cdot PC_RV(P_s) \\ &+ \omega_{pc}^{aon} \cdot \left[\sum_{k=1}^{PH} DF(p(k)) \cdot PC_AON^{loc}(p(k)) \right] \end{aligned} \quad (5.29)$$

Die Strafkosten des FOK werden mit den Strafkosten der Restriktionsverletzungen am FOK summiert und entsprechend durch die Gewichte $\omega_{rv}^{aon} + \omega_{pc}^{aon} = 1$ in ihrer Relevanz für die Höhe der Strafe gewichtet.

⁷⁵Siehe Abb. 5.16

⁷⁶In dieser Arbeit soll gezeigt werden, wie ein maschinelles Lernsystem von Regeln zur Steuerung der Änderungsplanung in Produktionsnetzwerken zu konzipieren ist. Die lineare Bewertung der Bereitstellungskosten kann nach Bedarf durch alternative Funktionen ersetzt werden.

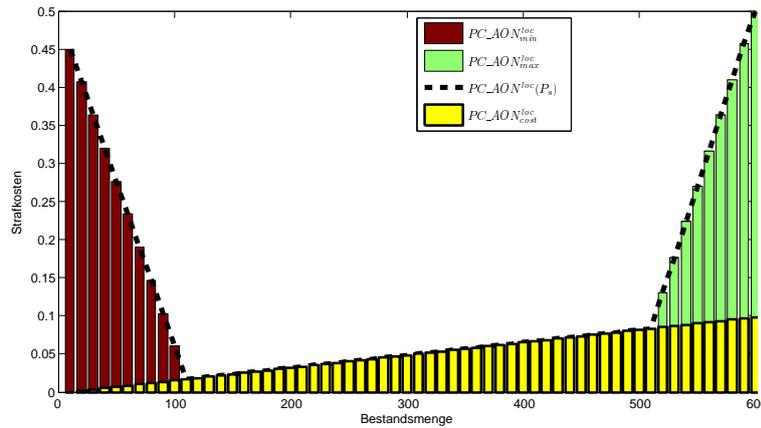


Abbildung 5.17.: Strafkostenberechnung am Beispiel eines FOK-Plans

Abbildung 5.17 zeigt die Anwendung der Strafkostenberechnung mit einer Plankonfiguration von:

$$\begin{aligned}
 p(k)_{min}^{sup} &= 110 & \omega_{min}^{loc} &= 0,45 \\
 p(k)_{max}^{sup} &= 510 & \omega_{max}^{loc} &= 0,45 \\
 p(k)_{absmax}^{sup} &= 600 & \omega_{cost}^{loc} &= 0,1 \\
 p(k)_{minlog}^{sup} &= 0
 \end{aligned}$$

Der Wert von $p(k)^{sup}$ läuft zwischen $[0, p(k)_{absmax}^{sup}]$. Aus der Resultierenden der Strafkostenfunktion können als maximale Strafkosten eine Periode $PC_AON_{max}^{loc}(p(k)) = 0.55$ abgelesen werden und für den gesamten Plan die maximalen Strafkosten von

$$PC_AON_{max}^{loc}(P_s) = PC_AON_{max}^{loc}(p(k)) \cdot PH = 5.5$$

mit $PH = 10$ für einen Planungshorizont von 10 Perioden berechnet werden.⁷⁷ Die Strafkosten für Restriktionsverletzungen fallen nur einmalig für die einzelnen Strafkosten am FOK an und werden hinzuaddiert.

5.2.8. Betriebsmittelstrafkosten am Kapazitätsobjektknoten

KOK modellieren die Kapazität verfügbarer Betriebsmittel für den Produktionsprozess. Die Interpretation der Kapazität am KOK erfolgt abstrahiert von der Materialart.⁷⁸ Durch das Lernverfahren werden für KOK Regeln gelernt, die die Anwendung

⁷⁷Ohne Berücksichtigung von DF .

⁷⁸Vgl Kap. 2.1.2, S. 16

5. Konzeption

von Änderungsplanungsverfahren für ungültige Zustände am KOK empfehlen. Beispiele hierfür sind⁷⁹:

- Leistungsgraderhöhung, bspw. durch zusätzliche Schichten
- Nettoangebotserhöhung (zeitabschnittsfixiert oder mengenfixiert) innerhalb von Kapazitätsrestriktionen im Rahmen eines Gegenvorschlages
- Nettoangebotsverdichtung durch Lossplittung oder Losverschiebung beginnend am Bedarfstermin in Richtung „Heutelinie“
- Bedarfsdeckung eines anfragenden PK durch eine Nettoangebotsreduzierung bei alternativ abgehendem PK

Bei der Änderungsplanung können Erhöhungen des oder Umplanungen innerhalb eines vorgegebenen Leistungsgrades auf die Prüfung der Verfügbarkeit von Betriebsmitteln bzw. deren Leistungsgrad beschränkt werden.

5.2.8.1. Parameter der Strafkostenfunktion

Beim KOK wird die Strafkostenfunktion unter Berücksichtigung der Restriktionsgrenzen am KOK in folgende Parameter aufgeteilt:⁸⁰

$p(k)_{min}^{pl}$	Minimale auszulastende Kapazität ⁸¹ eines KOK ⁸²
$p(k)_{mintemp}^{pl}$	Temporäre auszulastende Kapazität
$p(k)_{max}^{pl}$	Verfügbarer Leistungsgrad
$p(k)_{maxtemp}^{pl}$	Temporärer verfügbarer Leistungsgrad nach Anwendung eines Änderungsplanungsverfahrens
$p(k)_{absmax}^{pl}$	Physikalisches absolutes Maximum an Leistungsgrad der Betriebszeit eines Betriebsmittels
$p(k)^{pl}$	Zugewiesenes Kapazitätsangebot in einer Periode $p(k)$

Diese werden um die logische Grenze $p(k)_{absmax}^{pl}$ wie bei den FOK erweitert. Die Kostenbewertung im Lernprozess erfolgt auf den zur Verfügung gestellten Kapazitäten an

⁷⁹[Hei06], S. 111-119

⁸⁰Siehe Kap. 2.1.2.

⁸¹Engl. *performance level*

⁸²Ein Wert größer als Null ist durchaus sinnvoll, wenn bspw. eine Maschine aus wirtschaftlichen Gründen eine Minimallast fahren muss oder Maschinen aufgrund mehrwöchig in die Zukunft fixierter Schichtmodellen kontinuierlicher gefahren werden müssen und bspw. auf Lager gefertigt wird. Die Modellierung eines logischen minimalen Leistungsgrades entfällt, da ein negativer Leistungsgrad nicht sinnvoll ist.

einem KOK. Es gilt

$$p(k)_{mintemp}^{pl} < p(k)_{min}^{pl} < p(k)_{max}^{sup} < p(k)_{maxtemp}^{pl} < p(k)_{absmax}^{pl} \quad (5.30)$$

stets für alle Restriktionsgrenzen und

$$0 \leq p(k)^{pl} \leq p(k)_{absmax}^{pl} \quad (5.31)$$

für die Werte eines Planes.

5.2.8.2. Periodenweise Strafkosten am KOK

Da diskrete Belegungen unterschiedlicher KOK, wie bei den FOK, nicht unmittelbar miteinander vergleichbar sind, wird auch hier eine Normierung der Strafkosten im Intervall $[0, 1]$ vorgeschlagen. Dabei werden die Strafkosten nach dem gleichen Prinzip berechnet wie die Strafkosten der FOK. Die Parameter der Kostenfunktion am KOK⁸³ werden dabei als Strafkosten für die Unterschreitung der Minimalauslastung oder Überschreitung des Leistungsgrades und generelle Bereitstellungsstrafkosten für Betriebsmittel interpretiert. Als Strafkostenparameter je Periode eines Planes werden festgelegt:

$PC_CON_{max}^{loc}(P_s)$	Strafkosten für Überschreitung der logischen Leistungsgradgrenze, z. B. durch Einplanung von Zusatzschichten
$PC_CON_{min}^{loc}(P_s)$	Strafkosten für die Unterschreitung einer minimal auszulastenden Kapazität des KOK, z. B. bei Kurzarbeit o. ä. Ereignissen
$PC_CON_{cost}^{loc}(P_s)$	Generelle Kosten der Leistungsbereitstellung am KOK $p(k)$

Daraus ergeben sich je Periode $p(k) \in P^{CON}$ die Kosten als einzelne Quotienten analog zu den FOK:

$$PC_CON_{min}^{loc}(p(k)) = \omega_{min}^{con} \left(\min \left(1, \left[\frac{\max(0; p(k)_{min}^{pl} - p(k)^{pl})}{p(k)_{min}^{pl}} \right] \right) \right) \quad (5.32)$$

$$+ \omega_{mintemp}^{con} \left(1 - \min \left(1, \frac{(p(k)_{mintemp}^{pl} - p(k)_{min}^{pl})}{p(k)_{min}^{pl}} \right) \right)$$

⁸³Engl. *capacity object node*

$$PC_CON_{max}^{loc}(p(k)) = \omega_{max}^{con} \left(\min \left(1, \left[\frac{\max(0; p(k)^{pl} - p(k)_{max}^{pl})}{(p(k)_{absmax}^{pl} - p(k)_{max}^{pl})} \right] \right) \right) \quad (5.33)$$

$$+ \omega_{maxtemp}^{con} \left(\min \left(1; \left[\frac{(p(k)_{maxtemp}^{pl} - p(k)_{max}^{pl})}{(p(k)_{absmax}^{pl} - p(k)_{max}^{pl})} \right] \right) \right)$$

$$PC_CON_{cost}^{loc}(p(k)) = \omega_{cost}^{con} \left(\frac{p(k)^{pl}}{p(k)_{absmax}^{pl}} \right) \quad (5.34)$$

mit $\omega_{min}^{con}, \omega_{mintemp}^{con}, \omega_{max}^{con}, \omega_{absmax}^{con}, \omega_{cost}^{con} \in [0, 1]$

5.2.8.3. Kumulierte Strafkosten am KOK

Die gesamten Strafkosten für einen KOK ergeben sich aus der Summe der periodenbezogenen Strafkosten dessen Planes in Abhängigkeit von dessen Restriktionen in den Perioden:

$$PC_CON^{loc}(p(k)) = PC_CON_{min}^{loc}(p(k)) \quad (5.35)$$

$$+ PC_CON_{cost}^{loc}(p(k))$$

$$+ PC_CON_{max}^{loc}(p(k))$$

Wegen der Normierung der Strafkosten gilt für die Konfiguration der Gewichtungsfaktoren:

$$\omega_{min}^{con} + \omega_{mintemp}^{con} + \omega_{max}^{con} + \omega_{maxtemp}^{con} + \omega_{cost}^{con} = 1 \quad (5.36)$$

Die gesamten lokalen Strafkosten $PC_CON^{loc}(P_s)$ am KOK berechnen durch die Kumulation der um DF^{84} je Periode $p(k)$ abgezinsten Strafkosten dessen Planes P :

$$PC_CON^{loc}(P_s) = \omega_{rv}^{con} \cdot PC_RV(P_s) \quad (5.37)$$

$$+ \omega_{pc}^{con} \cdot \left[\left(\sum_{k=1}^{PH} DF(p(k)) \cdot PC_CON^{loc}(p(k)) \right) \right]$$

Die Strafkosten des KOK werden mit den Strafkosten der Restriktionsverletzungen am KOK summiert und entsprechend durch die Gewichte $\omega_{rv}^{con} + \omega_{pc}^{con} = 1$ in ihrer Relevanz für die Höhe der Strafe bewertet. Eine bildliche Darstellung der Strafkosten am KOK kann in Analogie zu den FOK der Abbildung 5.17⁸⁵ entnommen werden.

⁸⁴Siehe Formel (5.20)

⁸⁵Siehe S. 115

5.2.9. Beschaffungsstrafkosten am Fertigungsobjektknoten

In der Änderungsplanung ist es für die Beschaffungssteuerung möglich, Bedarfe oder Angebote an Lieferanten weiterzugeben, um benötigte Materialien in den erforderlichen Perioden zur Materialtransformation planerisch bereitstellen zu können. Die Beschaffung wird hier als wechselseitiger Koordinationsprozess zwischen Kunde und Lieferant betrachtet, bei dem je nach Koordinationsrichtung sowohl angebotsseitige als auch bedarfsseitige Änderungen angefragt werden können, die vorab kapazitär abgesichert werden müssen. Abbildung 5.18 illustriert eine erfolgreiche globale Koordination. In (1) fragt der Kunde bei K1 an, ob dieser eine Bruttobedarfserhöhung von 50 ST in p(3) erfüllen kann. K1 antwortet positiv, sodass diese Anfrage in (2) an L1 durchgeführt werden kann.

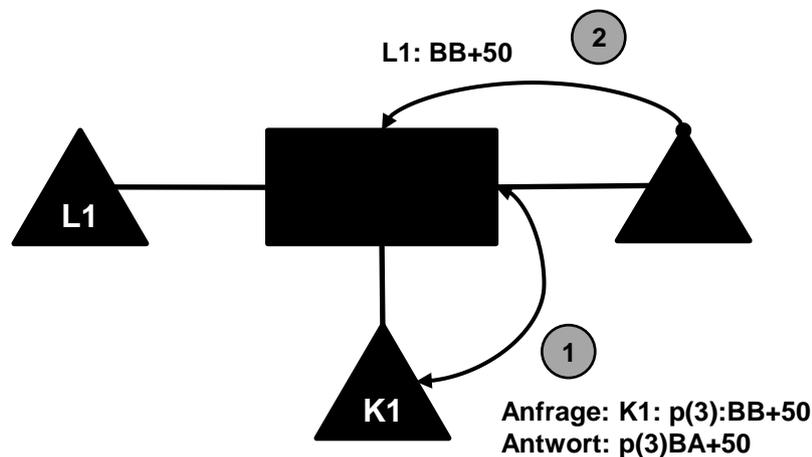


Abbildung 5.18.: Prinzip der globalen Koordination

Bei der Rewardberechnung wird die angebots- und bedarfsseitige Koordination gleich behandelt und aus Sicht des Initiators einer Koordination ermittelt. Die Höhe der Anfrage an einen Kunden oder Lieferanten kann in Höhe der benötigten Menge und unter Berücksichtigung zur Verfügung stehender Betriebsmittel in der Produktion erfolgen. Der Initiator ist im Rahmen der kooperativen Änderungsplanung derjenige Partner, der einen ungültigen Zustand auflösen möchte. Das Ergebnis der Änderungsplanung beim Initiator ist ein gültiger oder ungültiger Plan. Die Koordinationsrichtung bei der Änderungsplanung hat keinen Einfluss auf das Prinzip der Rewardberechnung, da der Reward trotz der Verwendung einer globalen Aktion bezogen auf den lokalen Plan des Initiators berechnet werden kann. Daher erfolgt die Darstellung der Rewardberechnung für Beschaffungsprozesse im Folgenden exemplarisch anhand der Bedarfskoordination.⁸⁶ Um die in der Rewardfunktion zu berücksichtigenden Strafkostenparameter

⁸⁶Durch die Koordinationsrichtung bedingte Unterschiede der definierten Strafkostenfunktion werden

bestimmen zu können, werden die Kosten für unterschiedliche Beschaffungspolitiken analysiert. Weiterhin wird ausgeführt, wie die KOK in den entstehenden kumulierten Strafkosten für Beschaffungsprozesse berücksichtigt werden.

5.2.9.1. Strafkosten für unterschiedliche Beschaffungssteuerungen

Bei der Bedarfskoordination muss zwischen zwei unterschiedlich zu bewertenden Beschaffungsarten unterschieden werden: Bestellpunktverfahren und Bestellzyklusverfahren. Beide Verfahren unterliegen unterschiedlichen Leistungsvereinbarungen, wie in Kapitel 2.1 definiert wurde.

Gemeinsam ist beiden Verfahren, dass jeweils in einem bestimmten Zeitraum eine maximale oder minimale Menge an Material⁸⁷ $p(b)_{max}^{mf}$ bzw. $p(b)_{min}^{mf}$ beschafft werden kann. Die entstehenden Kosten je Material im Beschaffungsprozess können wiederum als globale Strafkosten $PC_AON^{glob}(P_s)$ abgebildet werden. Konkret werden diese in Strafkosten für bestellpunktbasierte (PROC⁸⁸) Koordination $PC_AON^{proc}(P_s^D)$ ⁸⁹ und bestellzyklusbasierte (CREP)⁹⁰ Koordination $PC_AON^{crep}(P_s)$ aufgesplittet:⁹¹

$$PC_AON^{glob} = \begin{cases} PC_AON^{proc} & \text{für Bestellpunkt} \\ PC_AON^{crep} & \text{für Bestellzyklus} \end{cases} \quad (5.38)$$

Die Kosten für den Beschaffungsprozess am FOK sind mit den lokalen Kosten am FOK $PC_AON^{loc}(P_s)$ zu kumulieren, da die lokalen Kosten, wie z. B. Lagerkosten, für das beschaffte Material anfallen. Bei lokaler Änderungsplanung betragen die globalen Strafkosten für die Beschaffung 0. Die Strafkosten für die Restriktionsverletzungen werden addiert und durch ω_{aon}^{rv} gewichtet. Für Beschaffungen außerhalb von Leistungsvereinbarungen werden Strafkosten durch den Parameter $PC_OOP(P)$ ⁹² erfasst und durch ω_{aon}^{oop} gewichtet.

Für die Berechnung der Gesamtstrafkosten eines Planes P_s am FOK gilt:

$$\begin{aligned} PC_AON(P_s) &= \omega_{aon}^{loc} \cdot PC_AON^{loc}(P_s) \\ &+ \omega_{aon}^{glob} \cdot PC_AON^{glob}(P_s) \\ &+ \omega_{aon}^{rv} \cdot PC_RV(P_s) \\ &+ \omega_{aon}^{oop} \cdot PC_OOP(P_s) \end{aligned} \quad (5.39)$$

an gegebener Stelle erläutert.

⁸⁷Engl. *material flow*

⁸⁸Engl. *procurement*

⁸⁹ D für Demand

⁹⁰Abgeleitet von engl. *continuous replenishment*

⁹¹ P_s ist hier ein Beschaffungsplan zwischen zwei Objektknoten.

⁹² OOP für Out of Plan

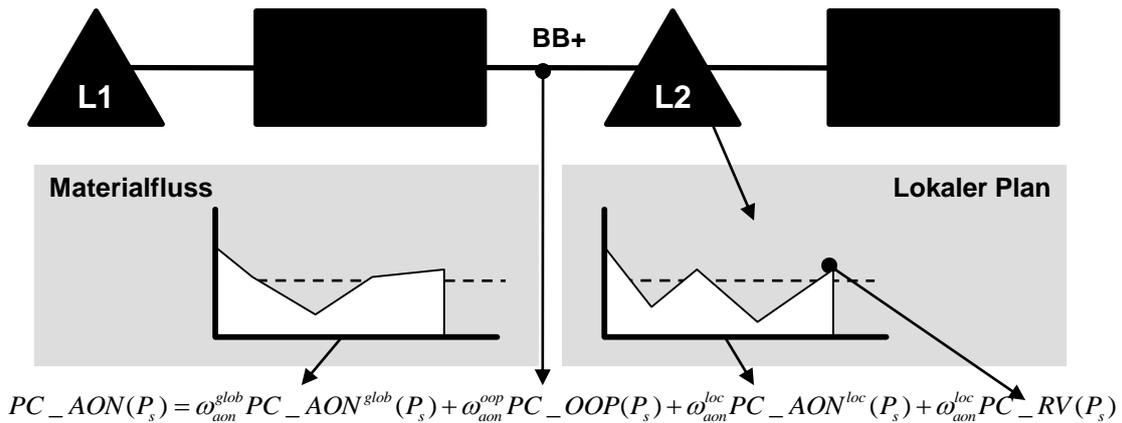


Abbildung 5.19.: Skizzierung der Strafkostenparameter am FOK

Für die Gewichte gilt $\omega_{aon}^{loc} + \omega_{aon}^{glob} + \omega_{aon}^{rv} + \omega_{aon}^{oop} = 1$. Die Gewichte dienen zur gezielten Bestrafung der Anwendung lokaler oder globaler Planungsstrategien während des Lernprozesses, wie Abbildung 5.19 zeigt.⁹³ Der Plan P_s mit seinen Perioden $p(k)^{mf}, k \in 1, \dots, PH$ beschreibt den Materialfluss des Beschaffungsprozesses. Für Beschaffungspläne werden um eine Vorlaufzeit verschobene Perioden $p(b)$ als *Beschaffungszeitpunkte* bezeichnet. Für die Perioden $p(b)^{mf} \in PHZ$ in denen keine Beschaffungsprozesse stattfinden gilt $p(b) = 0$.

Für die Berechnung der Beschaffungszeitpunkte muss eine Vorlaufzeitverschiebung LTS ⁹⁴ berücksichtigt werden, die vom ursprünglichen Bedarfszeitpunkt abgezogen wird.⁹⁵ Da die Perioden durch den Diskontfaktor DF gewichtet werden, wird dadurch im Rahmen der Strafkostenberechnung ausgedrückt, dass gerade kurzfristige bedarfsorientierte Beschaffungsprozesse teurer sind. Der Bedarfszeitpunkt der angefragten Fertigungsstufe ist als spätestster Endzeitpunkt SEZ einer nachgelagerten Stufe um LTS rückwärts umzeterminieren:

$$p(b) = p(k - LTS), \quad k - LTS > 0 \quad (5.40)$$

Da $p(b)$ nicht hinter der Heutelinie, ergo in der Vergangenheit, liegen kann, darf im Falle $(k - LTS) < 0$ die Anfrage nicht mehr koordiniert werden, da diese Koordination in jedem Fall zu einer Ablehnung führen wird. Abbildung 5.20 verdeutlicht die Wirkungsweise von $p(b)$ grafisch.

⁹³Der KOK ist bewusst nicht eingezeichnet und wird später gesondert als ausgehend vom anfragenden FOK betrachtet.

⁹⁴Engl. *leadtime shift*

⁹⁵Vgl. Kap. 2.1.2, S. 24

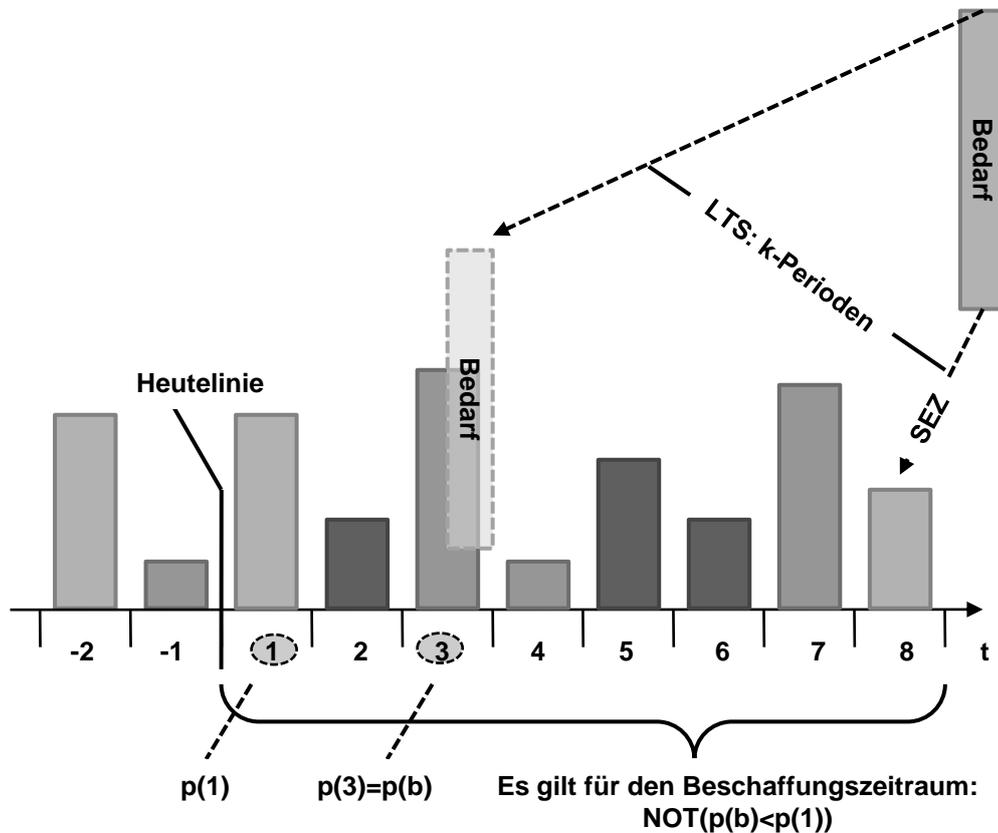


Abbildung 5.20.: Zusammenhang von Bedarfstermin, LTS und $p(b)$

5.2.9.2. Leistungsvereinbarungen im Beschaffungsprozess

Über die Festlegung von Leistungsvereinbarungen können die Beschaffungsstrategien, Bestellpunkt- und Bestellzyklusverfahren, unterschieden werden.⁹⁶ Für die globale Änderungsplanung bedeutet dieses, dass Beschaffungsprozesse globaler Aktionen zusätzlich zu den regulär stattfindenden Beschaffungen im Rahmen der Änderungsplanung zum Ausgleich ungleicher Zustände notwendig werden. Durch diese Verfahren werden entweder Bedarfe oder Angebote in Beschaffungsperioden entsprechend der Leistungsvereinbarungen gesenkt oder erhöht, oder es fallen zusätzliche Bedarfe oder Angebote in weiteren Planungsperioden an.

Das Ziel von Leistungsvereinbarungen ist, Beschaffungszeitpunkte oder Vereinbarungen über Beschaffungsmengen festzulegen, um Beschaffungskosten zu senken und die Planungssicherheit der beteiligten Partner zu erhöhen.⁹⁷ Diese Logik wird hier aufgegriffen und bei der Strafkostenberechnung berücksichtigt. Für gültige Beschaffungspläne, also Pläne für deren Beschaffungsprozesse die angefragten Materialmengen innerhalb

⁹⁶Siehe Kap. 2.1

⁹⁷Siehe Kap. 2.1.2, S. 17 ff.

der Leistungsvereinbarungen liegen, fallen keine Strafkosten an. Wurden Restriktionsgrenzen verletzt, z. B. im Falle einer unzulässigen Bedarfserhöhung innerhalb eines Bestellpunktverfahrens über ein vereinbartes Maximum, so fallen Strafkosten an. Für Beschaffungen entstehen außerhalb von Bestellzyklen Strafkosten. Je nach Ausgestaltung der Bestellpolitik der jeweiligen Beschaffungssteuerung unterscheiden sich die einzelnen Kostenbewertungen.

Für die Verletzung von Leistungsvereinbarungen im Beschaffungsprozess werden zusätzliche Kosten veranschlagt.⁹⁸ Sei NUM_MF_{all} die Anzahl aller Beschaffungsprozesse, NUM_MF_{raise} die Anzahl geänderter zu den ursprünglich vereinbarten Beschaffungsprozessen und NUM_MF_{out} die Anzahl aller nicht vereinbarten Prozesse. Es ergibt sich ein Verhältnis zwischen vereinbarten und nicht vereinbarten Beschaffungsprozessen eines Planes durch:

$$PC_OOP(P_s) = \frac{(NUM_MF_{out} + NUM_MF_{raise})}{NUM_MF_{all}} \quad (5.41)$$

Wurde in den Leistungsvereinbarungen keine spezielle Beschaffungssteuerung festgelegt, so sind Beschaffungsmenge und Beschaffungszeit variabel und es wird die (s, S) -Politik verwendet. Dann fallen die vollen Beschaffungskosten in jeder Beschaffungsperiode an. Die folgende Diskussion widmet sich den Bestellpunktverfahren mit der (s, q) - und (s, S) -Politik und den Bestellrhythmusverfahren mit der (t, S) -Politik.⁹⁹

5.2.9.3. Strafkosten bei Bestellpunktverfahren

Beim Bestellpunktverfahren können Bedarfe in jeder Periode geändert bzw. neu angefragt werden. Der angefragte Bedarfstermin darf nach Berücksichtigung einer etwaigen LTS nicht in der Vergangenheit liegen.¹⁰⁰

Zwischen zwei Objektknoten findet im Rahmen einer bestellpunktbasierter Koordination ein dedizierter Materialfluss zur Deckung der Bedarfe statt. Analog zu den oben skizzierten Strafkostenberechnungsvarianten eignet sich die Höhe des Materialflusses als Basis des Strafkostenmodells. Als Restriktionen für den Materialfluss wurden folgende Kennzahlen definiert:¹⁰¹

⁹⁸Siehe Formel (5.39)

⁹⁹Siehe [Sch05], S. 397 ff. Die Variablen bedeuten: $t \rightarrow$ Bestellzyklus, $q \rightarrow$ Bestellmenge, $s \rightarrow$ Bestellpunkt, $S \rightarrow$ Sollbestand.

¹⁰⁰Siehe Abb. 5.20

¹⁰¹Siehe Kap. 2.1.2, S. 17

5. Konzeption

$p(b)_{max}^{mf}$	Maximaler lieferantenspezifischer Materialfluss
$p(b)_{min}^{mf}$	Minimaler lieferantenspezifischer Materialfluss
$p(b)_{absmax}^{mf}$	Obergrenze des Materialflusses zwischen zwei Objektknoten
$p(b)^{mf}$	Materialfluss zwischen Kunden und Lieferanten

Um Restriktionen in der Kostenfunktion vergleichbar berücksichtigen zu können, muss normiert werden. Hier wird analog zu den oben eingeführten lokalen Strafkosten eine Aufteilung in die drei Strafkostenquotienten vorgenommen:

- $PC_AON_{max}^{proc}(P_s)$ Strafkosten für die Überschreitung einer Leistungsvereinbarung im bestellpunktbasierten Beschaffungsprozess
- $PC_AON_{min}^{proc}(P_s)$ Strafkosten für die Unterschreitung einer Leistungsvereinbarung im bestellpunktbasierten Beschaffungsprozess
- $PC_AON_{cost}^{proc}(P_s)$ Zusatzkosten bei einem Beschaffungsprozess außerhalb der Bestellpunktvereinbarungen und bei Beschaffungen höher als die vereinbarte Bestellmenge

Daraus ergeben sich als Strafkosten für die einzelnen Quotienten je Periode:

$$PC_AON_{min}^{proc}(p(b)) = \omega_{min}^{proc} \min \left(1, \left[\frac{\max(0; p(k)_{min}^{mf} - p(b)^{mf}}{p(k)_{min}^{mf}} \right] \right) \quad (5.42)$$

$$PC_AON_{max}^{proc}(p(b)) = \omega_{max}^{proc} \min \left(1, \left[\frac{\max(0; p(b)^{mf} - p(b)_{max}^{mf}}{(p(b)_{absmax}^{mf} - p(b)_{max}^{mf})} \right] \right) \quad (5.43)$$

$$PC_AON_{cost}^{proc}(p(b)) = \omega_{cost}^{proc} \left(\frac{p(b)^{mf}}{p(b)_{absmax}^{mf}} \right) \quad (5.44)$$

$$\text{mit } \omega_{max}^{proc}, \omega_{min}^{proc}, \omega_{cost}^{proc} \in [0, 1]$$

Zur Normierung gilt für die Konfiguration der Gewichtungsfaktoren:

$$\omega_{max}^{proc} + \omega_{min}^{proc} + \omega_{cost}^{proc} = 1 \quad (5.45)$$

Bei (s, q) -Politik legt $PC_AON_{min}^{proc}(p(b))$ die vereinbarte fixe Bestellmenge fest. Eine Abweichung führt zur Bestrafung mit vollen Kosten. Bei der (s, S) -Politik fallen die vollen Strafkosten an. Es ergeben sich die gesamten Strafkosten für einen bestellpunkt-basierten Beschaffungsprozess $PC_AON^{proc}(p(b))$ am FOK aus:

$$PC_AON^{proc}(p(b)) = \begin{cases} PC_AON_{min}^{proc}(p(b)) \\ + PC_AON_{cost}^{proc}(p(b)) \\ + PC_AON_{max}^{proc}(p(b)) & \text{für } p(b)^{mf} \neq p(b)_{min}^{mf} \\ 0 & \text{sonst} \end{cases} \quad (5.46)$$

Die gesamten Strafkosten $PC_CON^{proc}(P_s)$ eines Planes P bei Anwendung des Bestellpunktverfahrens am FOK berechnen durch die Kumulation der um DF^{102} je Periode $p(b)$ abgezinsten Strafkosten¹⁰³:

$$PC_AON^{proc}(P_s) = \sum_{b=1}^{PH} DF(p(b)) \cdot PC_AON^{proc}(p(b)) \quad (5.47)$$

5.2.9.4. Strafkosten bei Bestellzyklusverfahren

Die Beschaffung von Materialien nach Bestellzyklusverfahren zeichnet sich dadurch aus, dass Bedarfe in einem regelmäßigen Rhythmus oder Zyklus, z. B. alle drei Tage, jede Woche oder jeden Monat, beim Lieferanten eingestellt werden. Als Varianten werden nach der (t, q) -Politik feste Menge zu festen Zyklen oder nach der (t, S) -Politik variable Mengen zu festen Zyklen vereinbart.

Für Bestellzyklusverfahren werden die gleichen Strafkostenberechnungsquotienten wie für Bestellpunktverfahren angewendet:

$$PC_AON_{min}^{crep}(p(b)) = \omega_{min}^{crep} \min \left(1, \left[\frac{\max(0; p(b)_{min}^{mf} - p(b)^{mf})}{p(b)_{min}^{mf}} \right] \right) \quad (5.48)$$

$$PC_AON_{max}^{crep}(p(b)) = \omega_{max}^{crep} \min \left(1, \left[\frac{\max(0; p(b)^{mf} - p(b)_{max}^{mf})}{(p(b)_{absmax}^{mf} - p(b)_{max}^{mf})} \right] \right) \quad (5.49)$$

$$PC_AON_{cost}^{crep}(p(b)) = \omega_{cost}^{crep} \frac{p(b)^{mf}}{p(b)_{absmax}^{mf}} \quad (5.50)$$

$$\text{mit } \omega_{max}^{crep}, \omega_{min}^{crep}, \omega_{cost}^{crep} \in [0, 1]$$

Für die Konfiguration der Gewichtungsfaktoren gilt zur Normierung:

$$\omega_{max}^{crep} + \omega_{min}^{crep} + \omega_{cost}^{crep} = 1 \quad (5.51)$$

Die Summe der Strafkosten je Periode liefert:

$$PC_AON^{crep}(p(b)) = PC_AON_{min}^{crep}(p(b)) \quad (5.52)$$

$$+ PC_AON_{cost}^{crep}(p(b))$$

$$+ PC_AON_{max}^{crep}(p(b))$$

$$(5.53)$$

¹⁰²Siehe Formel (5.20)

¹⁰³Restriktionsverletzungen werden nicht zusätzlich bestraft, da die Bestrafung für Beschaffungsprozesse bereits Beschaffungsperioden außerhalb der Leistungsvereinbarungen bestraft.

Für die Anwendung der Strafkostenberechnung müssen verschiedene Fälle unterschieden werden. Es fallen Strafkosten an, wenn außerhalb eines Bestellzyklus beschafft wird und wenn die minimale Bestellmenge $p(b)_{min}^{mf}$ unterschritten wird. Findet keine Beschaffung statt, so gilt $PC_AON^{crep}(p(b)) = 0$ für die entsprechende Periode. Für die gesamten Strafkosten $PC_CON^{crep}(P_s)$ ergibt sich unter Berücksichtigung der diskutierten Fälle für einen Plan:

$$PC_AON^{crep}(P_s) = \sum_{b=1}^{PH} DF(p(b)) \cdot PC_AON^{crep}(p(b)) \quad (5.54)$$

5.2.9.5. Übertragung der Konzepte auf die angebotsseitige Koordination

Im Gegensatz zur Bedarfskoordination läuft der Koordinationsprozess bei der Angebotskoordination vorwärts. Die Vorlaufzeitverschiebung muss bei der Berechnung der geänderten Angebotsperioden in Richtung Zukunft im Planungshorizont addiert werden. Zur Berechnung des Angebotszeitpunktes gilt:

$$p(b) = p(k + LTS), \quad k + LTS < PH \quad (5.55)$$

Alle weiteren Berechnungen erfolgen analog zu denen bei der bedarfsseitigen Koordination.

5.2.9.6. Koordination zwischen FOK/KOK und FOK/FOK

Die Abstimmung globaler Koordinationsprozesse zwischen FOK/KOK und FOK/FOK erfolgt über den PK.¹⁰⁴ Dessen Belegung resultiert aus den geplanten Zu- und Abgängen der FOK und den dazu bereitstehenden Kapazitäten. Ein FOK kann nur so viel Material in den Produktionsprozess liefern, wie Kapazität der verfügbaren Betriebsmittel bereitgestellt wird.

Die Koordination zwischen FOK und KOK ist immer bei einer Angebots- oder Bedarfsänderung am Zu- oder Abgang eines FOK erforderlich. Diese Koordination von benötigten Kapazitätsbedarfen entfällt bei einer Planung gegen unendliche Kapazitäten am KOK. Werden aber Kapazitätsrestriktionen berücksichtigt, muss für die Änderung der potenziellen Belegung am PK durch erhöhte oder gesenkte Zu- und Abgänge an einem FOK eine entsprechende Anfrage zur Erhöhung oder Senkung des Kapazitätsangebotes zwischen FOK und KOK über den PK durchgeführt werden. Diese Anfrage erfolgt über einen kooperativen Änderungsplanungsprozess und daher bevor

¹⁰⁴Vgl. Kap. 2.1.2, S. 13 ff.

eine Bedarfs- oder Angebotsänderung an angrenzende Kunden oder Lieferanten weitergegeben wird.¹⁰⁵ Eine Anfrage an einen KOK wird entsprechend verfügbarer Kapazitäten am KOK beantwortet. Dabei folgt einer Anfrage durch einen FOK als Antwort durch den KOK eine:

1. Bestätigung der Anfrage, wenn genügend Kapazität vorhanden ist.
2. Bestätigung eines Gegenvorschlages, wenn die Anfrage das Kapazitätsangebot über- oder die Minimalauslastung unterschreitet.
3. Ablehnung ohne Gegenvorschlag, wenn der volle Leistungsgrad erschöpft oder der Minimalauslastung erreicht ist.

Eine Anfrage wird entsprechend dem Modell der Fertigung an einen oder mehrere alternative KOK übermittelt, wobei die Antwort kumulativ über alle angefragten KOK für die weitere Koordination berücksichtigt wird. Die objektnotenindividuellen Anfragen werden zunächst in den Plan des KOK eingerechnet. Folgt ein gültiger Zustand, so kann die vollständige Anfrage der Kapazität bestätigt werden. Sonst findet am KOK ein Änderungsplanungsprozess statt.¹⁰⁶ Ist es danach nicht möglich die vollständige Menge an Kapazität bereitzustellen, so wird die zur Verfügung stehende Menge als Gegenvorschlag bestätigt.

Die Koordination zwischen FOK und FOK erfolgt nach Bestätigung einer Leistungsgradänderung durch den betroffenen KOK. Es wird die bestätigte Menge bei Kunden oder Lieferanten angefragt. Der gesamte Ablauf der Koordination zwischen FOK/KOK und FOK/FOK richtet sich nach der koordinativen Änderungsplanung von Heidenreich.¹⁰⁷ Im Rahmen des Lernprozesses werden Änderungsplanungsprozesse durch die Rewardfunktionen bewertet und so entsprechende Regeln zur Steuerung der Änderungsplanung gelernt.

5.2.9.7. Bewertung globaler Beschaffungsprozesse durch lokal berechnete Strafkosten

Für die Berechnung der Strafkosten am FOK gilt, dass sowohl bei der Anwendung lokaler als auch globaler Planungsverfahren die beiden Strafkostenarten PC_AON^{loc} und PC_AON^{proc} bzw. PC_AON^{crep} zu berücksichtigen sind. Bei der Anwendung lokaler Planungsverfahren gilt $PC_AON^{proc} = 0$ bzw. $PC_AON^{crep} = 0$. Es fallen keine globalen Strafkosten an und es ist keine Beschaffung durchzuführen. Die Gewichtung der Strafkosten für lokale wie globale Planungsverfahren kann über die Gewichte ω_{loc}^{aon} bzw. ω_{glob}^{aon} gesteuert werden.

¹⁰⁵Siehe Kap. 2.2.4, S. 29 ff. und dort auch Abb. 2.6

¹⁰⁶Dieser Änderungsplanungsprozess wird beim Training des Lernsystems in eine Lernepisode eingebunden.

¹⁰⁷Siehe [Hei06] oder Kap. 2.1.3

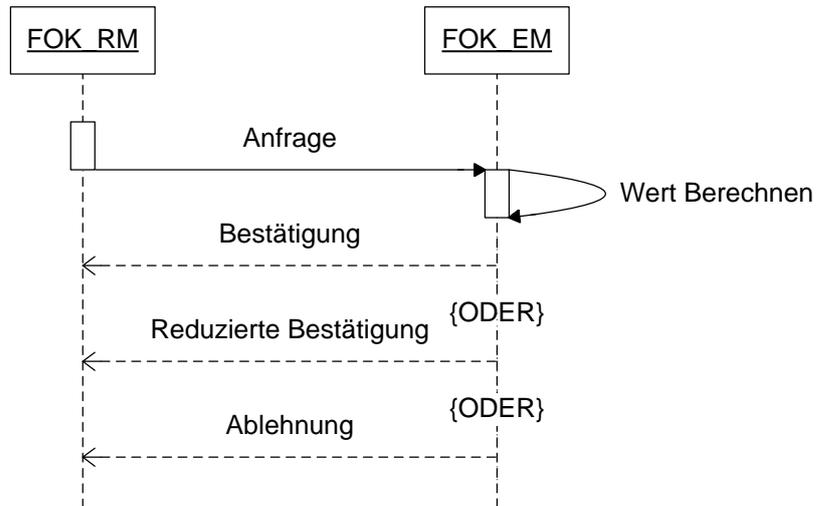


Abbildung 5.21.: Sequenz: Vereinfachtes Beispiel der Varianten der paarweisen Koordination in Sequenz zwischen Kunde (FOK_EM) und Lieferant (FOK_RM)

Bei der Berechnung von Strafkosten einer globalen Koordination zwischen *einem* Kunden und *einem* Lieferanten¹⁰⁸ sind im Rahmen des Koordinationsprotokolls bedarfs- als auch angebotsseitig drei Kommunikationsabläufe zu unterscheiden:¹⁰⁹

1. Anfrage → Bestätigung der Anfrage
2. Anfrage → Bestätigung eines reduzierten Gegenvorschlages
3. Anfrage → Ablehnung ohne Gegenvorschlag

Die Höhe der Anfrage richtet sich nach der Höhe der Bestätigung durch den KOK.¹¹⁰

Die gesamten Strafkosten für Beschaffungsprozesse werden im Weiteren durch die in Formel (5.39)¹¹¹ dargestellte Funktion berechnet. Je nach Art der Beschaffungsstrategie wird zur Strafkostenberechnung $PC_AON^{proc}(p(b))$ oder $PC_AON^{crep}(p(b))$ verwendet. Im Fall einer Bestätigung der Anfrage werden entsprechend der Menge des Materialflusses die vollen Strafkosten berechnet.

Durch die Summation von lokalen und globalen Strafkosten für Beschaffungsprozesse treten analytisch betrachtet, je nach Gewichtung der jeweiligen Strafkosten in Formel (5.39), folgende Effekte ein:

¹⁰⁸Es werden einzelnen Strafkosten ermittelt.

¹⁰⁹Siehe Abb. 5.21 und Kap. 2.1.3

¹¹⁰Vgl. Kap. 5.2.9.6

¹¹¹Siehe Seite 120

- Sind für die Beschaffungsprozesse die lokalen Strafkosten höher gewichtet als die globalen, so wird die Beschaffung gegenüber lokalen Umplanungen bevorzugt.
- Sind die lokalen Strafkosten geringer gewichtet als die globalen, wird der Beschaffungsprozess während des Lernprozesses verstärkt bestraft und lokale Verfahren bevorzugt.

Zu den Grundkosten des Beschaffungsprozesses selbst können separate Kosten durch Verletzungen globaler Leistungsvereinbarungen hinzukommen. Es wird die Anzahl nicht vereinbarter Beschaffungsprozesse im Verhältnis zu allen durchgeführten Beschaffungsprozessen, ähnlich wie bei den Restriktionsverletzungen, über den Faktor $PC_OOP(P)$ bewertet.¹¹²

Vorteil der Zusammenführung lokaler und globaler Strafkosten

Bei der Berechnung globaler Strafkosten scheint der reduzierte Gegenvorschlag eine kontraproduktive Wirkung auf die Gesamtstrafkostenberechnung zu entfalten, da der Materialfluss reduziert wird und so die Strafkosten für die Beschaffung entsprechend geringer ausfallen. An diese Stelle zahlt sich die kombinierte Betrachtung von lokalen und globalen Strafkosten aus. Bei einem reduzierten Gegenvorschlag tritt genau der Effekt ein, dass Bedarfe oder Angebote des Initiators der Koordination nicht gedeckt werden. Da im Rahmen der Änderungsplanung nur so viel Material disponiert wird, wie zur Beseitigung eines ungültigen Planes notwendig ist, verbleibt beim Initiator im Fall einer Unterdeckung seiner Anfrage ein ungültiger Plan, da nicht genügend Material vorhanden ist.

Der Folgezustand enthält weiterhin Restriktionsverletzungen, deren Strafkosten im Reward und so auch im Q-Update im Q-Wert der ausgeführten Aktion verrechnet werden. Der Q-Wert einer Aktion, die bei der Auflösung eines ungültigen Zustandes selten erfolgreich ist, steigt nicht so schnell wie z. B. der Q-Wert einer Aktion, deren Ausführung zumeist problemlos durchgeführt werden konnte. Durch diesen Effekt können insbesondere Regeln zur Auswahl von Lieferanten oder Kunden für spezifische Bedarfs- oder Angebotsänderungen gelernt werden, da während des Lernprozesses der zuverlässigere Lieferant auf Dauer den besseren Q-Wert aufweist.

5.2.9.8. Globale Koordination mit mehreren Partnern

Die bisherige Betrachtung bei der globalen Strafkostenberechnung bezog sich auf ein 1 : 1-Verhältnis zwischen Kunde und Lieferant. In dieser Arbeit sind 1 : n Verhältnisse zwischen Kunde und Lieferanten sowie vice versa zugelassen. Aus Formel (5.8)¹¹³

¹¹²Siehe Formel (5.41), S. 123

¹¹³Siehe S. 101

geht hervor, dass die Bedarfs- oder Angebotsmengen hier z. B. gemittelt über die einzelnen Objektknoten aufgeteilt werden. Angewendet auf die Strafkostenberechnung bedeutet dieses, dass der Initiator der Koordination jeweils alle entsprechenden Kunden respektive Lieferanten *einzel*n für die berechnete Menge anfragen muss. Es ergibt sich für die Strafkostenberechnung das Problem, dass die Bestätigungen oder Ablehnungen der Anfragen einzeln zurückgesendet wurden und jeweils je Anfrage einzelne Strafkosten berechnet werden müssen. Eine sequenzielle Berechnung der Strafkosten führt zu einer Verzerrung des Gesamtrwards, da sowohl die Akzeptanz als auch die Ablehnung einer einzelnen Anfrage nicht die gesamten entstehenden Strafkosten repräsentiert. Diese sind höher als bei einer 1:1-Koordination mit bestätigter Anfrage.

Die erste mögliche Lösungsvariante dieses Problems wäre die Summation der einzelnen Strafkosten zu einer Gesamtsumme von Strafkosten. Dieses ist nicht gangbar, da die Summe der einzelnen Strafkosten so höher wäre als bei einer 1:1-Koordination und das Lernergebnis im Hinblick auf die vergleichbare Bewertung von Aktionen beeinflussen würde.

Die zweite Lösungsvariante folgt dem bisherigen Gedanken der Arbeit: Die globale Koordination mit mehreren Partnern wird als eigenständige Aktion betrachtet und entsprechend durch das Lernsystem mit einem Q -Wert versehen und gelernt. Da die Aktion der verteilten Anfrage als *eigenständige* Aktion vom Lernsystem verwaltet wird, sollen die Strafkosten über die kumulierten Bestätigungen der Anfragen einer solchen Aktion berechnet werden. Hierzu wird die Summe der bestätigten Angebote oder Bedarfe in den lokalen Plan des anfragenden Objektknotens eingerechnet und entsprechend werden die Strafkosten gebildet. Beispiel 5.2 und Abbildung 5.22 illustrieren dieses.¹¹⁴

Beispiel 5.2 *Ein Objektknoten sendet in (1) eine Anfrage von jeweils 10 ST an drei Lieferanten L_1, L_2, L_3 . L_1 sendet eine Bestätigung, L_2 sendet einen Gegenvorschlag von 5 ST und L_3 eine Ablehnung. Daraus ergibt sich die in (2) dargestellte Menge von 15 ST, welche lokal eingeplant werden können. Durch die Planungsergebnisse wird der verbleibende Bedarf und nachfolgend der Reward berechnet.*

5.2.10. Bewertung eines Endzustandes

Bei der Aktualisierung des Q -wertes¹¹⁵ eines Zustandes P_s wird jeweils der maximale Q -Wert des Folgezustandes P_{s+1} mit eingerechnet. Unter Verwendung des Diskontfaktors γ werden die Q -Werte des Aktionspfades vom Anfang- zum Endzustand sukzessive in Abhängigkeit vom Q -Wert des Endzustandes berechnet.¹¹⁶ Da die Strafkosten

¹¹⁴Hier nicht geziegt, aber vor jeder globalen Koordination wird eine Anfrage an den betroffenen KOK gesendet, deren Antwort die Höhe der möglichen Änderungsanfragen bestimmt (Siehe Kap. 5.2.9.6.

¹¹⁵Siehe Formel (5.9), S. 103

¹¹⁶Siehe Kap. 5.2.1, S. 100

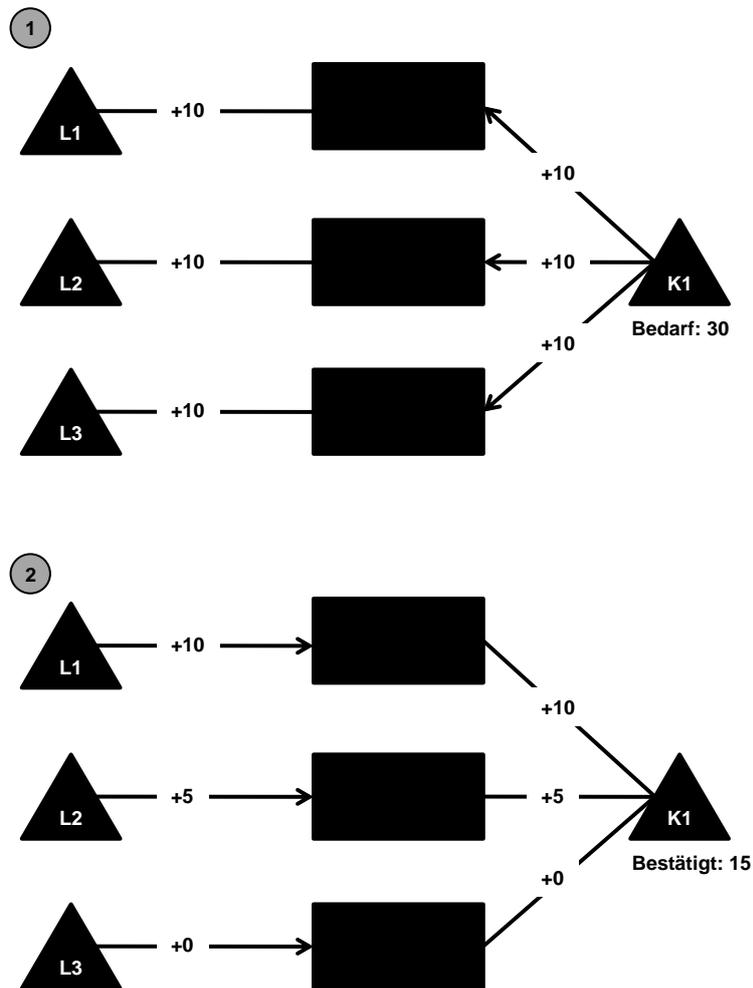


Abbildung 5.22.: Darstellung des Problems global verteilter Anfragen in der Strafkostenberechnung

in dieser Arbeit auf einen Wert im Intervall $[0, 1]$ normiert sind, ist der maximal zu erwartende Reward 1. Der maximale Q -Wert konvergiert gegen 1. Für Endzustände des Lernprozesses, also für gültige Planzustände, gilt $\max Q = 1$.

5.2.11. Anmerkung zum Q-Update auf Clusterebene

Das Update des Q -Wertes erfolgt über die Rewardberechnung zwischen charakteristischen Plänen einzelner Cluster, stellvertretend für die dedizierten Pläne der Zustände.¹¹⁷ Hierdurch kann der Abstraktionsmechanismus effektiv im Training verwendet

¹¹⁷Siehe Abb. 5.12, S. 102

und das Training effizient umgesetzt werden. Dabei tritt das Problem auf, dass die charakteristischen Pläne durch das Clustering als normierte Pläne vorliegen.¹¹⁸

Aus diesem Grund müssen im Falle der Rewardberechnung aus den normierten Plänen die diskreten Pläne zurückgerechnet werden. Durch das Umformen der zur Normalisierung verwendeten Formel (5.1)¹¹⁹ unter Verwendung der für den Cluster gültigen Restriktionsgrenzen ist es möglich, den entsprechenden Planwert zu berechnen.

$$p^*(k) = \frac{1}{p(k)_{max} - p(k)_{min}}(p(k) - p(k)_{min}) \quad (5.56)$$
$$\Leftrightarrow p(k) = p(k)^* \cdot (p(k)_{max} - p(k)_{min}) + p(k)_{min}$$

Die Normierung der Planwerte beim Clustering erfolgt stets in Bezug zur Standardrestriktionsgrenze des Planes *min* oder *max*. Während der Änderungsplanung werden Restriktionsgrenzen verschoben. Diese sollen bei der Rückrechnung insofern berücksichtigt werden, dass in dem Fall $p(k)_{min} = p(k)_{mintemp}$ und $p(k)_{max} = p(k)_{maxtemp}$ gilt.

5.2.12. Gesamtrewardfunktion

Die erarbeiteten Strafkostenfunktionen je Objekttyp oder bezogen auf die Koordinationsart können zur Rewardfunktion für das Q-Learning zusammengefasst werden. Dabei ermöglichen die anwendungs- und objekttypspezifischen Konfigurationsmöglichkeiten der Gewichtungsparmeter der Strafkostenfunktionen die Anpassung des Lernverfahrens an unterschiedliche Lernziele. Diese sind z. B. die Fokussierung auf die Anwendung lokaler Änderungsverfahren. Aus unternehmerischer Sicht wird hier das Ziel verfolgt, auftretende Probleme zunächst unternehmensintern zu lösen. Andersherum kann z. B. bei einem stark beschaffungsorientierten Unternehmen mit flexibler Lieferantenstruktur das Gewicht auf der Anwendung globaler Verfahren mit dem Ziel zur Auswahl spezifischer Lieferanten für Anfragen während der Änderungsplanung liegen.¹²⁰

Durch Formel (5.18) wurde festgelegt, dass der Reward einer Aktion, sei es für einen FOK oder KOK, als Differenz der berechneten gewichteten Strafkosten der Pläne vor und nach Ausführung der Aktion definiert wird. Die Rewardfunktion wird direkt durch die definierten Strafkostenfunktionen für FOK und KOK ausgestaltet.

¹¹⁸Siehe Kap. 5.1.2.1, S. 76 ff.

¹¹⁹Siehe S. 78

¹²⁰Vgl. Diskussion zur Lieferantenauswahl in Kap. 5.2.9.7, S. 129

Für FOK ergibt sich für einen Planverlauf P_s bei Anwendung eines Planungsverfahrens a_i ¹²¹ die Rewardfunktion:

$$R(P_s^{aon}, a_i) = PC_AON(P_s) - PC_AON(P_{s+1}) \quad (5.57)$$

Für KOK ergibt sich für einen Planverlauf P_s bei Anwendung eines Planungsverfahrens pa_j aus der Menge aller zugelassenen Planungsverfahren des KOK die Rewardfunktion:

$$R(P_s^{con}, pa_j) = PC_CON(P_s) - PC_CON(P_{s+1}) \quad (5.58)$$

Zur Stärkung oder Abschwächung einzelner Strafkostenfaktoren¹²² innerhalb der Rewardberechnung können die in den vorherigen Kapiteln dargestellten und über ϕ und ρ ausprägbar Gewichtungsfaktoren $\omega_\rho^\phi \in [0, 1]$ als Parameter definiert werden. Zur Konfiguration der Rewardberechnung sollten die Parameter in einer logischen Reihenfolge belegt werden. Als Erstes werden die allgemeinen Parameter festgelegt. Danach die Gewichtungparameter der objektnotenspezifischen Rewardfunktionsfaktoren und folgend die Gewichtungen der Rewardfunktionen untereinander.

5.3. Konzeption des Trainings, der Lernepisoden und der Regelgenerierung

Damit das Lernsystem die Q-Werte lernen kann, muss es den Anforderungen entsprechend trainiert werden. Das Training wird im Q-Learning allgemein als kontinuierlicher, quasi unendlicher Prozess betrachtet, der durch das Eintreten bestimmter Bedingungen oder durch den Benutzer zu beenden ist. Der Trainingsprozess besteht aus einer Sequenz von *Lernepisoden*, die in *Lernschritte* untergliedert ist.

Definition 5.2 (Lernschritt, Lernepisode, Training) *Als Lernepisode wird, ausgehend von definierten Ausgangszuständen, eine sich wiederholende Anwendung des Q-Learning-Algorithmus¹²³ im Rahmen einer kooperativen Änderungsplanung in einem Produktionsnetzwerk definiert. Jede stattfindende einzelne Aktualisierung eines Q-Wertes eines Objektnotens wird als Lernschritt bezeichnet. Ein Lernschritt für einen Objektnoten beginnt mit einem ungültigen Zustand und endet auf einem Folge- oder Endzustand. Die Gesamtheit der Sequenzen aller Lernepisoden wird als Training bezeichnet. Das Training endet bei Konvergenz des Lernverfahrens oder beim Erreichen einer definierten Abbruchbedingung.¹²⁴*

¹²¹Unabhängig davon, ob es sich um ein lokal oder global angewendetes Planungsverfahren handelt

¹²²Min-Bereich, Max-Bereich, Kosten

¹²³[Mit97]

¹²⁴Siehe auch Kap. 2.2.5, S. 37

Bezogen auf den Untersuchungsgegenstand wird das Training des Lernverfahrens als kontinuierliche Simulation von Änderungsplanungsprozessen mit eingebetteten Lernepisoden in einem Produktionsnetzwerk auf der Basis realer Ausgangsdaten verstanden. Je Lernschritt werden die durchgeführten Änderungsplanungsprozesse zur Auflösung ungültiger Zustände durch die objektknoten- bzw. koordinationspezifisch angewendete Rewardfunktion bewertet.¹²⁵ Zur Durchführung eines lernepisodenbasierten Trainings sind konzeptionell folgende Anforderungen zu erfüllen:

- Vorgabe eines Szenarios durch vollständig konfigurierte Objektknoten
- Vorgabe der Parameter der Rewardfunktion für jeden Objektknoten
- Vorgabe eines definierten Ausgangszustands der involvierten Objektknoten für jede Lernepisode
- Vorgabe einer oder mehrerer Abbruchbedingungen zur Beendigung des Trainingsprozesses
- Eine Funktion, welche die Änderungsplanungsverfahren, zufällig oder systematisch, auswählt

Nachdem der allgemeine Aufbau einer Lernepisode erläutert wurde, werden in den folgenden Kapiteln iterativ die oben aufgestellten allgemein formulierten Anforderungen aufgegriffen und diskutiert.¹²⁶

5.3.1. Lernepisoden und deren Ausgangsdaten

Die Lernepisode bildet den Rahmen oder ein Szenario im Training, indem einzelne Lernschritte durchgeführt werden. Jede Lernepisode besteht aus einer endlichen Anzahl von Lernschritten je Objektknoten. Zu Beginn einer Lernepisode werden den Plänen der Objektknoten Initialbelegungen zugewiesen. Diese sind reale, gültige oder auch ungültige Pläne des Produktionsnetzwerkes, extrahiert aus Realdaten z. B. von ERP-Systemen.¹²⁷ Die initialen Zustände einer Lernepisode werden als *Initialzustand* bezeichnet.

¹²⁵Siehe Kap. 5.2, S. 100 ff.

¹²⁶Die Konfiguration der Objektknoten und die Parametrisierung der Rewardfunktion wurden in den Kap. 2.3.2 auf Anforderungsebene und in Kap. 5.2.12 konzeptionell behandelt.

¹²⁷Diese Daten werden auch im Clustering benötigt. Daher vgl. Details zur Gewinnung von Realdaten in Kap. 5.1.3.8, S. 94 ff. Hier werden zusätzlich auch gültige Zustände benötigt, deren Datenstruktur aber von denen der ungültigen Zustände nicht abweicht.

5.3.1.1. Lernschritte am Objektknoten

Aus Sicht eines Objektknotens stellt sich ein Lernschritt im Rahmen einer Lernepisode als 4-stufiger Prozess dar, indem eine kooperative Änderungsplanung durch die Rewardfunktion bewertet wird. Der Reward wird wie Abbildung 5.12¹²⁸ dargestellt

Algorithmus 5.3 : Lernschritt

Eingabe : Ausgangszustand

Ausgabe : Endzustand

```
1 while NOT Endzustand do  
2   Änderungsplanungsverfahren auswählen  
3   Änderungsplanung durchführen  
4   Reward berechnen  
5   Q-Wert aktualisieren  
6 end
```

berechnet. Weitet sich die Änderungsplanung über mehrerer Knoten aus, so wird bei jedem weiteren Objektknoten ebenso eine Lernepisode gestartet, die erst beendet wird, wenn ein Endzustand erreicht wurde. Hierdurch wird erreicht, dass Zusammenhänge im Netzwerk durch Koordination zwischen den Objektknoten gelernt werden. Es kann zwischen bedarfsorientierten und angebotsorientierten Lernepisoden unterschieden werden. Abbildungen 5.23 und 5.24 und skizzieren beide Varianten.

5.3.1.2. Sequenz von Lernepisoden

Zu Beginn einer Lernepisode im Training des Lernverfahrens wird jedem Objektknoten zunächst ein Initialzustand zugewiesen. Liegt ein ungültiger Zustand vor, wird die Änderungsplanung angestoßen. Der zum Plan zugehörige Cluster wird ermittelt. Am betroffenen Objektknoten wird ein zugelassenes Änderungsplanungsverfahren des Clusters ausgewählt und die Strafkosten des Centroiden $C(k)$ gespeichert. Die Koordination wird durch eine Anfrage angestoßen, indem an die in Planungsrichtung benachbarte Fertigungsstufe eine Nachricht gesendet wird. Diese wird verrechnet und eine Antwort zurückgesendet. Diese enthält entweder die Bestätigung der Änderungsanfrage, einer Ablehnung der Änderungsanfrage oder einen Gegenvorschlag bzgl. der Änderungsanfrage.

Wurde im partizipierenden Objektknoten eine Restriktionsverletzung festgestellt, so startet die Koordination. Die endgültige Antwort wird in den Plan der jeweils anfragenden Objektknoten eingerechnet. Danach wird der Reward berechnet und der Q-Wert am Cluster des Nachfolgezustandes aktualisiert. Die Durchführung von Lernschritten

¹²⁸Siehe Abb. 5.12, S. 102

5. Konzeption

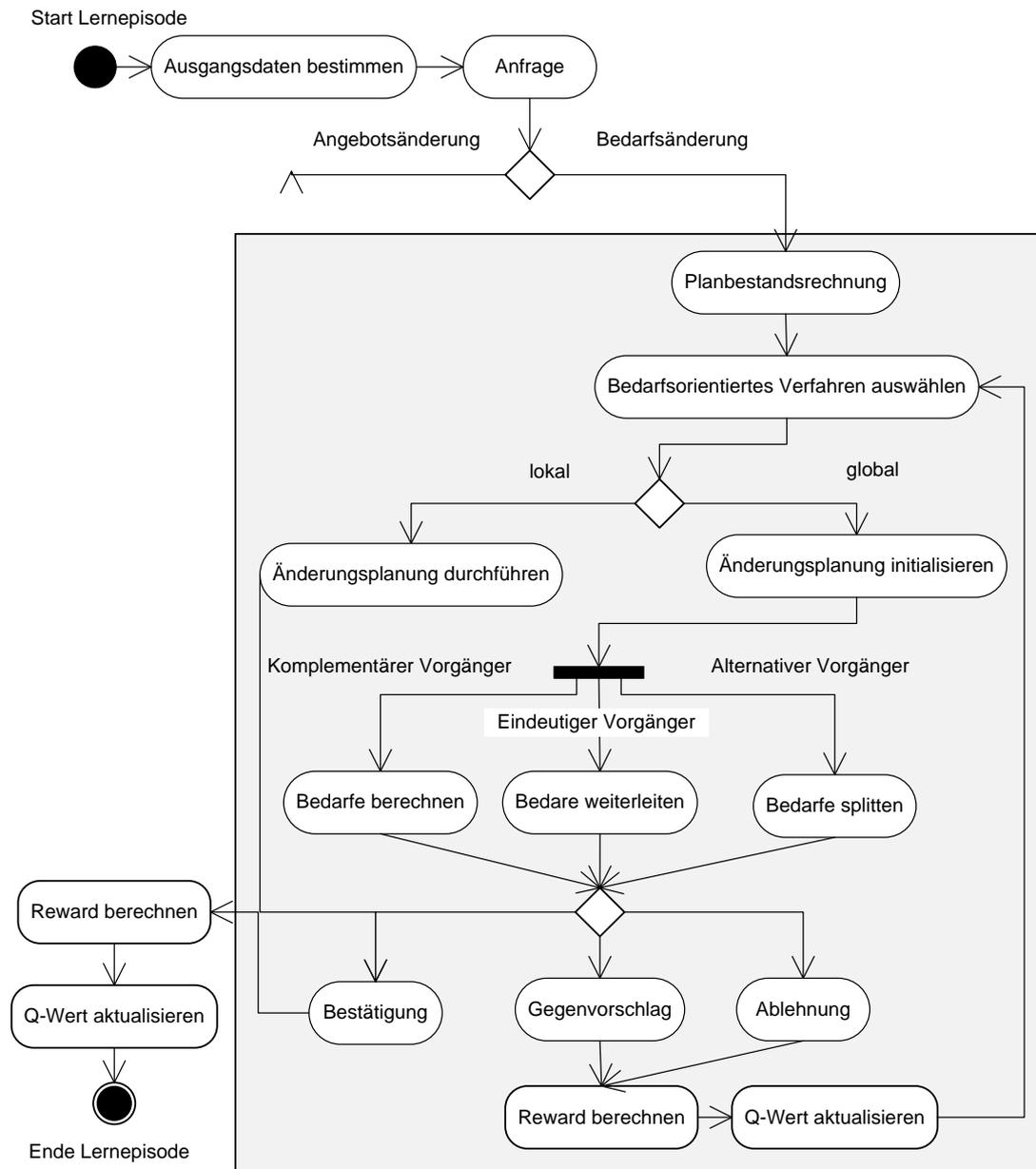


Abbildung 5.23.: Aktivität: Lernepisode (bedarfsorientiert)

5.3. Konzeption des Trainings, der Lernepisoden und der Regelgenerierung

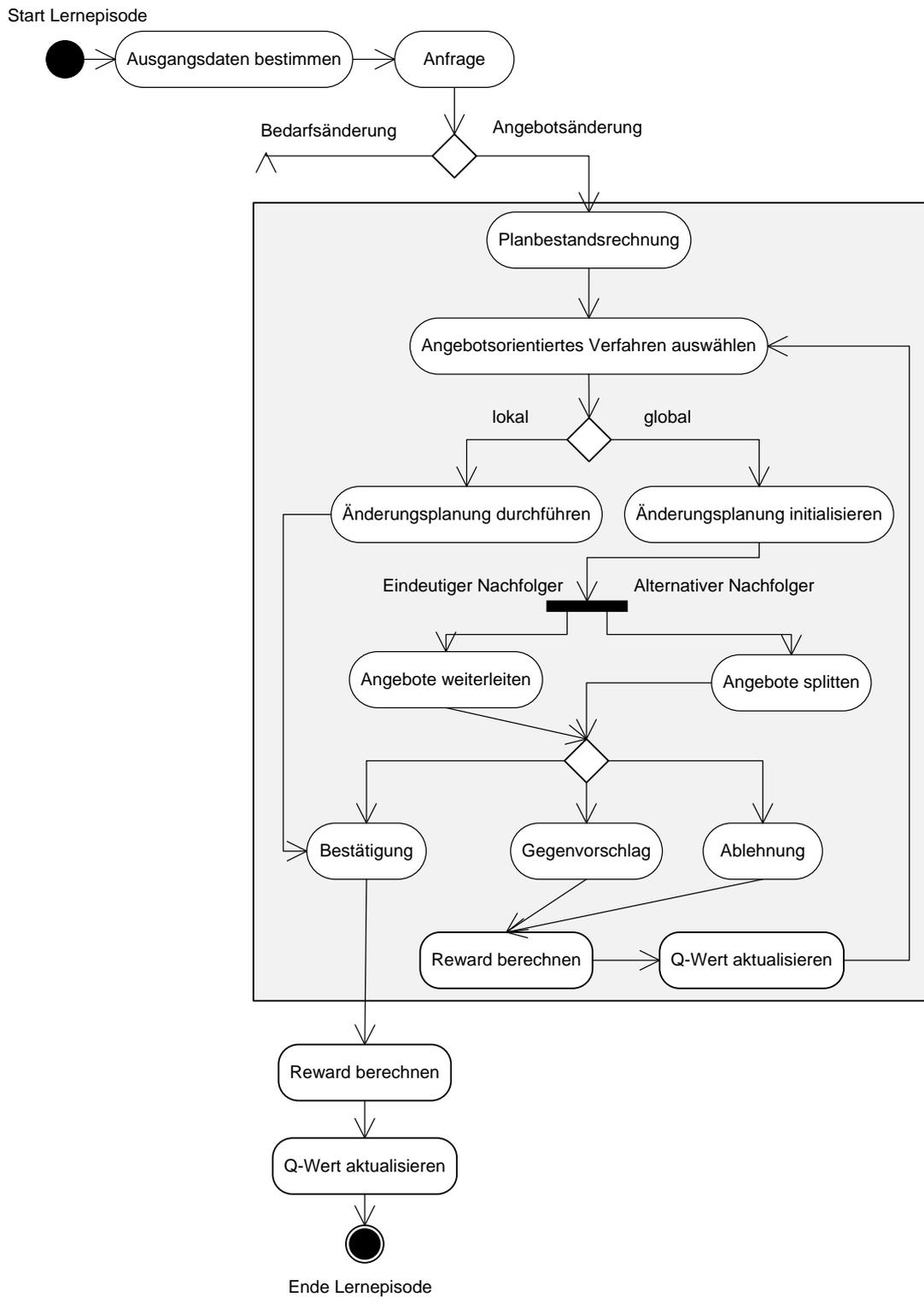


Abbildung 5.24.: Aktivität: Lernepisode (angebotsorientiert)

im Rahmen der koordinierten Änderungsplanung läuft so lange, bis alle Objektknoten im Produktionsnetzwerk einen gültigen Zustand erreicht haben oder eine definierte Abbruchbedingung eintritt. Wurde der letzte Objektknoten aktualisiert, ist die Lernepisode beendet. Es werden neue Initialdaten bereitgestellt.

5.3.1.3. Auswahl der Änderungsplanungsverfahren in einer Lernepisode

Die Auswahl einer Aktion im Training sollte einer bestimmten Strategie folgen, um effektiv möglichst viele Zustands-/Aktionspaare des Explorationsraumes bewerten zu können. Eine im Reinforcement-Learning weitverbreitete, sowie effizient zu implementierende Strategie zur Auswahl der nächsten Aktion ist die ϵ -Greedy Strategie¹²⁹

Eine reine *Greedy*-Strategie zeichnet sich dadurch aus, dass diejenige Aktion gewählt wird, die den größten Gewinn beim Zustandsübergang verspricht. Hier wäre dieses das Änderungsplanungsverfahren, welches mit hoher Wahrscheinlichkeit einen gültigen Plan mit minimalen Strafkosten erzielt. Dieses Vorgehen ist aber erst sinnvoll anwendbar, wenn eine optimale Strategie vorliegt. Die Greedy-Strategie dient daher zum Steuern des Prozesses, für den das Lernverfahren entwickelt wurde.

Zu Beginn des Lernprozesses ist es wenig hilfreich, eine Greedy-Strategie anzuwenden, da bei Verfolgung dieser Strategie stets die gleichen Aktionen aus einem Zustand heraus ausgeführt und so die Werte für andere Zustands-/Aktionspaare nicht mehr aktualisiert werden. Während des Trainings muss dafür gesorgt werden, dass zum betrachteten Zeitpunkt schlechter bewertete Aktionen mit einer bestimmten Wahrscheinlichkeit ausgeführt werden und der gesamte Zustands-/Aktionsraum berücksichtigt wird. Dies kann durch die Abwandlung einer reinen Greedy-Strategie erreicht werden, indem ein Parameter ϵ eingeführt wird, der bestimmt, mit welcher Wahrscheinlichkeit keine Greedy-Aktion, sondern eine zufällige Aktion, ausgeführt wird. Diese Strategie wird als ϵ -Greedy bezeichnet.

Der Wert ϵ liegt im Intervall $[0, 1]$, wobei ein Wert von 0 einer Auswahl der besten Aktion mit einer Wahrscheinlichkeit von 100 Prozent und einer Greedy-Strategie entspricht. Ein Wert von 1 beachtet die aktuellen Wertigkeiten der Aktionen nicht und wählt die nächsten Aktionen zufällig aus. Es wird häufig ein ϵ -Wert im Bereich von 0,1 gewählt, der die Greedy-Aktionen in 90 Prozent aller Fälle auswählt. Es bietet sich an, den Wert für ϵ zu Beginn hoch zu wählen, um viele Aktionen zu besuchen, im Laufe des Trainings zu senken, um die bis dato besten Aktionen noch zu verbessern und schließlich zum Zwecke der Steuerung auf 0 zu setzen, sodass die besten Aktionen gewählt werden. Wird eine reine Greedy-Strategie angewendet, so kann sich das System nicht mehr auf Änderungen einstellen und muss im Falle einer Änderung neu trainiert werden.

¹²⁹Bspw. in [Mit97]

ϵ -Greedy-Methoden werden aufgrund ihrer Einfachheit mit Erfolg in Reinforcement-Learning-Anwendungen eingesetzt.¹³⁰ Ein weiterer Vorteil dieser Methode ist, dass die Einstellung des Parameters ϵ der ϵ -Greedy-Methode für den Benutzer leichter nachvollziehbar ist. Ein Nachteil dieser Methoden besteht in der zufälligen Auswahl einer Aktion, da suboptimale Aktionen im Training durchgeführt werden können. Durch die ϵ -Greedy-Methode wird die Diversifikation der besuchten Zustände im Zustandsraum umgesetzt.

5.3.2. Funktionale Einbindung der Lernepisoden in das Training

Ziel des Trainingsprozesses ist es, eine fortlaufende Sequenz von Lernepisoden durchzuführen und so eine quasi beliebig lange Laufzeit des Trainings bis zum Erreichen einer Abbruchbedingung oder der Konvergenz des Lernsystems zu ermöglichen.

Für die einzelnen Lernepisoden muss im Rahmen der Änderungsplanung eine vollständige Koordination zwischen den beteiligten Objektknoten durchgeführt werden. Diese Koordination erfolgt nach dem definierten Ablauf der kooperativen Änderungsplanung.¹³¹ Bei jeder Art der erforderlichen Koordination wird beim KOK angefragt, ob genügend Kapazitäten vorhanden sind. Ist dieses nicht der Fall, erfolgt eine Ablehnung oder ein Gegenvorschlag. Zur Koordination werden die von Heidenreich vorgestellten Koordinationsprotokolle für die Änderungsplanung verwendet.¹³² Das Grundprinzip der Koordination wird in Abbildung 5.25 dargestellt.

5.3.2.1. Lernrate und Abbruchbedingungen

Während des Lernprozesses kann die Lernrate, d. h. der quantitative Einfluss einer Lernepisode auf die Q-Werte, durch den Lernfaktor α parametrisiert werden. Der Lernfaktor $\alpha \in [0, 1]$ gewichtet bei jedem Lernschritt, d. h. bei einer Aktualisierung eines Q-Wertes, den zu verrechnenden Reward.

Eine Lernepisode wird nach Definition 5.2¹³³ beendet, sobald ein gültiger Zustand nach Durchführung einer endlichen Anzahl von Lernschritten im Sinne der Änderungsplanung erreicht ist oder eine Abbruchbedingung zutrifft.

¹³⁰Zur Vertiefung sei auf [SB98], Kap. 2.2 verwiesen.

¹³¹Siehe Kap. 2.1.3, S. 18 ff.

¹³²Siehe [Hei06], S. 122-139

¹³³Siehe S. 133

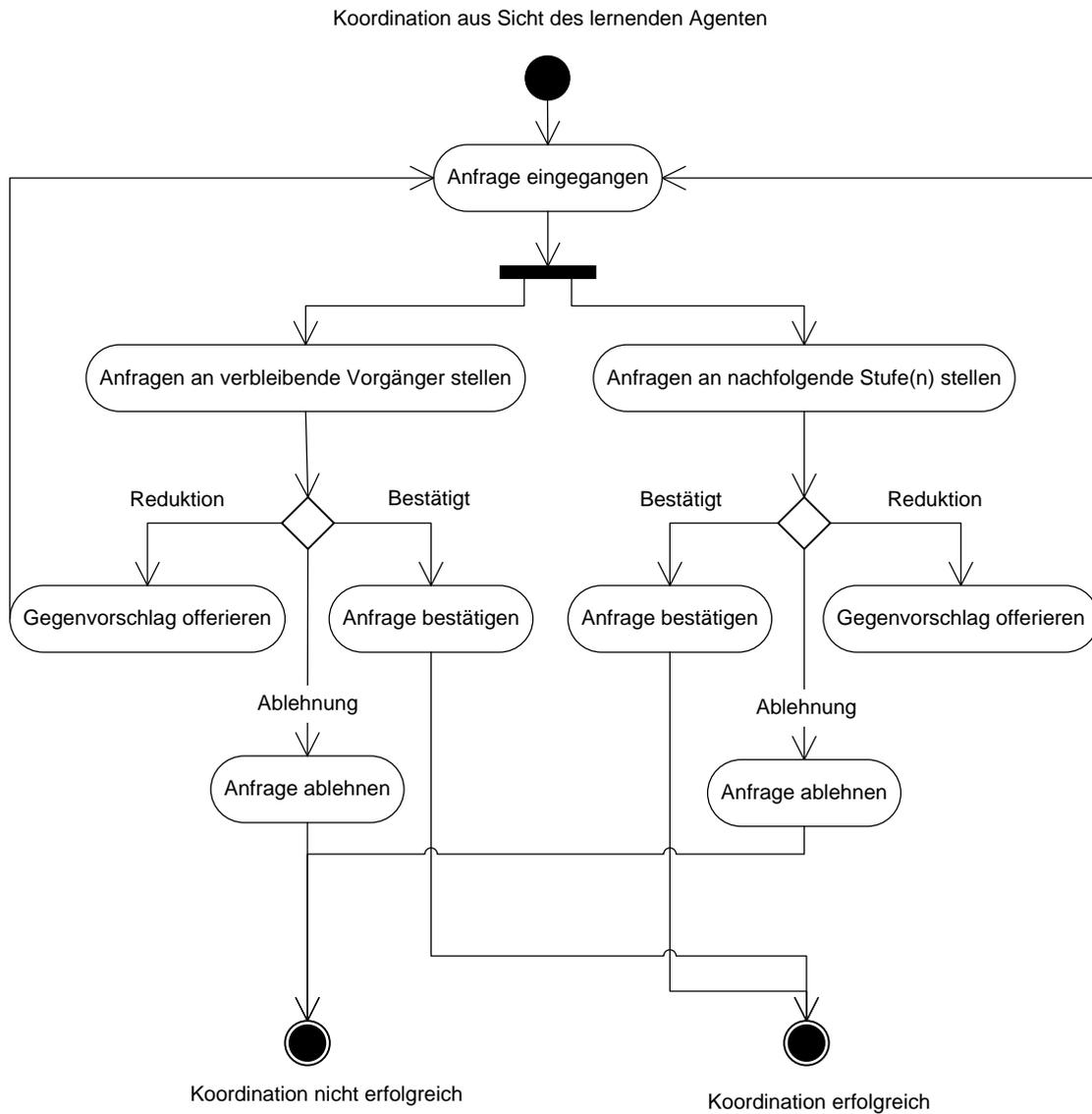


Abbildung 5.25.: Aktivität: Erweitertes Koordinationsprotokoll für Angebotsänderungen eines Knotens bei komplementären Zugängen im Nachfolgeknoten

Folgende elementaren Abbruchbedingungen sind konzeptuell sinnvoll:

1. *Abbruch nach Durchführung von n -Lernschritten ($abort_{ls}$):* beendet eine Lernepisode stets in gleicher Weise und ermöglicht ein zeitlich gleichmäßiges Durchsuchen des Explorationsraumes, sowie eine genaue Abschätzung der Trainingsdauer
2. *Abbruch nach x Ablehnungen während der Koordination ($abort_{ac}$):* verhindert ein Aufschaukeln von Angebot und Gegenvorschlag im Rahmen des Koordinationsprozesses der Änderungsplanung, falls kein gültiger Planzustand nach Durchführung von x Änderungen oder Ablehnungen in den Lernschritten erreicht wird
3. *Abbruch nach Unterschreitung eines prozentual anteiligen minimalen Rewards ausgehend vom letzten berechneten Reward bei Anwendung des derzeit Q -besten Planungsverfahrens für ein Cluster-/Aktionspaar ($abort_{ur}$):* deutet darauf hin, dass das Lernverfahren für diese Lernepisodenvariante des spezifischen Cluster-/Aktionspaares konvergiert und eine Neuinitialisierung und weiteres explorieren des Zustandsraumes sinnvoll ist
4. *Zufälliger Abbruch nach der Ziehung einer Zufallszahl in einem vorgegebenen Intervall ($abort_{rnd}$):* kann die Diversifikation der besuchten Zustände erhöhen und das Verlassen lokaler Zustandsbereiche im Explorationsraum ermöglichen
5. *Abbruch gesteuert durch die Anzahl der verarbeiteten Änderungen ($abort_{nc}$):* vermeidet oder erzwingt, je nach Ziel, extreme Zustände der Höhe und Art der Änderungen mit min/max-Restriktionsverletzungen und kann so die Exploration bestimmter Zustände fördern

Es ist möglich, die oben aufgestellten Abbruchbedingungen anwendungsfallsspezifisch disjunktiv zu konkatenieren. Eine Abbruchbedingung A^* setzt sich aus einer Menge disjunktiv verknüpfter elementarer Abbruchbedingungen a zusammen und lässt sich formal beschreiben als:

$$\begin{aligned}
 A^* &= \{\Theta, \Sigma, \delta, S\} & (5.59) \\
 \Theta &= \{S\} \\
 \Sigma &= \{a, \vee\} \\
 \delta &= \{S \rightarrow a, S \rightarrow a \vee S\} \\
 S &= \{S\}
 \end{aligned}$$

Diese formale Grammatik ermöglicht eine beliebige Erweiterung der hier aufgeführten Abbruchbedingungen.

Da in der Theorie des Q-Learnings der Trainingsprozess auf eine unendliche Dauer ausgelegt ist, muss ein Abbruchkriterium definiert werden, bei dem der Trainingsprozess beendet wird. Folgende Abbruchkriterien für den Trainingsprozess werden vorgeschlagen:

1. *Abbruch nach Anzahl n durchgeführter Lernepisoden ($abort_{t_{nle}}$):* definiert numerisch das Ende des Lernprozesses und ermöglicht so eine gute Abschätzung der Dauer des Trainings¹³⁴
2. *Abbruch nach einer definierten Laufzeit ($abort_{t_{nrt}}$):* ermöglicht volle Kontrolle über die Laufzeit des Lernverfahrens

5.3.3. Generierung und Verwendung von Regeln

5.3.3.1. Regelgenerierung – Von Q-Werten zum Regelsystem

Die gelernten Q-Werte werden als Tupel der Datenstruktur

$$q_i : c_m \times a_n \tag{5.60}$$

mit $a_n \in A$ als eine Aktion aus der Menge aller zugelassenen Aktionen bzw. Änderungsplanungsverfahren A und $c_m \in C$ als einen Cluster aus der Menge aller erzeugten Cluster C definiert, wobei m jedes $c \in C$ eindeutig identifiziert. Für $q_i \in Q$ gilt $\forall q \in Q \subset \mathbb{R}$.

Über reelle Zahlen kann eine Ordnung dieser Zahlen definiert werden. Die Ordnung (Q_c, \geq) sortiert die $q_i \in Q$ eines Clusters c absteigend. Daraus resultiert z. B. für alle Q-Werte des Clusters c_1 mit den zugelassenen Aktionen $A = \{a_1, a_2, a_3\}$ die Ordnung der Tupel $q_1 \geq q_2 \geq q_3 = (c_1, a_3) \geq (c_1, a_2) \geq (c_1, a_1)$. Diese Ordnung ist sinnvoll, da das Tupel q mit dem größten Q-Wert des Clusters c_m als bestmögliche Aktion am Anfang der Menge der Q-Werte steht. Die Ordnung der Q_i bestimmt die Priorität der Regeln über den Parameter ID .

Mit dieser Ordnung können die Regeln der Form

$$ID \ q_i \ WENN \ c_m \ DANN \ a_n \tag{5.61}$$

für alle Cluster in absteigender Reihenfolge sortiert generiert werden. Hierzu kann Algorithmus 5.4 verwendet werden. Dabei wird der Korpus der *WENN*-Bedingung durch die im Clustering implizit verarbeiteten Nebenbedingungen subsummiert. Da in einem Cluster mehrere Aktionen den gleichen Q-Wert besitzen können, ist die Disjunktion von Aktionen bei der Regelerzeugung zulässig.

¹³⁴In dieser Abschätzung müssen die Abbruchbedingungen der Lernepisode berücksichtigt werden.

Algorithmus 5.4 : Generierung der Steuerungsregeln je Zustandscluster c_m

Eingabe : Cluster C_i mit absteigend sortierten Q-Werten

Ausgabe : Priorisierte Regeln

```
1 WHILE  $c_m$  DO
2   WHILE  $i$  DO  $i++$ 
3     wähle( $q_i$ )
4     generiereRegel(„ID  $i$  WENN  $c_m$  DANN  $a_n(q_i)$ “)
5 END
```

5.3.3.2. Partielle Aktualisierung von Regeln

Wurde nach einem Training die Konfiguration des Produktionsnetzwerkes geändert, so wirkt sich dieses auf die Gültigkeit der Regeln im Netzwerk aus, z. B. durch die Änderung der Lagerkapazität eines Objektknotens. Da die Cluster mit der originären Konfiguration erzeugt wurden, sind diese nach Änderung des Produktionsnetzwerkes als charakteristische Zustände teilweise nicht mehr repräsentativ. Es muss eine Aktualisierung der Regeln der betroffenen Cluster erfolgen.

Da die Cluster für einen einzelnen Objektknoten erzeugt werden, sind nur die mit den vakanten Clustern verknüpften Regeln betroffen. Da es sich um eine beschränkte Anzahl an Regeln handelt, ist es sinnvoll, diese neu zu lernen. Um Regeln eines betroffenen Objektknotens partiell zu aktualisieren, muss folgendermaßen vorgegangen werden:

1. Löschen der Cluster
2. Anpassen der Szenariokonfiguration
3. Durchführen des Clusterings für die betroffenen Objektknoten
4. Durchführen des Lernprozesses für die betroffenen Objektknoten
5. Generieren des neuen Regelsystems für die betroffenen Objektknoten

Da in den hier vorgestellten Trainingsverfahren die Effizienz des Verfahrens im Vordergrund steht, ist eine lokal begrenzte Neuerstellung von Regeln durchaus praktikabel und unter Nutzung der gegebenen Konzepte gut umzusetzen. Der Nachteil ist, dass die Steuerung des Produktionsnetzwerkes für diese Zeit mit ungültigen Regeln operieren würde.

Prüfen bestehender Regeln vor der Anwendung

Trotz geänderter Regeln kann jedoch durch einen Workaround im Rahmen der Regelanwendung überprüft werden, ob die ursprünglichen Regeln noch verwendbar sind. Dieses kann durch eine zu definierende Distanztoleranzfunktion $\Delta(D)$ durchgeführt

werden, die feststellt, ob für einen Zustand ein tolerierbarer Cluster bzw. ein charakteristischer Planverlauf vorhanden ist. Diese Funktion verwendet die Distanzfunktion dieser Arbeit.¹³⁵ Dabei wird zusätzlich der relative maximale Abstand zwischen Clustern und Plänen über einen prozentualen Wert bestimmt. Bei positiver Evaluation kann die Regel verwendet werden.

5.3.3.3. Steuerung – Vom Zustand zur Regelanwendung

Die gelernten Regeln sollen im Betrieb eines Produktionsnetzwerkes zur Steuerung der Änderungsplanung eingesetzt werden. Die schnelle Verfügbarkeit ist zur effizienten Anwendung der Regeln zur Steuerung der Änderungsplanung erforderlich. Ad-hoc auf ungültige Zustände anwendbare Regeln können den Änderungsplanungsprozess des Planers beschleunigen und rechtfertigen den Aufwand des Trainingsprozesses des maschinellen Lernsystems.

Die Rückführung von Zuständen auf Regeln funktioniert in Analogie zum Clusteringprozess aus Kapitel 5.1.2. Ein Zustand kann über die Clusterfunktion eindeutig auf einen zugehörigen Cluster abgebildet werden. Wurde der passende Cluster identifiziert, kann über dessen Q-Werte die effektivste Regel ausgewählt werden, die in diesem Fall das Tupel (c_m, a_n) ist, welches dem höchsten Q-Wert zugeordnet wurde.

$$\max(q_i) \rightarrow \text{best}(c_m, a_n) \quad (5.62)$$

Abbildung 5.26 fasst die erforderlichen Schritte zur Regelanwendung zusammen.

5.3.4. Konvergenzbetrachtung des Lernverfahrens

Wesentliche Kriterien zur bewiesenen¹³⁶ Sicherstellung der Konvergenz des Q-Learning-Ansatzes sind:

1. die fortlaufende Iteration und Bewertung von Zustands-/Aktionspaaren,
2. dass erlernte Q-Werte $Q(s, a)$ in einer endlichen Tabelle speicherbar sind und dort für jedes Tupel (s, a) ein Wert vorhanden ist und
3. dass das Lernproblem durch einen Markov-Decision-Prozess abbildbar ist.

Fall (1) ist dadurch erfüllt, dass das Lernverfahren während des Trainings für einen beliebigen Zeitraum über den Explorationsraum iterieren kann. Ein Abbruch des Trainings kann dadurch bestimmt werden, dass keine signifikante Verbesserung der Q-Werte stattfinden und eine Konvergenz des aktuellen Lernproblems angenommen werden kann.

¹³⁵Siehe Kap. 5.1.3, S. 84 ff.

¹³⁶Siehe z. B. zusammengefasste Referenzen in [Mit97], S. 386 ff.

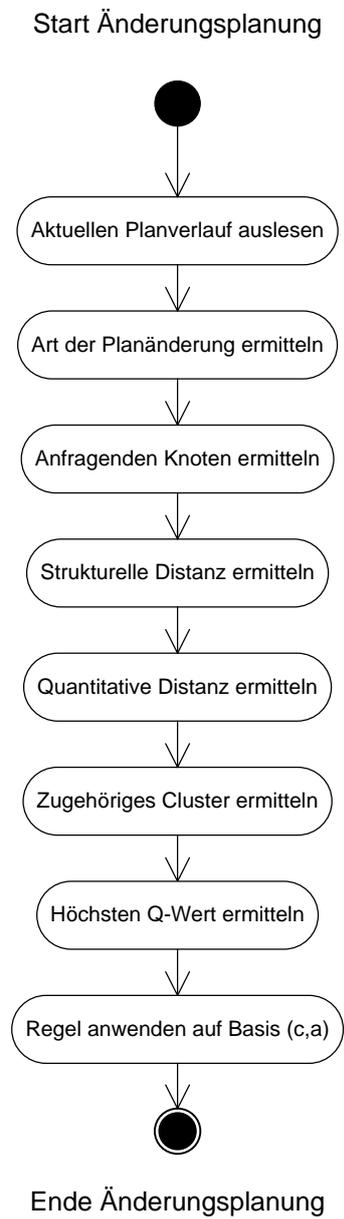


Abbildung 5.26.: Aktivität: Schrittweiser Ablauf einer Regelanwendung

Fall (2) ist durch das Clustering sichergestellt. Die Anzahl der Cluster kann vorgegeben werden und der quasi unendliche Zustandsraum eines Produktionsnetzwerkes wird auf die Anzahl vorgegebener Cluster verdichtet und kann in einer endlichen Tabelle gespeichert werden. Die (c, a) sind repräsentativ für die oben aufgeführten (s, a) verwendbar, da ausschließlich auf Clusterbasis gelernt wird.

Für den Fall (3) wurde in Kapitel 2.2.4.3 hergeleitet, dass sich das Problem dieser Arbeit als MDP formulieren lässt. Es wurde ausgeführt: „Das Lernproblem dieser Arbeit ist in einem MDP darstellbar. Jede Auswahl eines lokalen oder globalen Änderungsplanungsverfahrens verwendet den aktuellen Produktionsplan eines Objektknotens. Vergangene Pläne müssen bei dieser Entscheidung nicht explizit berücksichtigt werden. Der aktuelle Produktionsplan beinhaltet das Ergebnis aller vorher durchgeführten Planungsaufgaben.“¹³⁷

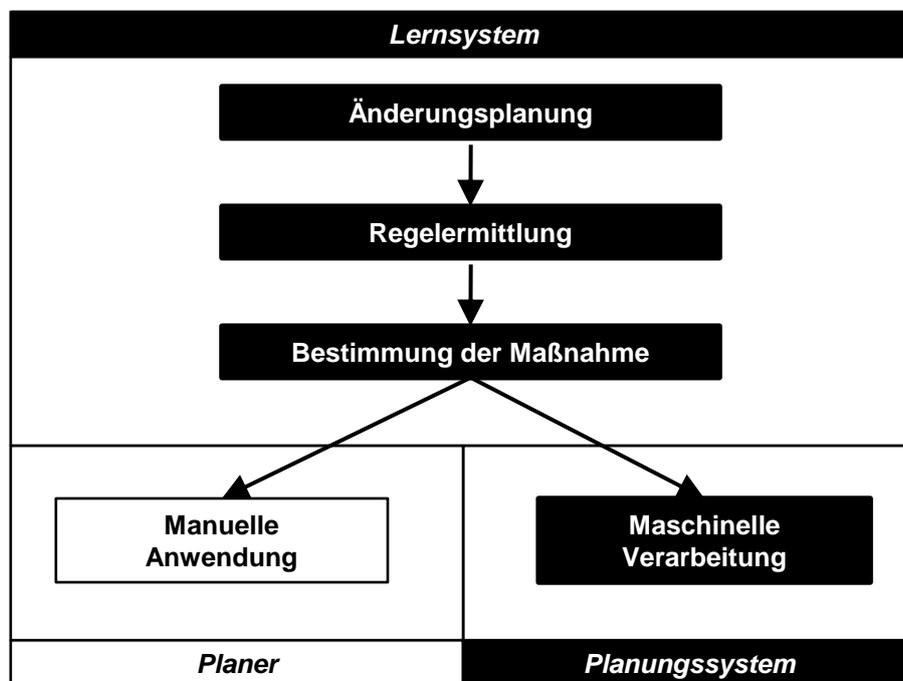


Abbildung 5.27.: Einbindung des Regelsystems in den Anwendungskontext

5.3.5. Zusammenfassung Training und Regelanwendung

Es wurde der Ablauf von Lernepisoden und Lernschritten und deren Integration in den Trainingsprozess vorgestellt. In Bezug auf das Clustering wurden Rahmenbedingungen für benötigte Ausgangsdaten dargestellt, die im Training zur Verfügung ste-

¹³⁷Siehe Kap. 2.2.4.3, S. 36

hen müssen. Um das Training kontrollierbar zu gestalten, wurden konkatinierbare Abbruchbedingungen für Lernepisoden und das Training erarbeitet.

Zur Erzeugung des Regelsystems wurde ein Algorithmus konzipiert. Es wurde diskutiert, was im Falle einer Änderung der Netzwerkkonfiguration mit den gelernten Regeln geschieht und wie diese weiter verwendet und aktualisiert werden können. Abschließend wurde erläutert, wie die Regeln zur Steuerung der Änderungsplanung angewendet werden können.

Durch das Lernen von clusterspezifischen Q-Werten zu Cluster-/Aktionspaaren können Regeln über Q-Werte anwendungsbezogen bereitgestellt werden, ohne dass sie explizit erzeugt werden müssen. Sie können in der Änderungsplanung verwendet werden, um effektiv Maßnahmen zur Auflösung ungültiger Zustände im Produktionsnetzwerk festzulegen. Beide Anforderungen an das Lernsystem

- von Q-Werten zum Regelsystem und
- vom Zustand zur Regelanwendung

sind abgedeckt.

Die Einbindung des Lernsystems in ein Planungssystem, z. B. ein ERP-System, kann dadurch erfolgen, dass im Falle der Auslösung eines Änderungsplanungsprozesses die erforderliche Maßnahme durch eine gelernte Regel vorgegeben wird. Der Vorteil des Lernsystems ist, dass die Steuerung der Änderungsplanung mit den gelernten Regeln nach wie vor manuell durchgeführt werden kann, indem der Planer einen Vorschlag für ein anzuwendendes Änderungsplanungsverfahren umsetzt. Da durch den Regelgenerator ausgeleitete Regeln maschinell lesbar sind, können sie unabhängig vom Lernsystem in ein Decision-Support-System über eine Regelschnittstelle integriert werden. Abbildung 5.27 stellt dar, wie eine Regel angewendet werden kann. Das vorgestellte Konzept bietet zwei Varianten. Die Regeln können erzeugt oder zur Laufzeit über die geordneten Q-Werte je Cluster direkt durch eine Cluster-/Planzuordnung angewendet werden.

5.4. Zusammenfassung und Bewertung der Konzeption

Es wurde ein neuer Ansatz zum Entwurf eines maschinellen Lernverfahrens zum Lernen von Regel zur Steuerung der Änderungsplanung in Produktionsnetzwerken dargestellt. Durch die offene Ausgestaltung der Konzepte können die bestehenden Funktionen durch ergänzende Funktionen, z. B. alternative Kostenfunktionen zur Strafkostenberechnung, erweitert werden. Die Parametrisierbarkeit der Lernfunktionen und des Trainingsprozesses hinsichtlich des Anwendungsfalles ermöglicht die Übertragbarkeit

der Ergebnisse auf verschiedene Produktionsnetzwerke der Serienfertigung, wie sie in Kapitel 2.1 klassifiziert wurden.

Um automatisiert ein Regelsystem zur Steuerung der Änderungsplanung zu erzeugen, muss der in Abbildung 5.28 zusammenfassend dargestellte Prozess durchgeführt werden. Zunächst muss das Produktionsnetzwerk in die MFERT-Notation überführt werden, um die Fertigungsstufen des Produktionsnetzwerkes als Objektknoten zu modellieren. Es werden sowohl Restriktionsgrenzen jedes einzelnen Objektknotens als auch die Leistungsvereinbarungen zwischen den Objektknoten definiert.

Zur Durchführung des Trainings des Lernsystems, aber auch des Clusterings, sind Ausgangsdaten notwendig. Es wurde erläutert, welche Arten von Ausgangsdaten für das Lernsystem benötigt werden und wie diese beschafft werden können. Als nächstes werden die Parameter der Lernfunktionen und die Trainingsparameter, sowohl des Clusterings als auch des Lernverfahrens, definiert. Beim Clustering kann festgelegt werden, wie viele Cluster erzeugt werden sollen, welche Ausgangsdaten verwendet werden sollen und wie oft beim Clustervorgang iteriert werden soll. Beim Lernverfahren müssen die Gewichte der einzelnen Quotienten der Rewardfunktion bestimmt und Abbruchbedingungen für die Lernepisoden und den Trainingsprozess festgelegt werden. Hiernach wird der Lernprozess angestoßen.

Das Clustering des Zustandsraumes wird durchgeführt. Nach Abschluss des Clusterings wird mit dem Lernverfahren je zugelassener Aktion für die Änderungsplanung eines Objektknoten ein Q-Wert gelernt und auf Clusterebene gespeichert. Durch die Sortierung der Q-Werte können die Regeln abgeleitet werden. Je höher der Q-Wert einer Aktion, desto besser wird das wahrscheinliche Ergebnis der mit den gelernten Regeln gesteuerten Änderungsplanung. Der Algorithmus zur Erzeugung des Regelsystems kann dann die Regeln aus den Q-Werten ableiten.

In den folgenden Tabellen 5.3 – 5.5 werden die erstellten Konzepte den Forschungsfragen A–C der Arbeit zugeordnet und mit den jeweiligen Anforderungen verglichen.

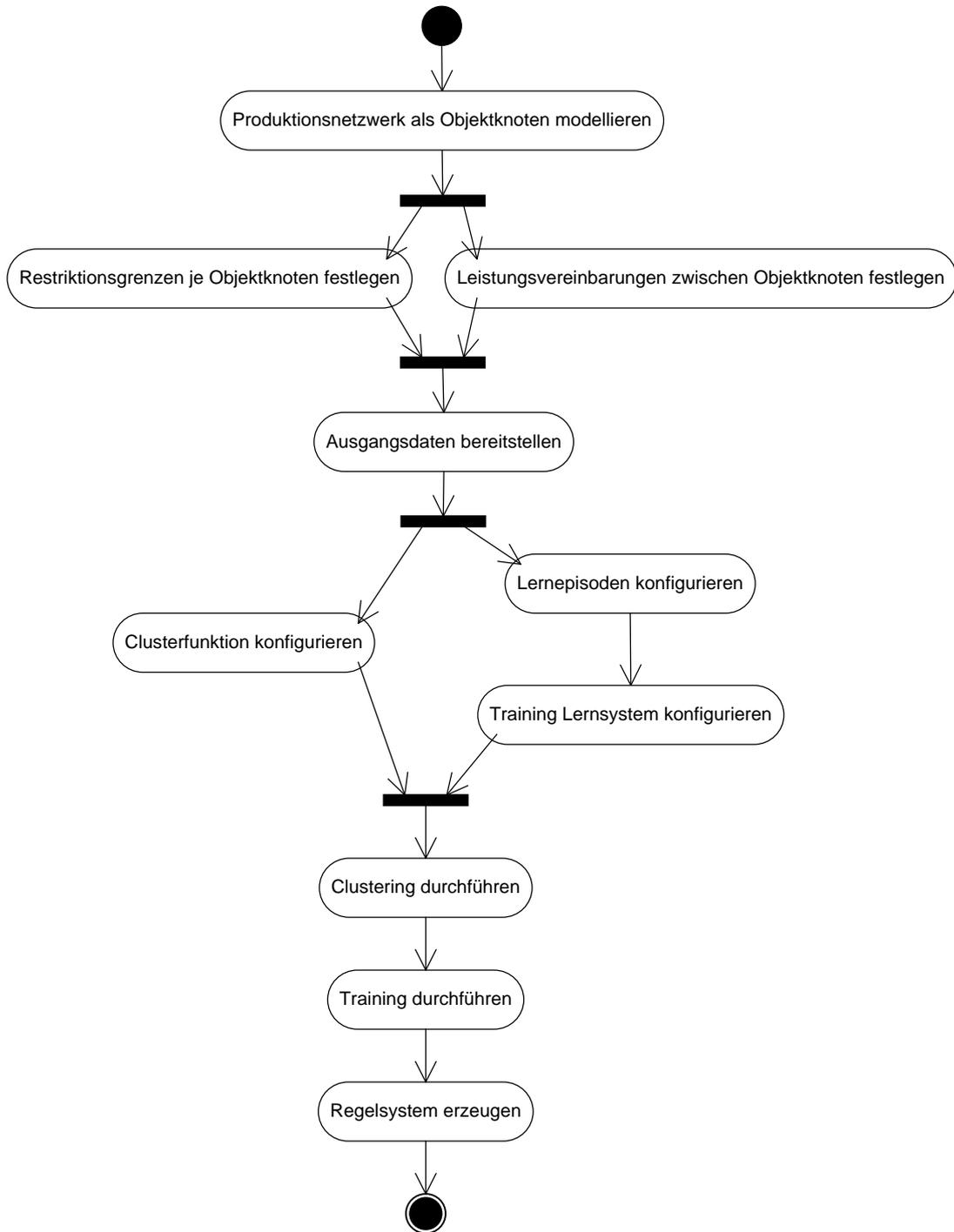


Abbildung 5.28.: Aktivität: Erstellung des Regelsystems

5. Konzeption

Tabelle 5.3.: Zusammenfassung Anforderungen und Lösungskonzepte zur Forschungsfrage A

Beschreibung	Anforderung	Lösungskonzept
Abstraktion des Zustandsraumes Reduktion der Problemkomplexität	Zu- zur Pro- Eindeutigkeit und Vollständigkeit des Abstraktionsverfahrens	Sicherstellung der eindeutigen Abbildung im Clusterverfahren aller Zustände des Produktionsnetzwerkes auf die Cluster
	Betriebswirtschaftlich interpretierbar	Konzeption der domänenspezifischen Distanzfunktion unter der Berücksichtigung menschlich analytischer Vorgehensweisen
	Skalierbarkeit des Explorationsraumes	Sicherstellung der Skalierbarkeit des Explorationsraumes durch Parametrisierbarkeit der Clusterfunktion

Tabelle 5.4.: Zusammenfassung Anforderungen und Lösungskonzepte zur Forschungsfrage B

Beschreibung	Anforderung	Lösungskonzept
Realitätsnahe Funktion zum Lernen anwendungsbezogener Regeln	Quantitative Bewertung von Zuständen	Aufstellung der zustandsbezogenen Rewardfunktion auf Kostenbasis, die über eine Abbildung an clusterbezogene Q-Werte gekoppelt wird
	Berücksichtigung systeminhärenter lokaler Kosten	Parameter der Rewardfunktion, die lokale Charakteristika der Planverläufe der Objektknoten (FOK/KOK) bewertet und im Zuge der Rewardberechnung einer Aktion durch Normierung von Plänen deren Vergleichbarkeit sichert
	Berücksichtigung globaler Kosten zur Bewertung von Lieferantenbeziehungen und globaler Zusammenhänge	Benennung von Parametern, die unter Verwendung einer spezifischen Beschaffungssteuerung globale Zusammenhänge kostentechnisch mit der Rewardkalkulation bewerten

Tabelle 5.5.: Zusammenfassung Anforderungen und Lösungskonzepte zur Forschungsfrage C

Beschreibung	Anforderung	Lösungskonzept
Realitätsbezogenes Trainingskonzept	Effiziente Lernepisoden für leistungsfähiges Training	Konzeption der Lernepisoden
	Gesamttrainingskonzept mit kontrollierbarer Laufzeit	Konzeption eines Trainingskonzeptes über definierte Lernepisoden mit parametrisierbaren Abbruchbedingungen
	Ausgangsdaten	Ausarbeitung eines Konzeptes zur Bereitstellung von realen Ausgangsdaten und zur Einbindung dieser Ausgangsdaten in das Training
	Konvergenzeigenschaften	Hergeleitet durch die Einhaltung der Vorbedingungen des Q-Learnings und die Modellierung des Problems als MDP
Generierung der Regeln aus den Lernergebnissen	Generieren von Regeln	Konzeption eines Algorithmus der Regeln zur Steuerung der Änderungsplanung des Produktionsnetzwerkes aus den gelernten Q-Werten generiert
	Verwendung der Regeln im Anwendungskontext	Konzeption eines Algorithmus zur Extraktion der Regeln zur Laufzeit einer Änderungsplanung und Beschreibung möglicher Einbindung des Verfahrens in ein reales ERP-System

6. Validierung

Die Erfahrung lässt sich ein
furchtbar hohes Schulgeld
bezahlen, doch sie lehrt wie
niemand sonst!

(Thomas Carlyle)

In diesem Kapitel wird die Validierung der Konzepte aus Kapitel 5 beschrieben. Die Validierung erfolgt in Kapitel 6.1 für das Clustering und in Kapitel 6.2 für das Lernverfahren mithilfe überschaubarer Szenarien. Durch die Validierung soll gezeigt werden, ob eine effektive Lernfunktion konzipiert wurde, die im Sinne der Problemstellung sinnvolle Regeln zur Steuerung der Änderungsplanung in Produktionsnetzwerken lernt, und ob durch die Nutzung des Abstraktionsverfahrens ein effizientes Training des Lernverfahrens möglich wird. Da die Ausgestaltung der Abstraktionsfunktion und der Lernfunktion für FOK und KOK konzeptionell ähnlich ist, wird der Fokus bei der Validierung auf das Zusammenspiel der FOK gelegt. Diese sind die Schnittstelle für das Lernen globaler Regeln zur Steuerung der koordinationsbasierten Änderungsplanung zwischen Kunde und Lieferant.

6.1. Validierung des Abstraktionsverfahrens

Die Validierung für das Clustering erfolgt mit einem Szenario bestehend aus jeweils drei FOK, KOK und PK. Dieses ist in Abbildung 6.1 dargestellt. Bei der Validierung des Clusterings in diesem Kapitel wird untersucht, wie viele Cluster erzeugt werden müssen, um eine gute Streuung der Zustände zu erhalten. Es wird betrachtet, ob die Rewards der einzelnen Planverläufe eines Clusters ähnliche Werte annehmen. Dieses ist wichtig, da die Q-Werte des Lernverfahrens nur dann sinnvoll auf Clusterebene gelernt werden können, wenn einem Cluster durch die Abstraktionsfunktion Zustände mit ähnlichen Strafkosten zugeordnet worden sind.

In Kapitel 6.1.1 wird das Evaluationsszenario beschrieben. In Kapitel 6.1.2 wird detailliert betrachtet, wie sich verschiedene Parametrisierungen auf das Clustering auswirken. Im Kapitel 6.2.3 wird die Kompatibilität zwischen Clustering und Lernfunktion

betrachtet, indem die Ähnlichkeiten der einzelnen Planbewertungen der Cluster untersucht werden. Es wird abschließend gezeigt, dass die Planverläufe der Cluster ähnlichen Rewardbewertungen unterliegen und die Verwendung der Cluster beim Lernen der Q-Werte möglich und sinnvoll ist.

6.1.1. Szenario zur Validierung der problemspezifischen Abstraktion

Um die Funktionsweise des Clusterings für Trainingsdaten zu zeigen, werden Ergebnisse eines Clusteringprozesses für ein unterschiedlich konfiguriertes Produktionsnetzwerk durchgeführt. Die charakteristischen Planverläufe werden für jeden Objektknoten einzeln gelernt, sodass knotenspezifische Eigenschaften beim Generieren der nötigen Trainingsdaten berücksichtigt werden müssen.¹ In dieser Validierung wird ein Fertigungsobjektknoten FOK betrachtet, der Zugänge an Material von einem vorgelagerten Fertigungsprozess P_{in} erhält und diese über zwei Kanten alternativ an die nachgelagerten Prozesse P_{out}^1 und P_{out}^2 abgibt. Diese Struktur ist in Abbildung 6.1 grafisch dargestellt.

Zwischen den Objektknoten im Produktionsnetzwerk können Leistungsvereinbarungen getroffen werden, die eine unterschiedliche Beschaffungssteuerung festlegen. Pläne mit mehreren Beschaffungsarten weisen unterschiedliche Muster auf. Tabelle 6.1 zeigt die insgesamt vier Szenarien, für die beispielhaft aus einer Menge von jeweils 2000 Planverläufen Cluster erzeugt wurden.²

Die Trainingsdaten wurden in 200 Cluster abstrahiert. Abbildung 6.2 zeigt für jedes dieser Szenarien ein Planverlaufs-Cluster, an denen sich die Unterschiede zwischen den Planverläufen je nach den konfigurierten Zu- und Abgangsmustern erkennen lassen.³

In Szenario $S_{Z,ZP}$ finden die Zugänge in festen Zyklen mit einem Intervall von 3 Planungsperioden statt. Die Abgänge erfolgen kontinuierlich in jeder Periode sowie punktuell mit einer Wahrscheinlichkeit von 0,2. Die Dominanz von Bewegungen in festen Zyklen führt zu gleichmäßigen und periodischen Planverläufen. Abbildung 6.2(a) zeigt beispielhaft eine Menge von ähnlichen Planverläufen, in denen die zyklischen Zugänge

¹Mangels Zugriff auf Realdaten mussten hier realitätsbezogene Ausgangsdaten erzeugt werden. Details dazu sind in Anhang B dargestellt. Die Validierung auf erzeugten Ausgangsdaten mindert aber nicht die Aussagekraft der Validierung. Die verwendeten Daten können erstens so oder so ähnlich in der Realität vorkommen und zweitens erfüllen die erzeugten Ausgangsdaten alle definierten Eigenschaften für Ausgangszustände dieser Arbeit. Die Funktionsweise des Abstraktionsverfahrens und der Lernfunktion kann dementsprechend genauso gezeigt werden.

²Details zu Trainingsdaten siehe im Anhang B

³Es handelt sich um jeweils eines von 200 generierten Planverlaufsclustern.

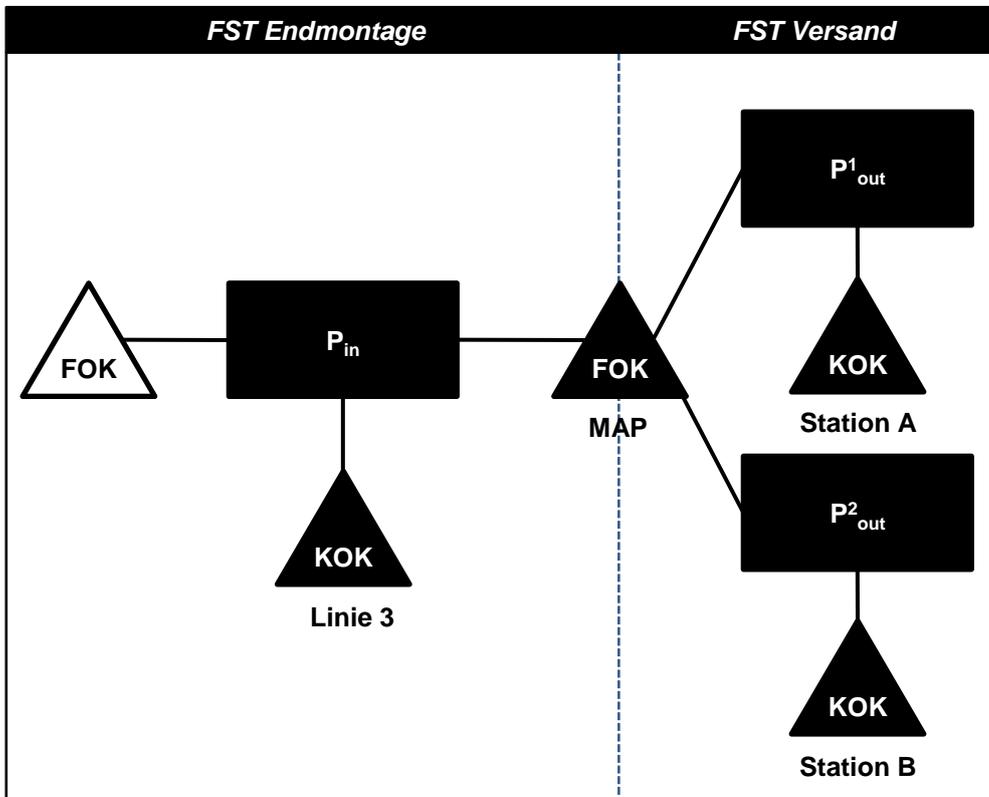


Abbildung 6.1.: Beispielproduktionsnetzwerk für die Validierung des Clustering

Tabelle 6.1.: Definition der Testszenarien

Szenario	Zugang	Abgang
$S_{Z,ZP}$	P_{in} : Zyklus ($\Delta = 3; I = [40, 50]$)	P_{out}^1 : Zyklus ($\Delta = 1; I = [10, 15]$) P_{out}^2 : Punkt ($w = 0, 2; I = [10, 15]$)
$S_{Z,PP}$	P_{in} : Zyklus ($\Delta = 3; I = [40, 50]$)	P_{out}^1 : Punkt ($w = 0, 6; I = [10, 15]$) P_{out}^2 : Punkt ($w = 0, 3; I = [5, 10]$)
$S_{P,ZP}$	P_{in} : Punkt ($w = 0, 7; I = [10, 20]$)	P_{out}^1 : Zyklus ($\Delta = 4; I = [40, 50]$) P_{out}^2 : Punkt ($w = 0, 3; I = [10, 15]$)
$S_{P,PP}$	P_{in} : Punkt ($w = 0, 6; I = [15, 30]$)	P_{out}^1 : Punkt ($w = 0, 5; I = [10, 15]$) P_{out}^2 : Punkt ($w = 0, 2; I = [5, 10]$)

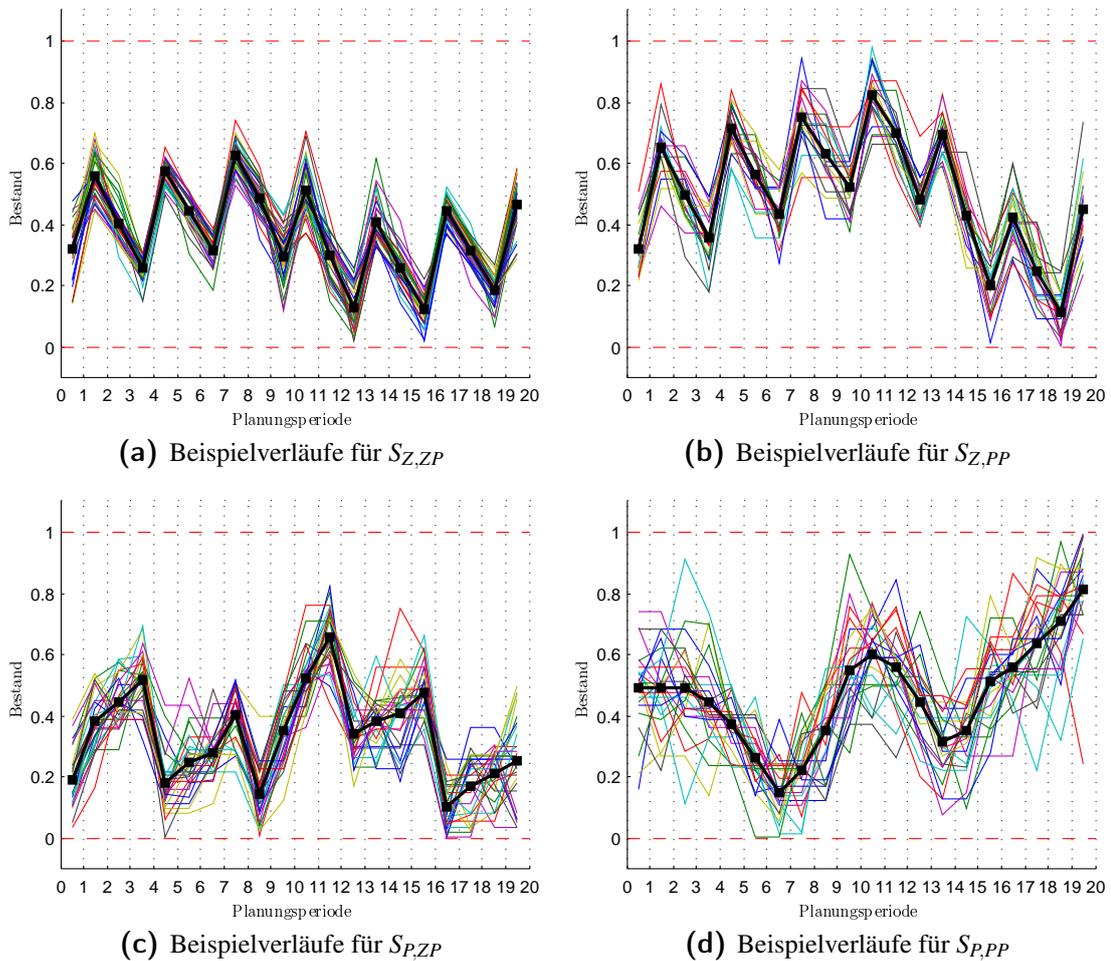


Abbildung 6.2.: Planverläufe für unterschiedliche Zu- und Abgangsmuster

sowie die größtenteils kontinuierlichen Abgänge deutlich zu erkennen sind. In Szenario $S_{Z,PP}$ bleiben die Zugänge unverändert, während die Abgänge ohne feste Zyklen stattfinden. Die resultierenden Pläne behalten einen klar erkennbaren zyklischen Verlauf, der deutlich stärkere Schwankungen aufweist. Beispiele für derartige Planverläufe sind in Abbildung 6.2(b) zu sehen.

In Szenario $S_{P,ZP}$ erfolgen die Zugänge nicht mehr zyklisch, sondern punktuell, während die Abgänge sowohl zyklisch als auch zusätzlich punktuell erfolgen. Durch die unstetigen Zugänge weisen die Planverläufe größere Schwankungen auf. Wie sich aus den Planverläufen in Abbildung 6.2(c) erkennen lässt, bilden die festen Zyklen für die Abgänge die einzige Regelmäßigkeit in den Verläufen. Unterliegen wie in Szenario $S_{P,PP}$ weder die Zu- noch die Abgänge festen Zyklen, werden die Schwankungen der Planverläufe noch größer. Dies zeigt insbesondere ein Vergleich des beispielhaft ausgewählten Clusters in Abbildung 6.2(d) mit den weiteren Trainingsdaten, die für dieses Szenario generiert wurden.

Die beispielhaft ausgewählten Cluster zeigen bereits durch die Muster auf visueller Ebene, dass durch das Abstraktionsverfahren für verschiedene Szenarien zielführende Pläne zu charakteristischen Plänen abstrahiert werden können. Die Centroiden repräsentieren in ihrem Verlauf typische Zustände des Produktionsnetzwerkes unter gegebenen Leistungsvereinbarungen. Trotz Abstraktion kann eine direkte Verbindung zwischen den ursprünglichen diskreten Zuständen des Produktionsnetzwerkes und den abstrahierten Zuständen hergestellt werden.

Die Auswertungen in den folgenden Kapiteln wurden mit denselben Trainingsdaten durchgeführt. Zu diesem Zweck wurden 10.000 Beispiel-Planverläufe mit der Trainingsdatenkonfiguration des Szenarios $S_{P,ZP}$ generiert und anschließend Bruttobedarfs-erhöhungen mit einer zufällig gezogenen Höhe aus dem Intervall $[0,3;0,6]$ und zufällig gewählten Zeitpunkten modifiziert. Auf jeden Plan wurden so lange Planänderungen angewendet, bis er mindestens eine Restriktionsverletzung aufweist und dann durch das Clusteringverfahren abstrahiert.

6.1.2. Validierung der Parametereinstellungen

In diesem Kapitel wird diskutiert, wie sich Änderungen verschiedener Parameter der Abstraktionsfunktion auf die Ergebnisse des Clusterings auswirken. Im Fokus dieser Betrachtungen steht die Güte der resultierenden Cluster, gemessen an der Varianz innerhalb der Cluster. Diese Varianz ist als Summe der quadrierten Abweichungen aller Cluster-Elemente vom Clustermittelpunkt definiert⁴. Die Güte einer Partitionierung der Trainingsdatenmenge kann also über die Summe dieser Varianzen

$$SSE = \sum_{C_i} \sum_{x_j \in C_i} d(x_j, c_i)^2 \quad (6.1)$$

bestimmt werden.⁵

6.1.2.1. Anzahl der Cluster

Einer der zentralen Parameter der Zustandsabstraktionsfunktion ist die Festlegung der Anzahl zu erzeugender Cluster k . Diesem Parameter kommt eine wichtige Bedeutung zu, da er implizit die Anzahl der resultierenden abstrahierten Zustände festlegt, die später im Rahmen des Q-Learnings verwendet werden. Der Grad der Zustandsabstraktion wird so durch die Wahl von k bestimmt. Problematisch ist der offensichtliche Zielkonflikt zwischen einem kleinen Zustandsraum für das Lernverfahren und einer guten Abbildung von einzelnen Zuständen auf die abstrahierten Zustände. Da sich dieses

⁴Engl. *Sum of squared errors*, SSE

⁵Die Minimierung dieser Varianz ist gerade das Ziel des k -means-Clustering, vgl. Kap. 3.1.3, S. 55.

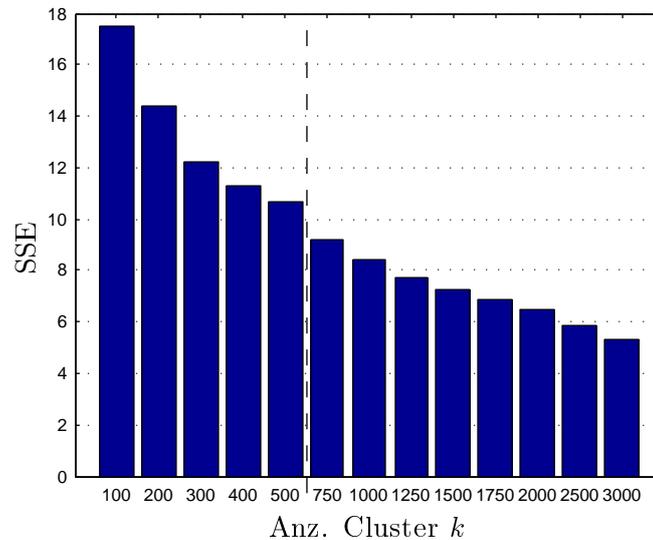


Abbildung 6.3.: Summe quadrierter Distanzen für verschiedene k

Entscheidungsproblem nicht lösen lässt, existiert keine exakte Methode, eine optimale Anzahl an k -Clustern zu bestimmen. Stattdessen ist dieser Parameter so zu wählen, dass sowohl die Varianz innerhalb der Cluster als auch die Anzahl der generierten charakteristischen Planverläufe für die abstrahierten Zustände akzeptabel bleibt. Diese Auswahl kann durch eine empirische Näherung erfolgen, indem verschiedene Werte für k getestet werden.

Wenn die Laufzeit des Clusterverfahrens kein kritischer Faktor ist, sollten verschiedene Konfigurationen durchgeführt werden, um anschließend einen Wert für k zu wählen, der das beste Verhältnis von gemessener Varianz in den Clustern zur Anzahl generierter Cluster bietet. Abbildung 6.3 zeigt die Summe der quadrierten Distanzen (SSE) für verschiedene k . Dabei wurde das Clusterverfahren jeweils auf die oben beschriebene Trainingsdatenmenge mit 10.000 Planverläufen angewendet, bis in einer Iteration weniger als 1% der Elemente einem anderen Cluster zugewiesen wurden. Die maximale Anzahl von Iterationen war auf 50 begrenzt. Dieses Abbruchkriterium trat hier nicht ein, da das Clusterverfahren in allen Fällen zuvor konvergierte. Die Gewichte der Distanzfunktion wurden auf $D_S = 0,7$ und $D_Q = 0,3$ gesetzt, wobei Restriktionsverletzungen sowohl beim Berechnen der Distanz als auch bei der Aktualisierung der Clustermittelpunkte doppelt gewichtet wurden ($w_{RV}^d = w_{RV}^c = 2$).

Wie erwartet sinkt der Fehler mit steigender Anzahl der Cluster, da die Anzahl von Planverläufen pro Cluster sinkt und die Mittelpunkte besser an die Cluster-Elemente angepasst werden können. Dieser Zusammenhang ist nicht linear. Ab einer bestimmten Anzahl von Clustern sinkt die Summe der quadrierten Distanzen langsamer. Man beachte insbesondere, dass die Abstände zwischen den verschiedenen Werten für k auf der x-Achse nicht konstant sind. Der starke Abfall des Fehlers auf den ersten Balken

ist bei Erhöhung von k in 100er Schritten zu beobachten, während ab $k = 500$ die Erhöhung in 250er und ab 2000 sogar in 500er Schritten erfolgt. Die Erhöhung von 100 auf 500 Cluster senkt den Fehler um 64,8%, während die Erhöhung von 500 auf 1000 Cluster nur noch eine Verbesserung von 26,6% bewirkt.

Da eine höhere Anzahl von Clustern einen höheren Lernaufwand für das Lernverfahren impliziert, muss ein geeigneter Kompromiss zwischen dem akzeptierten Fehler und der Anzahl von Clustern gesucht werden. In diesem Beispiel sind für k Werte zwischen 500 und 1000 angebracht. Werte von mehr als 1000 senken den erwarteten Fehler derart langsam, dass hier der erhöhte Aufwand für die größere Anzahl von abstrahierten Zuständen überwiegen dürfte. Es ist es ratsam, initiale Werte für k im Bereich von 5 %-10 % der Anzahl von Trainingsdaten zu wählen und sich einem zielführenden Wert wie in diesem Beispiel empirisch zu nähern. Die Tatsache, dass die Summe der quadrierten Distanzen ab einem bestimmten Wert für k langsamer sinkt, kann als Indiz dafür angesehen werden, dass die Anzahl der charakteristischen Planverläufe, die sich aus den Trainingsdaten sinnvoll ableiten lassen, nicht beliebig groß ist. Wenn eine Erhöhung von k keine signifikante Verbesserung der Qualität der Cluster bewirkt, so sind die charakteristischen Merkmale der Planverläufe offensichtlich durch die existierenden Cluster bzw. deren Mittelpunkte abgedeckt.

6.1.2.2. Gewichte der Distanzfunktion

Die Implementierung der Distanzfunktion erlaubt eine problemspezifische Definition der Gewichtung von struktureller und quantitativer Distanz. Die Wahl dieser Parameter legt fest, wie stark Abweichungen in der Höhe der Planverläufe gegenüber Abweichungen in der Verteilung der Planungsperioden mit Restriktionsverletzungen gewichtet werden. Dieses beeinflusst schließlich die Auswahl der Planverläufe, die über die gemessene Distanz zu abstrahierten Zuständen zusammengefasst werden, und hat Einfluss darauf, wie während der Lernphase über die einzelnen Zustände abstrahiert wird. Die Wahl hängt stark von den realen Voraussetzungen am betrachteten Objektknoten ab.

Im Falle von Fertigungsobjektknoten ist zu entscheiden, wie kritisch Restriktionsverletzungen anzusehen sind und ob Unterschiede in den Planverläufen innerhalb der Restriktionsgrenzen einen starken Einfluss auf die Bewertungen dieser Verläufe haben. Fallen für Bestände z. B. hohe Lagerkosten an, so kommt der quantitativen Distanz eine größere Bedeutung zu, als wenn diese Größe vernachlässigbar ist. Ähnlich ist bei Kapazitätsobjektknoten zu beurteilen, wie stark verschiedene Höhen der Auslastungen innerhalb der Restriktionsgrenzen die Bewertung eines Planverlaufes beeinflussen.

In der Mehrzahl der Anwendungsfälle ist zu erwarten, dass die zeitliche Verteilung der Restriktionsverletzungen einen größeren Einfluss auf die Wahl eines geeigneten Planungsverfahrens hat, da Restriktionsverletzungen zu inakzeptablen Zuständen führen.

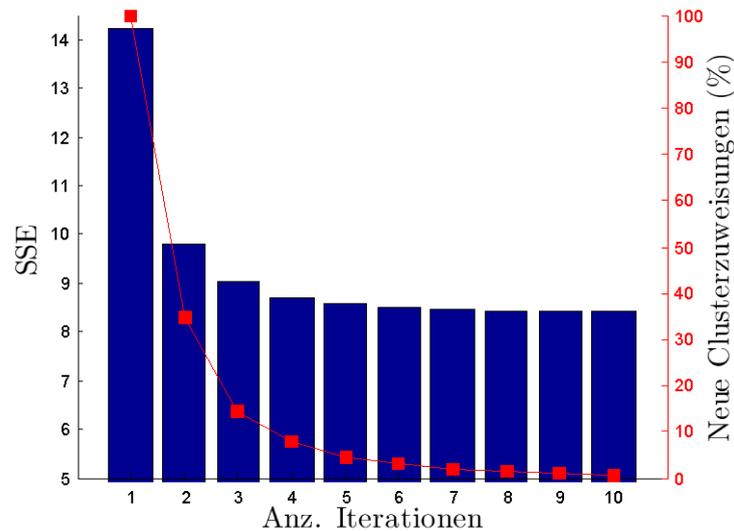


Abbildung 6.4.: Entwicklung des quadrierten Fehlers und Anteil neuer Clusterzuweisungen über die Iterationen des Clusterings

Deshalb sollte die strukturelle Distanz D_S in der Regel höher gewählt werden als die quantitative D_Q . Um eine annähernd gleiche Verteilung der Restriktionsverletzungen der Planverläufe eines Clusters zu erreichen, haben sich Werte für D_S aus dem Intervall $[0,6;0,8]$ in mehreren Durchläufen als vorteilhaft erwiesen. Für D_Q ergeben sich die Werte aus dem Intervall $[0,2;0,4]$.

6.1.2.3. Maximale Anzahl Iterationen und Konvergenztoleranz

Das Festlegen der Parameter für die Abbruchkriterien ist ähnlich wie das Festlegen eines Wertes für die Anzahl der Cluster ein empirischer Prozess. Um die Laufzeit des Clusterverfahrens nicht unnötig zu erhöhen, sollte der Algorithmus terminieren, sobald weitere Iterationen keine nennenswerten Auswirkungen mehr auf die erzeugten Cluster haben. Wann dieser Punkt eintritt, kann vom Benutzer als Parameter T festgelegt werden. Sobald in einer Iteration weniger als T Prozent der Trainingsdaten einem neuen Cluster zugewiesen werden, terminiert der Algorithmus. Dieser Parameter kann natürlich auf 0 gesetzt werden, um eine Konvergenz im Sinne des ursprünglichen k -means-Algorithmus zu erzwingen. In den verschiedenen Tests hat sich gezeigt, dass für diesen Parameter Werte im Bereich $0 < T \leq 5$ angebracht sind. Das Setzen dieses Parameters auf 0 kann zu unnötig langen Laufzeiten der Clusterings führen, bei denen insbesondere in den letzten Iterationen kaum noch eine Verbesserung des Zielfunktionswertes eintritt. Deshalb erscheint eine gewisse Toleranz bei diesem Abbruchkriterium sinnvoll. Um die Wahl dieses Parameters empirisch zu belegen, können die Entwicklungen der quadrierten Fehler sowie der Anteil der neu zugewiesenen Trainingsdaten über die Iteration eines Clusterlaufes beobachtet werden.

Abbildung 6.4 illustriert diese Beobachtungen beispielhaft am Verlauf des Clusterings auf dem oben beschriebenen Trainingsdatensatz für $k = 1000$. Es ist deutlich erkennbar, dass der Nutzen zusätzlicher Iterationen nach wenigen Wiederholungen rapide abnimmt. In der ersten Iteration werden alle Trainingsdatenpunkte einem neuen, initialen Cluster zugewiesen, was zu einem quadrierten Fehler von 14,24 führt. Die zweite Iteration senkt diesen Fehler auf 9,79 ab, wobei noch 34,91 % der Elemente ihre Clusterzugehörigkeit ändern. Dieser Abfall nimmt in den folgenden Iterationen weiter zu, sodass ab der 5. Iteration der Fehler kaum noch merklich sinkt und der Anteil der Trainingsdaten, die ihr Cluster wechseln, kontinuierlich weniger als 5 % beträgt. Nach der 10. Iteration terminiert der Algorithmus, da nur noch 0,69 % der Trainingsdaten das Cluster wechseln. Der quadrierte Fehler beträgt hier 8,40 und es ist absehbar, dass er sich durch zusätzliche Iterationen nicht mehr signifikant senken lässt.

6.1.3. Effektivität der Clusteranwendung

Bei den Auswertungen in den vorangehenden Kapiteln wurde die Funktionsweise des Clusterverfahrens vermittelt und es wurden verschiedene Parametereinstellungen des Clusterings und dessen Ergebnisse diskutiert. Dabei wurde die Qualität der resultierenden Cluster mit der eingesetzten Distanzfunktion des Clusterings quantifiziert. Es bleibt zu zeigen, dass die Reduktion der Mengen von Planverläufen auf jeweils einen charakteristischen Planverlauf für die Anwendung indem betrachteten Lernverfahren zu sinnvollen Ergebnissen führt.

Die Clustern sind für das Lernverfahren abstrahierte charakteristische Zustände, zu denen die einzelnen Aktionsbewertungen in Form von Q-Werten gespeichert werden.⁶ Die Q-Werte ergeben sich jeweils aus den Rewards, die bei Anwendung der zugehörigen Aktion auf dem Centroiden eines abstrahierten ungültigen Zustandes erzielt werden⁷. Die jeweiligen Zustandsbewertungen errechnen sich bei der Rewardberechnung aus verschiedenen Strafkostenparametern, die für bestimmte Planverläufe problemspezifisch, z. B. für die Unterschreitung eines Sicherheitsbestandes, anfallen. Es soll deshalb gezeigt werden, dass die Planverläufe, die mit dem vorgestellten Clusterverfahren abstrahiert werden, ähnliche Bewertungen nach der in Kapitel 5.2 definierten Rewardfunktion erhalten. Zu diesem Zweck sind in Abbildung 6.5 verschiedene Cluster zusammen mit der Verteilung der Strafkosten für die Elemente des jeweiligen Clusters dargestellt. Die Cluster entstammen mit einer Ausnahme⁸ den Ergebnisclustern des in Kapitel 6.1.2 beschriebenen Testlaufes für $k = 500$. Das erste Cluster ohne Restriktionsverletzungen aus Abbildung 6.5(b) ist ein Cluster des Testszenarios $S_{P,ZP}$.⁹

⁶Vgl. Kap. 5.1.2, S. 76

⁷Vgl. Kap. 5.2, S. 100

⁸Cluster aus Abb. 6.5(b)

⁹Vgl. Kap. 6.1.1

6. Validierung

Zu beachten ist, dass in den Histogrammen die *Strafkosten* dargestellt sind, sodass kleine Werte gute Bewertungen für die Planverläufe bedeuten.

Abbildung 6.5(a) zeigt einen Cluster, der Planverläufe ohne Restriktionsverletzungen beinhaltet. Da nur die anfallenden Lagerkosten bewertet werden, erhalten alle Planbewertungen relativ wenig Strafkosten. Die meisten Pläne bewegen sich in dem kleinen Intervall $[0,125;0,150]$ ¹⁰, was mit einem Unterschied von 0,025 klein bemessen ist. Die Pläne sind bzgl. der Strafkosten ähnlich.

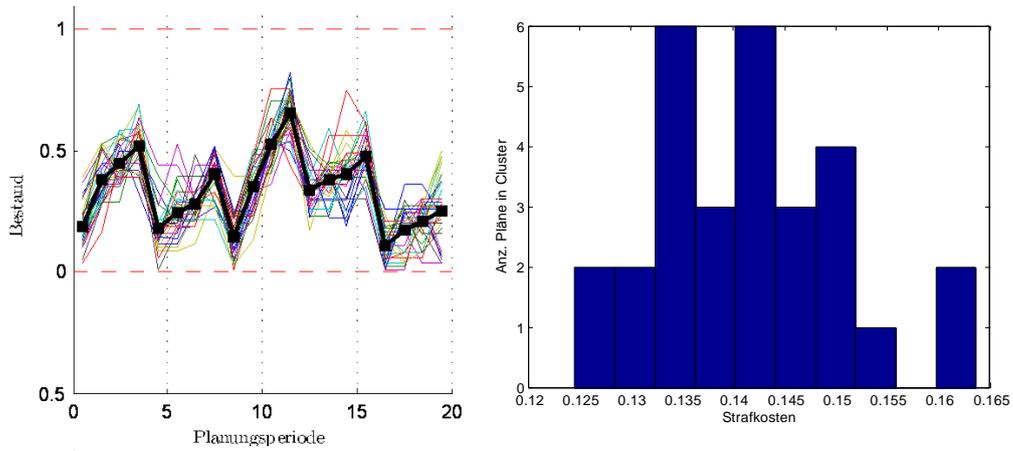
Die Planverläufe in Abbildung 6.5(b) weisen wie der zugehörige Clustermittelpunkt nur eine Restriktionsverletzung in Planungsperiode 17 auf. Deshalb können diese Verläufe noch als gut angesehen werden, zumal Restriktionsverletzungen gegen Ende des Planungshorizontes nicht so negativ zu bewerten sind wie solche, die näher an der Heutelinie liegen. Dieser Tatsache trägt die Strafkostenfunktion Rechnung, indem die Strafkosten für Planungsperioden in der Zukunft nur diskontiert in die Berechnung eingehen. Die meisten Planbewertungen bewegen sich im Intervall $[0,17;0,2]$ und fallen erneut ähnlich aus. Die Intervallgröße von 0,03 ähnelt dem Intervall aus Abbildung 6.5(a). Die Ausreißer auf 0,205 liegen mit 0,005 nah an den Strafkostenbewertungen der übrigen Clusterpläne, wobei aufgrund der Konvergenzeigenschaften der Lernfunktion und Discountfaktoren der Lernfunktion¹¹ keine Probleme im Lernverfahren zu erwarten sind.

Das Cluster in 6.5(c) fasst eine Menge kritischer Planverläufe zusammen, die in einem Großteil der Planungsperioden Restriktionsverletzungen in unterschiedlichen Höhen aufweisen. Aufgrund der Ausschläge der Planverläufe über und unter den charakteristischen Planverlauf und der hohen Strafkosten zeigen diese die aus der Abbildung zu ersiehenden Sprünge in der Bewertung. Trotzdem fällt das Spektrum der Bewertungen mit einem Bewertungsintervall von $[0,725;0,755]$, das eine Größe von 0,03 aufweist, ähnlich klein wie bei den vorhergehenden Intervallen, aus.

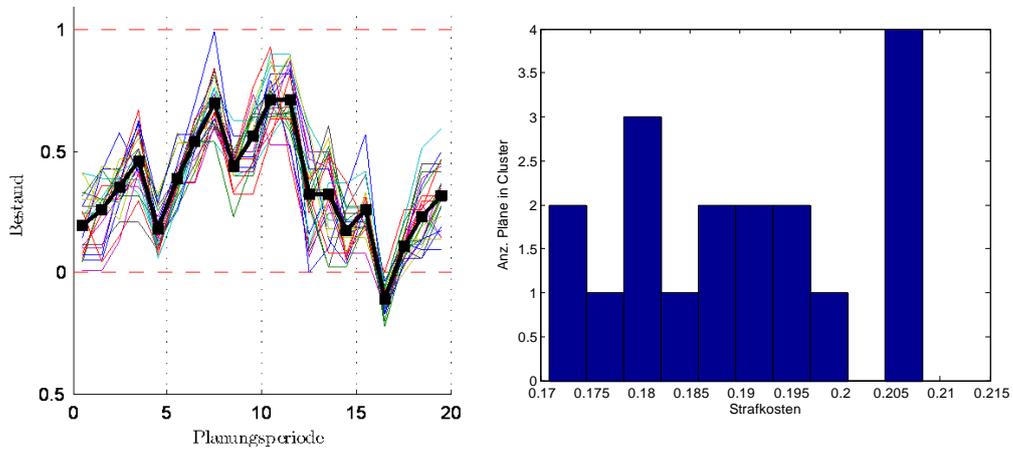
Abbildung 6.6(a) zeigt eine Unterdeckung im charakteristischen Planverlauf. Dieser weist in der zweiten Hälfte des Planungshorizontes mehrere Restriktionsverletzungen in etwa gleicher Höhe auf. Diese lassen sich mit einem erhöhten Zugang beheben. Insgesamt variieren die Bewertungen im Intervall $[0,031;0,041]$, mit einer Konzentration im Bereich $[0,34;0,38]$. Das Intervall mit den meisten Plänen ist mit einer Größe von 0,04 nah an den Bewertungen der vorhergehenden Pläne. Die vereinzelte Streuung der Strafkosten kann über die Anzahl der Cluster bzw. Anzahl der Iterationen verbessert werden. Entscheidend ist, dass die meisten Pläne bei der Bewertung in einem engen Intervall liegen und als charakteristisch im Sinne eines *abstrahierten Zustandsraumes* betrachtet werden können.

¹⁰ Alle Werte auf bis zu drei Nachkommastellen gerundet. Die Originaldaten hierzu weisen eine Genauigkeit von 15 Nachkommastellen auf.

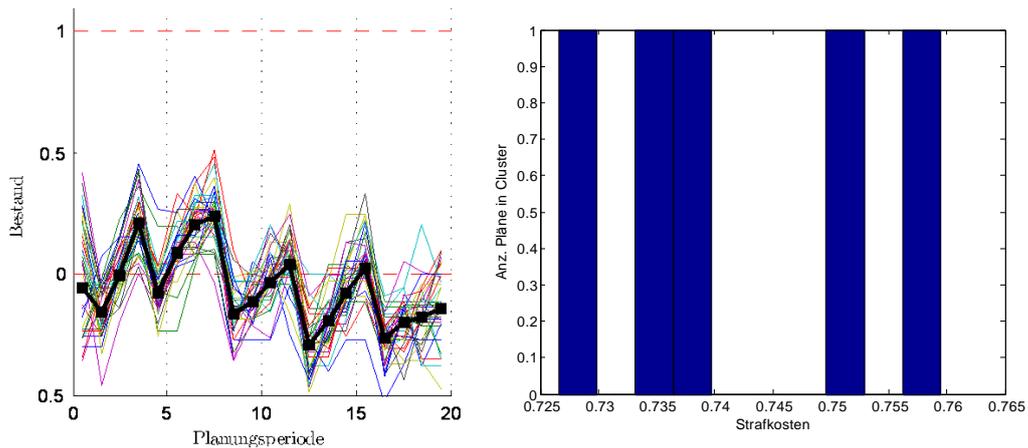
¹¹ Siehe Auswertungen zum Lernverfahren in Kap. 6.2, S. 165



(a) Planverläufe ohne Restriktionsverletzungen



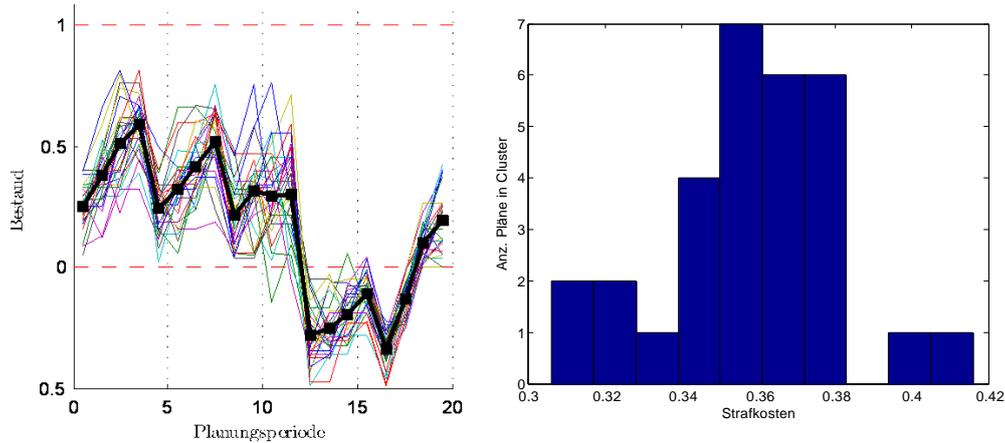
(b) Planverläufe mit wenigen geringfügigen Restriktionsverletzungen



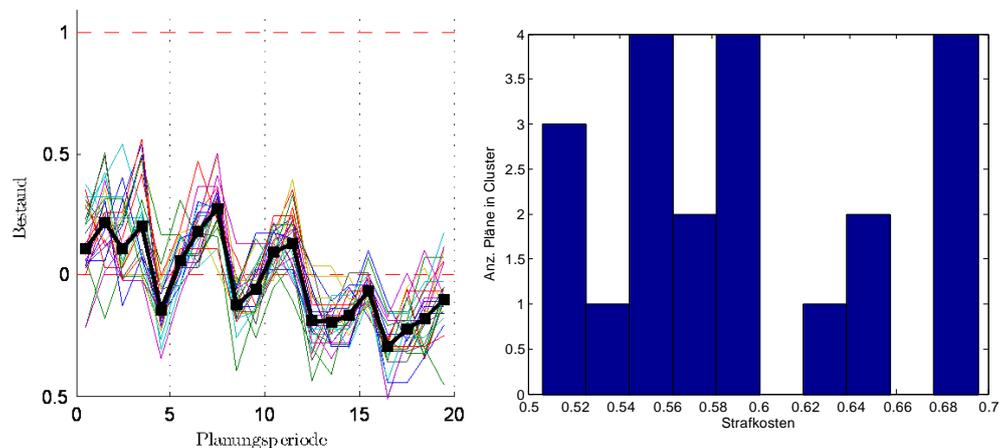
(c) Planverläufe mit zahlreichen Restriktionsverletzungen über den gesamten Planungshorizont

Abbildung 6.5.: Verteilung der Strafkosten für die Planverläufe verschiedener Cluster

6. Validierung



(a) Planverläufe mit wenigen Restriktionsverletzungen am Ende des Planungshorizontes



(b) Planverläufe mit sich aufschaukelnden Restriktionsverletzungen

Abbildung 6.6.: Verteilung der Strafkosten für die Planverläufe verschiedener Cluster (fortgesetzt)

Der charakteristische Planverlauf in Abbildung 6.6(b) zeigt beispielhaft Restriktionsverletzungen, die über die Bestellzyklen ausgeprägter werden. Durch die zahlreichen Restriktionsverletzungen streuen die Planbewertungen innerhalb des Intervalls $[0,51;0,68]$ etwas weiter. Die meisten streuen Werte im Intervall $[0,51;0,58]$, welches mit 0,07 breiter ist als die Bisherigen. Dieses Problem wird durch den Kompromiss des Clusterings hervorgerufen, wobei die Strafkostenberechnung der Pläne unterschiedliche Planbewertungen berechnet. Da im Lernverfahren durch die Bewertung des charakteristischen Planverlaufs gelernt wird, ist eine Streuung des Q-Wertes nicht zu erwarten. Dieses wäre der Fall, wenn die Q-Werte über die konkreten Planverläufe berechnet und dann auf Clusterebene aktualisiert würden.

6.1.4. Zusammenfassung

Die Analyse des Konvergenzverhaltens hat gezeigt, dass das Clustering in endlicher Zeit konvergiert. Dabei hat sich in verschiedenen Testläufen ergeben, dass nach 5 Iterationen des Clusteringverfahrens keine signifikante Verbesserung im Bezug auf den Rechenaufwand erwartet werden kann. Für die Anzahl der Cluster hat sich für 10.000 Ausgangsdatensätze ein Wert von 500 oder allgemein 5 – 10% Cluster der Trainingsdatenmenge als geeignet herausgestellt. In der Konsequenz kann der Zustandsraum des Produktionsnetzwerkes z. B. auf eine Anzahl von 500 Zuständen verdichtet werden.

Die Betrachtung der Bewertungen für die Planverläufe einzelner Cluster zeigt, dass die Zugehörigkeit mehrerer Planverläufe zu einem Cluster mit ähnlichen Bewertungen innerhalb eines Intervalls mit einer Größe von ungefähr $[0,03; 0,04]$ einhergeht. Es bestätigt, dass die Grundlage für die Berechnung von Rewards für alle Zustände eines Clusters annähernd gleich ist und charakteristische Pläne als Repräsentanten diskreter Planverläufe im Lernverfahren verwendet werden können. Die Effizienz des Lernverfahrens kann durch das Lernen auf dem abstrakten Zustandsraum gesteigert werden, da im Training weniger Zustands-/Aktionspaare bewertet werden müssen.

6.2. Lernverfahren und Training

In Kapitel 6.2.1 wird das Szenario zur Validierung der Lernfunktion und des Trainings eingeführt. Das Szenario ist so gewählt, dass die Ergebnisse des Lernprozesses nachvollzogen werden können und eine Generalisierung der Ergebnisse auf komplexe Szenarien möglich wird. Bei der Validierung werden verschiedene Konfigurationen der Lernfunktion untersucht und deren Effekte auf die Lernergebnisse diskutiert, sowie gelernte Regeln abgeleitet. Um zu zeigen, dass der Lernprozess auf dem abstrahierten Zustandsraum gleiche Ergebnisse erzeugt wie auf diskreten Zuständen, wird in Kapitel 6.2.2 ein Lernprozess mit diskreten Zuständen durchgeführt und dessen Ergebnisse mit einem Lernprozess auf den abstrahierten Zuständen der diskreten Zustände verglichen. Mithilfe der Ergebnisse der Validierung der Lernfunktion wird in Kapitel 6.2.3 die Effizienz des Trainingsprozesses analysiert. Es wird abgeschätzt, wie viel Zeit im Training bis zur Konvergenz der Q-Werte für verschiedene Szenarien benötigt wird, und diskutiert, welchen Einfluss das Clustering auf die Effizienz des Trainings ausübt.

6.2.1. Szenario zur Validierung des Lernverfahrens

Das Validierungsszenario besteht aus einer Endmontagelinie *EM* und zwei Vormontagelinien *VOR1* und *VOR2*. Es handelt sich um ein beispielhaftes Produktionssystem,

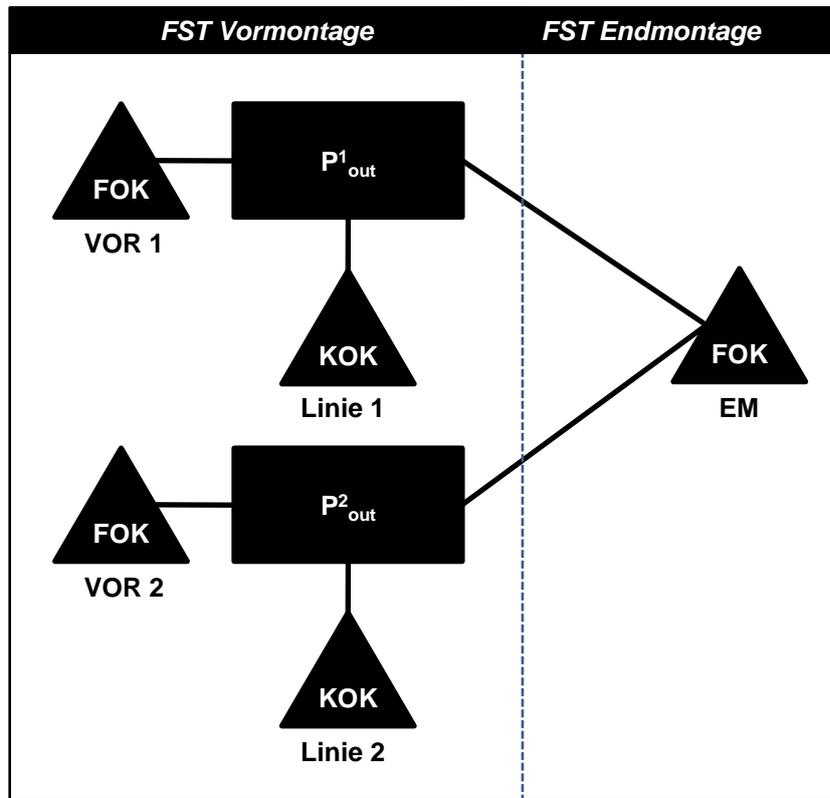


Abbildung 6.7.: Beispielproduktionsnetzwerk zur Validierung des Lernverfahrens

wie es in dieser Komplexität bei kleinen oder mittleren Unternehmen zu finden ist. Tabelle 6.2 zeigt die Konfiguration des Objektknotens *EM*. Die KOK werden, wie in Kapitel 6.1 diskutiert, gegen unendliche Kapazität geplant. Die Beschaffung zwischen Vormontage und Endmontage wird über eine (s, S) -Politik gesteuert. Die dadurch entstehende Flexibilität des Beschaffungsprozesses ermöglicht einen guten Vergleich zwischen der Gewichtung der Strafkostenfunktion für lokale und globale Verfahren im Lernprozess, da in die Berechnung der Strafkosten bei beiden stets die gleichen Strafkostenfaktoren einfließen.

Das Szenario wird über festgelegte Ausgangszustände validiert, um die Ergebnisse der einzelnen Parameterkonfigurationen der Lernfunktion vergleichen zu können. Es wird

Tabelle 6.2.: Konfiguration *EM* im Lernszenario

$p(k)_{min}^{sup}$	$p(k)_{logmin}^{sup}$	$p(k)_{mintemp}^{sup}$	$p(k)_{max}^{sup}$	$p(k)_{maxtemp}^{sup}$	$p(k)_{absmax}^{sup}$
100	0	100 (initial)	550	550 (initial)	800
200	0	200 (initial)	400	400 (initial)	600
300	0	300 (initial)	800	800 (initial)	900

auf einem diskreten Zustand gelernt und zum Vergleich der Ergebnisse auf dessen Zustandscluster. Da das Training durch die Verwendung eines realitätsnahen Ausgangszustandes zielgerichtet konvergieren kann, können die entstehenden Ergebnisse, wie die Dauer des Konvergenzprozesses, ausgewertet werden.

Durch den Vergleich der Lernergebnisse des Lernens auf Zuständen und Clustern soll evaluiert werden, ob der Lernprozess für beide Varianten äquivalente Regeln erzeugt. Es wurde gezeigt, dass das Lernverfahren trotz Abstraktion effektiv lernt und die Effizienz des Lernsystems, wie in der Problemstellung gefordert, durch das Lernen auf dem reduzierten Zustandsraum erhöht werden kann.

6.2.2. Validierung der Lernfunktion

In den folgenden Szenarien wird erst auf Zustandsebene und dann auf Clusterebene gelernt. Es wird gegenübergestellt, ob und wann lokale oder globale Aktionen bevorzugt werden und wie dieses mit den Parametereinstellungen der Lernfunktion beeinflusst werden kann. Beim Lernen auf Clusterebene kann die Effizienz der Lernfunktion überprüft werden. Hier wird betrachtet, ob trotz Abstraktion zweckdienliche Q-Werte gelernt werden können. Ist dieses der Fall, so kann der Lernprozess trotz Clustering als effektiv bezeichnet werden, da problemspezifische Regeln gelernt werden. Da der abstrahierte Zustandsraum wesentlich schneller durch das Q-Learning analysiert werden kann als der vollständige Zustandsraum, wird die Effizienz des Lernverfahrens deutlich gesteigert.

6.2.2.1. Szenario 1 - Effektivität der Lernfunktion

In Szenario 1 soll die Effektivität der Lernfunktion überprüft werden. Hierzu wurde das Lernsystem auf Zustandsebene trainiert, um festzustellen, ob die Lernfunktion nachvollziehbare und korrekte Q-Werte liefert. Es wurde der in Abbildung 6.8 dargestellte, ungültige Plan zugrunde gelegt. In diesem existieren zwei Restriktionsverletzungen, in den Perioden $RV(p(5)) = +50$ und $RV(p(9)) = +100$.

Zur Auflösung dieses ungültigen Zustandes können entweder die Restriktionsgrenzen der einzelnen Perioden angehoben oder eine Bedarfsreduzierung bei einem oder anteilig bei beiden Lieferanten VOR1 und VOR2 durchgeführt werden. Bei der Durchführung des Trainings für Szenario 1 wurden durch die verwendeten Planungsverfahren die in Tabelle 6.3 dargestellten Planungsergebnisse erzielt.

Der Lernprozess in Szenario 1 wurde mit zwei Konfigurationen der Rewardfunktion durchgeführt. Diese sind in Tabelle 6.4 dargestellt. Dabei wurden die Gewichte der Gesamtstrafkostenfunktion so gewählt, dass im ersten Fall lokale Verfahren höhere Strafkosten erzielen und im zweiten Fall Änderungen des Materialflusses durch ein

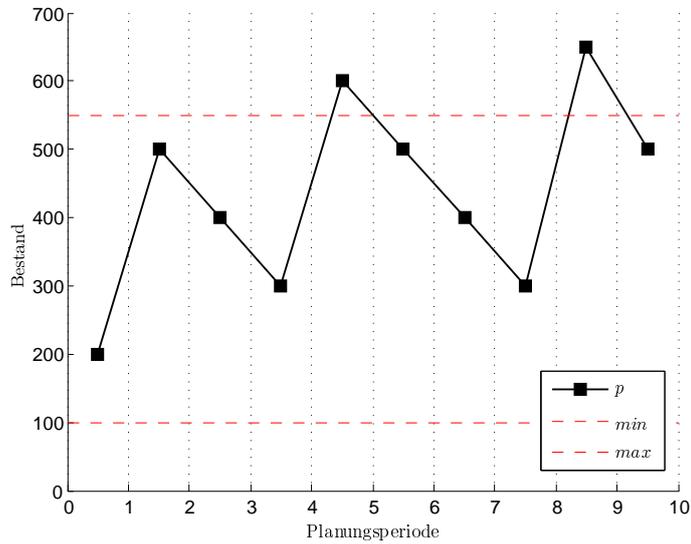


Abbildung 6.8.: Ausgangsplan Szenario 1

Tabelle 6.3.: Planungsergebnisse Szenario 1.1

Planungsverfahren	Art	Ergebnis
Mindestbestandserhöhung Dynamisch	lokal <i>EM</i>	$p(5)_{maxemp}^{sup} = +50$ $p(9)_{maxemp}^{sup} = +100$
Mindestbestandserhöhung Prozentual	lokal <i>EM</i>	$p(5)_{maxemp}^{sup} = +50$ $p(9)_{maxemp}^{sup} = +100$
Mindestbestandserhöhung Absolut	lokal <i>EM</i>	$p(5)_{maxemp}^{sup} = +60$ $p(9)_{maxemp}^{sup} = +110$
NB Reduzierung	global <i>VOR1</i>	$p(5)^{mf} = NB - 50$ $p(9)^{mf} = NB - 50$
NB Reduzierung	global <i>VOR1</i>	Ablehnung

Tabelle 6.4.: Parameter im Lernszenario „Effektivität“

Lokal	ω_{min}^{loc}	$\omega_{mintemp}^{loc}$	ω_{max}^{loc}	$\omega_{maxtemp}^{loc}$	ω_{cost}^{loc}
	0,40	0,10	0,40	0,1	0,1
Materialfluss	ω_{min}^{proc}	ω_{max}^{proc}	ω_{cost}^{proc}		
	0,45	0,45	0,10		
Gesamtfunktion	ω_{aon}^{loc}	ω_{aon}^{glob}	ω_{aon}^{rv}	ω_{aon}^{oop}	
Fall 1	0,20	0,40	0,10	0,30	
Fall 2	0,50	0,30	0,10	0,10	

globales Verfahren mit höherer Gewichtung bestraft werden. Die Auswahl eines Änderungsverfahrens wurde mit der ε -Greedy-Strategie¹² durchgeführt. Der Lernfaktor α wurde mit einem Wert von 0,2 belegt, um das Konvergieren der Q-Werte zu unterstützen und Effekte von Ausreißern mit stark abweichenden Rewards auf die Q-Werte zu dämpfen.¹³ Dieses kann z. B. geschehen, wenn durch die Auswahlstrategie eine Aktion durchgeführt wird, die schlechte Ergebnisse erzielt. Für γ wurde 0,9 gewählt, um eine graduelle Abstufung der Q-Werte zum Zielzustand zu erreichen.¹⁴

1. Fall

Abbildung 6.9 zeigt das Lernergebnis für den Fall 1 des Szenarios 1. In der Abbildung 6.9(a) ist zu erkennen, dass die lokalen Verfahren *ABSOLUT*, *DYNAMISCH* und *PROZENTUAL* im Q-Wert dicht an dem Wert 0,9 liegen, während der Q-Wert der globalen Aktion *VOR1* bei ungefähr 0.85 eingeschwungen ist. Da in diesem Szenario bei der Koordination zwischen *EM* und *VOR2* stets die Anfrage ablehnt wird, bleibt der Q-Wert für diese Aktion bei 0¹⁵ und wird in der Grafik nicht berücksichtigt.¹⁶

Im Detailausschnitt 6.9(b) sind nur noch die Q-Werte der lokalen Verfahren dargestellt, um die Unterschiede der einzelnen Q-Werte sehen zu können. Es wird deutlich, dass das Verfahren *ABSOLUT* schlechter bewertet wurde als die Verfahren *DYNAMISCH* und *PROZENTUAL*. Der Grund dafür ist das untereinander abweichende Planungsergebnis der einzelnen Verfahren. Wie in Tabelle 6.3 zu sehen ist, planen *DYNAMISCH* und *PROZENTUAL* in den ungültigen Perioden mengenmäßig nur so viel um, wie zur Beseitigung der Restriktionsverletzung benötigt wird. Hierzu verschieben sie die Restriktionsgrenze um den Überschuss (+50, +100) nach oben. *ABSOLUT* führt eine

¹²Vgl. Kap. 5.3.1.3, S. 138

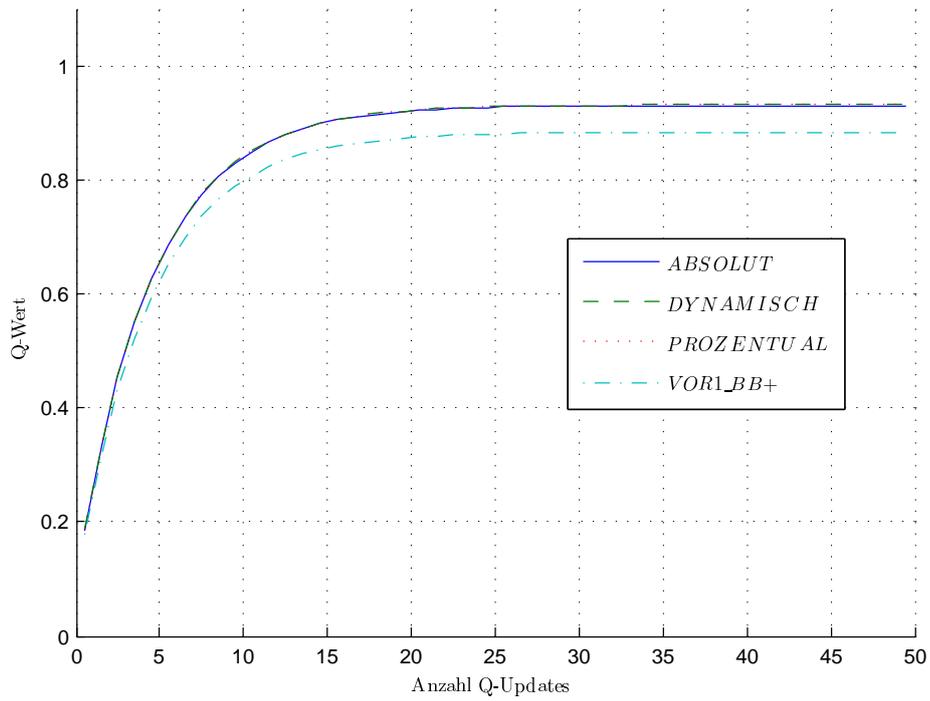
¹³Vgl. Kap. 5.3.2.1, S. 139

¹⁴Vgl. Kap. 5.2.10, S. 130

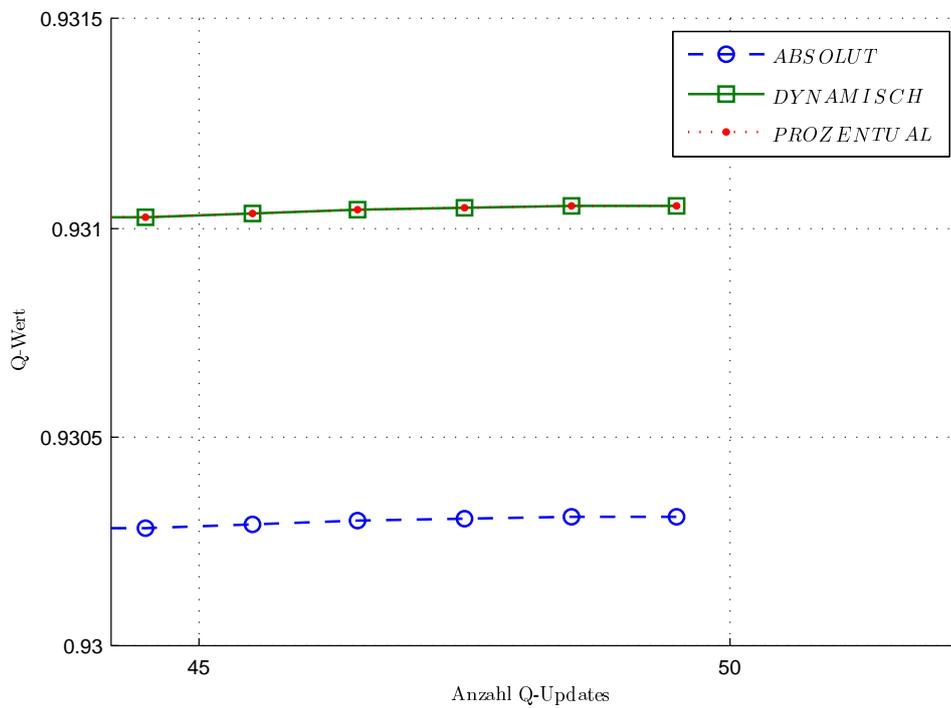
¹⁵ $Q(s_i, a_i) = Q(s_i, a_i) + \alpha [reward_{s_i} + \gamma \max(Q(s_{i+1}, a) - Q(s_i, a_i))] = 0 + 0,2 [0 + 0,9 \cdot 0 - 0] = 0$

¹⁶Wären auf dem Zustand von *VOR2* Q-Werte mit $Q_i > 0$ zu verzeichnen, so hätte hier ein Q-Wert ermittelt werden können, der aufgrund des Rewards der Höhe 0 der durchgeführten Aktion gering ausgefallen wäre.

6. Validierung



(a) Übersicht der Q-Wert-Entwicklung



(b) Konvergenzbereich der Q-Werte

Abbildung 6.9.: Szenario 1, Fall 1 - Lokale Aktion wird bevorzugt

Verschiebung der Restriktionsgrenze um $(+60, +110)$ in den Perioden durch. Die Aktion *ABSOLUT* erzielt im Vergleich zu *DYNAMISCH* und *PROZENTUAL* schlechtere Planungsergebnisse, da sie zur Erzeugung eines gültigen Planes größere Änderungen am ungültigen Plan vornimmt als die anderen beiden Aktionen. Dieses bedeutet, dass die Strafkosten für *ABSOLUT* höher ausfallen und der erzielte Reward für die Aktionen *DYNAMISCH* und *PROZENTUAL* besser ist. Der Q-Wert von *DYNAMISCH* und *PROZENTUAL* stabilisiert sich auf höherem Niveau als der Q-Wert für *ABSOLUT*.¹⁷

Die Rewards der lokalen Aktionen liegen nahe zusammen, da nur geringfügige Veränderungen am Plan von *EM* durchgeführt werden mussten, um die Restriktionsverletzungen im ungültigen Plan zu beseitigen. Es konnte trotz der geringen Unterschiede der ungültigen und der gültigen Pläne der Reward so berechnet werden, dass korrekte Q-Werte für die lokalen Aktionen gelernt werden konnten. Der Grund für das schlechtere Ergebnis der globalen Aktion *VOR1* ist mit der Gewichtung der Lernfunktion zu begründen. Die Strafkosten für den Materialfluss und die zusätzlich anfallenden Strafkosten des Beschaffungsprozesses, fallen durch die höhere Gewichtung von $\omega_{aon}^{glob} = 0.40$ und $\omega_{aon}^{oop} = 0.30$ zu $\omega_{aon}^{loc} = 0.20$ und $\omega_{aon}^{rv} = 0.10$ bei der Rewardberechnung deutlich stärker ins Gewicht, als es bei den Strafkosten der lokalen Verfahren der Fall war. Die nach Abschluss des Trainings gelernten Regeln für die Zustands-/Aktionspaare lauten:

- ID1* WENN *plan* = *z1* DANN *dynamisch* ODER *prozentual*
ID2 WENN *plan* = *z1* DANN *absolut*
ID3 WENN *plan* = *z1* DANN *vor1*

2. Fall

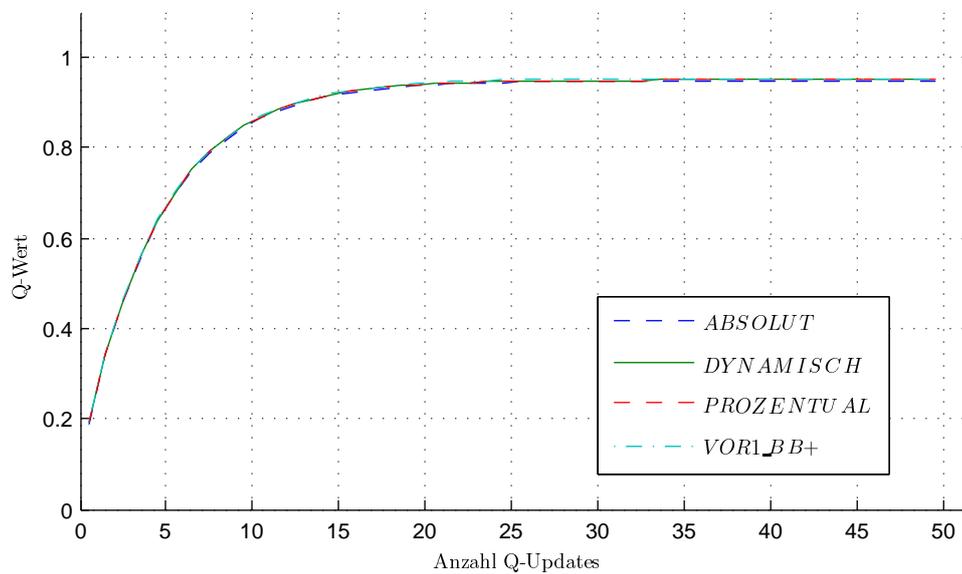
Für die Validierung von Fall 2 wurden die Gewichtungparameter der Lernfunktion wie in Tabelle 6.4 dargestellt geändert. Hier wurden die anfallenden Strafkosten für lokale Aktionen stärker bewertet, als die anfallenden Strafkosten für globale Aktionen.

Das Ergebnis des Lernprozesses ist in Abbildung 6.10 dargestellt. In Abbildung 6.10(a) ist zu sehen, dass die Q-Werte der Aktionen aufgrund der geringen Unterschiede in den Strafkosten nah beieinanderliegen. In Abbildung 6.10(b) sind die Unterschiede zwischen den Aktionen detaillierter aufgelöst. Der Q-Wert von *ABSOLUT* ist am schlechtesten bewertet. *DYNAMISCH* und *PROZENTUAL* liegen gleich auf und *VOR1* wurde am besten bewertet.¹⁸

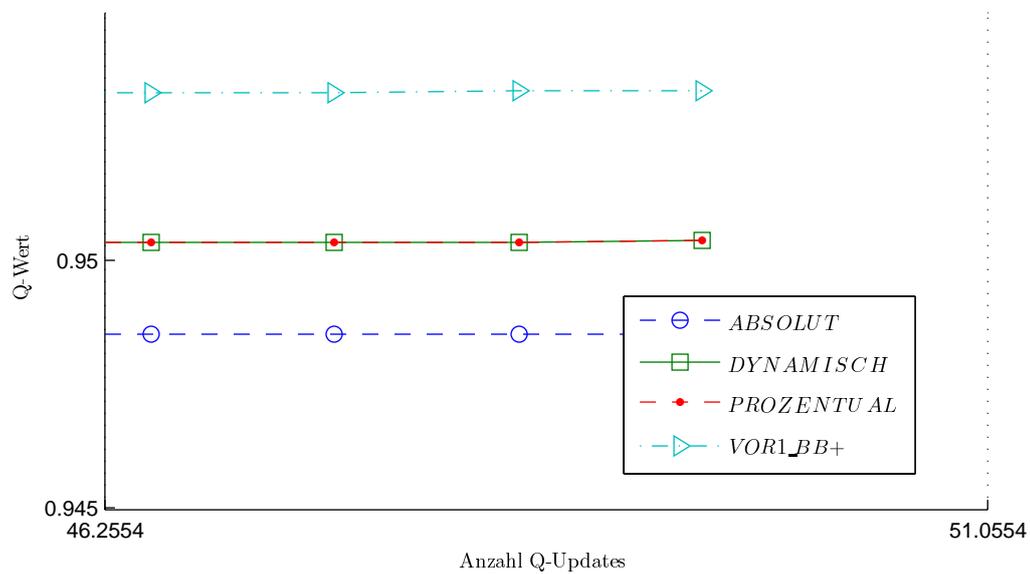
¹⁷Siehe Formel (5.18), S. 106

¹⁸*VOR2* wurde herausgelassen, da der Q-Wert 0 beträgt, wie oben erläutert.

6. Validierung



(a) Übersicht der Q-Wert-Entwicklung



(b) Konvergenzbereich der Q-Werte

Abbildung 6.10.: Szenario 1, Fall 2 - Globale Koordination wird bevorzugt

Da *ABSOLUT* einen gültigen, aber schlechteren Plan erzeugt wie *DYNAMISCH* und *PROZENTUAL* wurde dieser schlechter bewertet. Der Q-Wert fällt niedriger aus. *VOR1* führt eine globale Aktion durch, für die Strafkosten anfallen. Bei der Q-Wert-Berechnung schlagen diese aufgrund der geringen Gewichtung und des höheren erzielten Rewards niedriger zu Buche. *VOR1* weist am Ende des Trainingsprozesses den besten Q-Wert auf. Die gelernten Regeln für die Zustands-/Aktionspaare lauten:

ID1 WENN plan = z1 DANN vor1

ID2 WENN plan = z1 DANN dynamisch ODER prozentual

ID3 WENN plan = z1 DANN absolut

Fazit zur Effektivität der Lernfunktion

Die Gewichtungsfaktoren und die Lernfunktion auf Zustandsebene erfüllen ihre intendierten Funktionen. Die Gewichtungsfaktoren ermöglichen durch Gewichte an der Gesamtstrafkostenfunktion eine implizite Priorisierung von lokalen oder globalen Aktionen, was sich in der Abstufung der gelernten Q-Werten widerspiegelt. Die Lernfunktion arbeitet effektiv, da sie korrekte Ergebnisse erzielt.

Die höhere Gewichtung globaler Strafkosten kann z. B. für Unternehmen mit hoher Wertschöpfungstiefe sinnvoll sein. Das verfolgte Ziel könnte sein, ungültige Zustände über Eigenerzeugnisse oder aus dem Lagerbestand aufzulösen und durch das Lernsystem Regeln zur Priorisierung lokaler Änderungsplanungsverfahren zu lernen. Eine höhere Gewichtung der lokalen Strafkosten ist z. B. für Unternehmen sinnvoll, die ihre Zulieferteile Just-In-Time verarbeiten und ein kleines Lager besitzen. Die Auflösung ungültiger Zustände wird hier stark beschaffungsorientiert durchgeführt. Es könnte das Ziel verfolgt werden, Regeln für die Lieferantenanbindung bzw. implizit zu deren Bewertung zu lernen.

6.2.2.2. Szenario 2 - Effizienz der Lernfunktion

Im vorherigen Kapitel wurde die Effektivität des Lernverfahrens validiert, indem gezeigt wurde, dass die Lernfunktion die richtigen Regeln für ein Szenario auf Zustandsebene lernen kann. Würde der Lernprozess stets direkt auf den Zuständen des Produktionsnetzwerkes durchgeführt, so könnte dieser kaum oder gar nicht in endlicher Zeit beendet werden. Es wäre darüber hinaus nicht möglich, alle erforderlichen Q-Werte in einer Datenbank zu speichern.¹⁹

¹⁹Siehe Kap. 2.3.1, S. 41

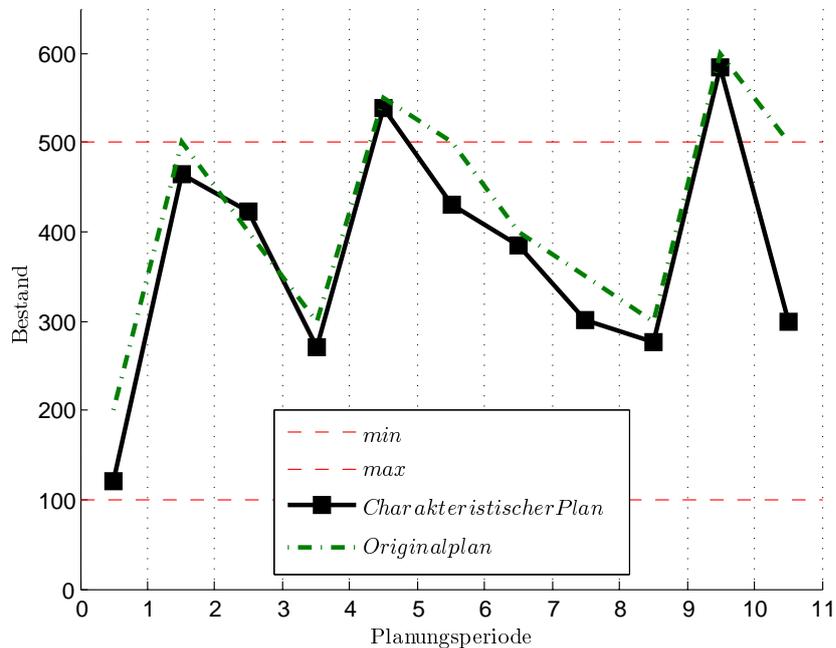


Abbildung 6.11.: Originalplan und Centroid des Clusters

Aus diesem Grund wurde das Clusteringverfahren entwickelt, um den Zustandsraum so zu abstrahieren, dass der Lernprozess auf dem abstrahierten Zustandsraum effizient durchführbar ist. Um die dadurch erreichbare Effizienzsteigerung des Lernverfahrens zu zeigen, muss geprüft werden, ob trotz des Clustering korrekte Lernergebnisse erzielt werden können. Die Effektivität der Clusterfunktion muss untersucht werden.

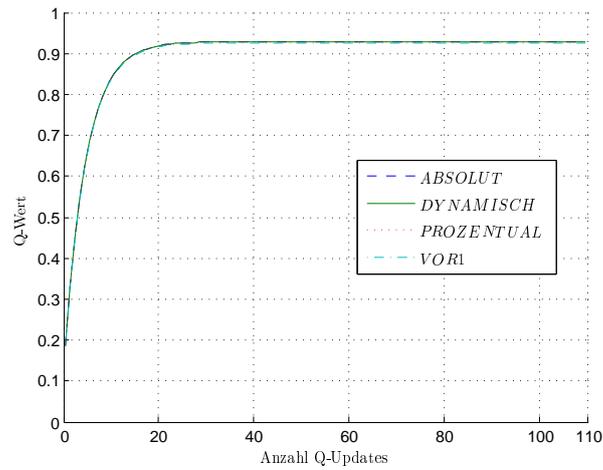
Als Vergleichswert dienen die Ergebnisse des Lernszenarios auf Zustandsebene aus Kapitel 6.2.2.1. Hier wird als Referenzplan der gleiche Plan gewählt wie in Szenario 1. Zu diesem Plan wird durch das Clusterverfahren während des Trainings der zugehörige Cluster ausgewählt.²⁰

1. Fall

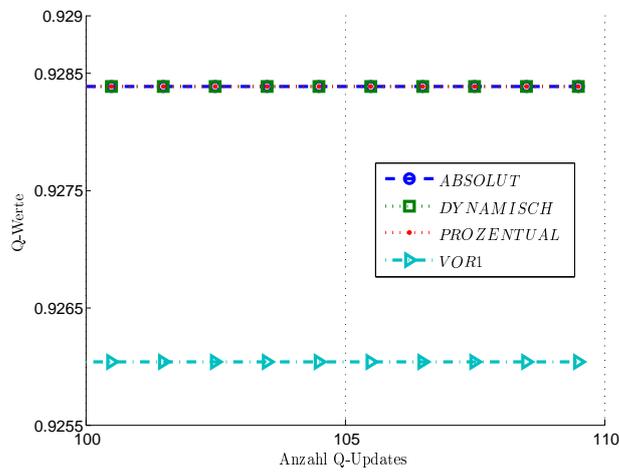
Die Lernfunktion wird so gewichtet, dass im Lernprozess Strafkosten lokaler Aktionen geringer bestraft werden. Es wird das Lernziel verfolgt, ungültige Zustände eher durch lokale Planungsverfahren zu bearbeiten und weniger Bedarfs- oder Angebotsänderungen mit den Lieferanten oder Kunden zu koordinieren. In Abbildung 6.12 ist das Ergebnis des Lernprozesses in verschiedenen Detailausschnitten dargestellt.

Aus Abbildung 6.12(a) ist gut ersichtlich, dass die Q-Werte wie beim zustandsbasierten Lernen nach ca. 100 Schritten konvergieren. Die Änderungen im hinteren Drittel des Graphen sind minimal und werden, wie die Differenzierung der einzelnen Q-Werte

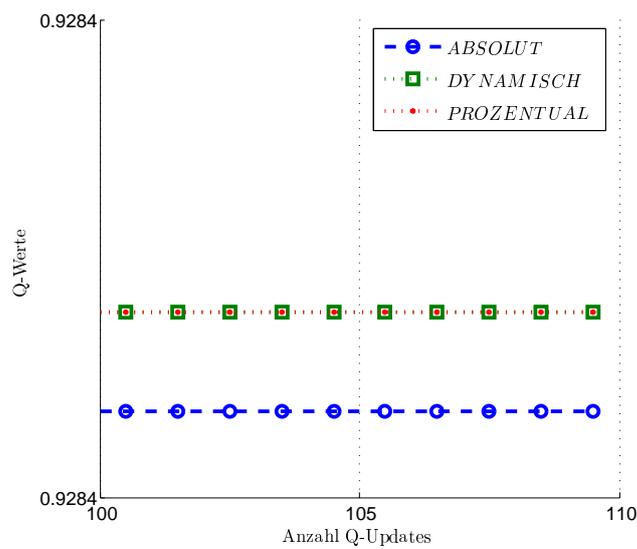
²⁰Hierzu wird die gleiche Funktion genutzt, die während der Zuweisung von Plänen zu Clustern im Clustering verwendet wurde. Weiteres siehe Kap. 5.3.3.1, S. 142 ff.



(a) Übersicht der Q-Wert-Entwicklung



(b) Konvergenzbereich der Q-Werte (Details 1)



(c) Konvergenzbereich der Q-Werte (Details 2)

Abbildung 6.12.: Szenario 2, Fall 1 - Lokale Aktion wird bevorzugt

je Aktion, in dieser Abbildung nicht detaillierter dargestellt. In Abbildung 6.12(b) ist dann gut erkennbar, dass die lokalen Verfahren *ABSOLUT*, *DYNAMISCH* und *PROZENTUAL* einen besseren Q-Wert aufweisen als die globale Aktion *VOR1*. Die Q-Werte unterscheiden sich zwischen dem lokalen und dem globalen Verfahren um rund 0,003 Punkte. In Abbildung 6.12(c) wurden die Q-Werte der lokalen Aktionen so weit aufgelöst, dass eine Abstufung sichtbar wird. Es ist zu erkennen, dass das Verfahren *ABSOLUT* schlechter bewertet wurde als die Verfahren *PROZENTUAL* und *DYNAMISCH*. Diese weisen einen Wert vergleichbar mit dem in Szenario 1 auf.

In Szenario 2 konnte trotz des Lernens auf Clustern das gleiche Lernergebnis erzielt werden wie in Szenario 1. Durch die gleichen Abstufungen der Q-Werte können dieselben Regeln wie in Szenario 1 erzeugt werden, die über die Cluster und nicht über die Zustände referenziert werden.

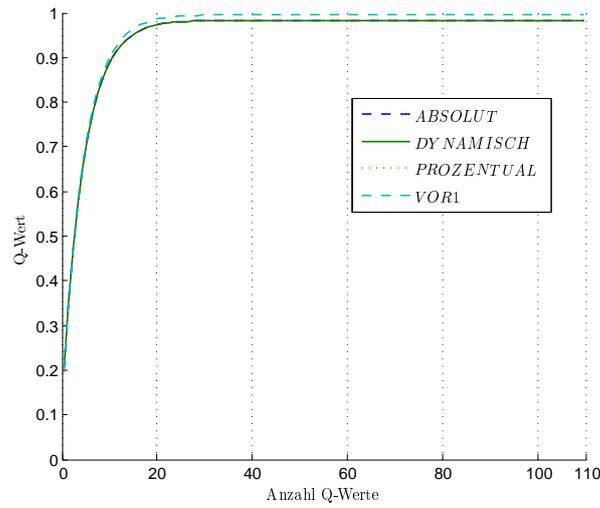
ID1 WENN cluster = c1 DANN dynamisch ODER prozentual
ID2 WENN cluster = c1 DANN absolut
ID3 WENN cluster = c1 DANN VOR1

2. Fall

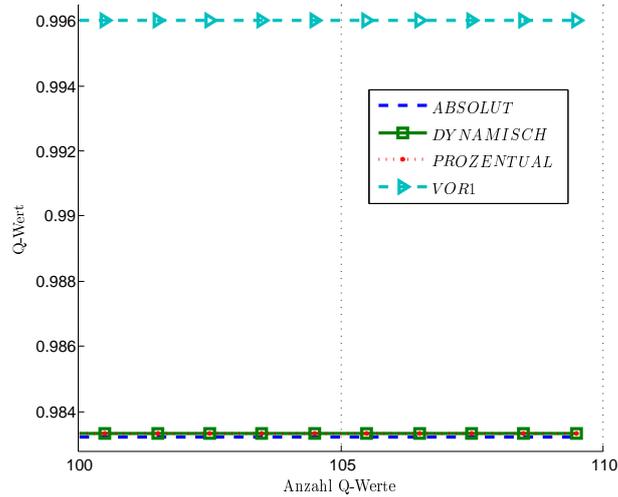
In Fall 2 wurden globale Aktionen implizit bevorzugt, indem die Gewichte der Strafkostenfunktion konfiguriert wurden. Die Ergebnisse des Lernprozesses sind in Abbildung 6.13 dargestellt. In Abbildung 6.13(a) ist gut erkennbar, dass die Q-Werte mit Fortschreiten des Trainings konvergieren. Die Q-Werte liegen wiederum nah zusammen, was mit der geringen Anzahl und Höhe der Restriktionsverletzungen zu begründen ist. Die Rewards weisen je Aktion geringe Differenzen auf. In Abbildung 6.13(b) ist gut zu erkennen, dass die Aktion *VOR1* einen deutlich besseren Q-Wert erzielt als die anderen Aktionen. Die implizite Bevorzugung globaler Aktionen in der Strafkostenberechnung hat Wirkung gezeigt. In Abbildung 6.13(c) sind die Q-Werte der lokalen Verfahren noch detaillierter dargestellt. Dieses Ergebnis ist mit den bisherigen Auswertungen im Einklang, da *ABSOLUT* schlechter bewertet wird als *DYNAMISCH* und *PROZENTUAL*.

Es werden folgende Regeln gelernt:

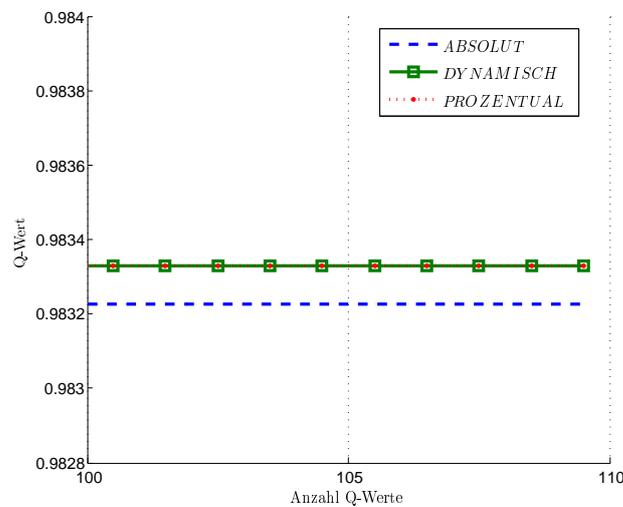
ID3 WENN cluster = c1 DANN vor1
ID1 WENN cluster = c1 DANN dynamisch ODER prozentual
ID2 WENN cluster = c1 DANN absolut



(a) Übersicht der Q-Wert-Entwicklung



(b) Konvergenzbereich der Q-Werte (Details 1)



(c) Konvergenzbereich der Q-Werte (Details 2)

Abbildung 6.13.: Szenario 2, Fall 2 - Globale Koordination wird bevorzugt

Fazit Effizienz der Lernfunktion

Die Untersuchungen zeigen, dass das Training auf Clusterebene trotz abweichender Q-Werte die gleichen Regeln hervorgebracht hat wie das Lernen auf Zuständen. Diese Abweichung ist mit der Abstraktion der Zustände begründbar. Da auf Clusterebene nicht der Originalplan zum Lernen verwendet wird, unterscheiden sich die Q-Werte auf Clusterebene von den zustandsbasierten Q-Werten. Für das Lernen auf Clusterebene kann eine positive Bilanz gezogen werden. Die Abstraktion auf Clusterebene ist im Lernprozess effektiv nutzbar. Die Gesamteffizienz des Lernverfahrens steigert sich, da wesentlich weniger Zustände durch das Lernverfahren bewertet werden müssen. Die Problemkomplexität sinkt.

6.2.3. Validierung des Trainingsprozesses

Die zentrale Aufgabe des Trainingsprozesses ist es, unter gegebenen Voraussetzungen effizient ein verwendbares Lernergebnis zu erzielen. Während in der Analyse im vorigen Kapitel 6.2.2 nur exemplarische Lernepisoden untersucht wurden, werden zum Validieren des Trainingsprozesses viele Lernepisoden durchgeführt. Im Clustering wurden die benötigten Ausgangsdaten erzeugt. Es wurden mit den generierten Daten 50 Lernepisoden durchgeführt.²¹

Zur Validierung des Trainings wurden insbesondere die benötigten Zeiten für die Durchführung einzelner Lernepisoden gemessen. Es wurde ermittelt, wie viel Zeit bei der Verwendung von Clustern im Lernprozess verloren geht, da je Lernschritt die Pläne ihren Clustern zugeordnet werden müssen. Je mehr Cluster verwendet werden, desto größer ist der Zeitverlust. Ziel war es, mithilfe dieser Einzelwerte eine Abschätzung der Trainingsdauer für unterschiedliche Produktionsnetzwerkgrößen mit unterschiedlichen Clustermengen durchzuführen.

6.2.3.1. Konvergenz des Verfahrens

Es wurden die Werte für die Dauer einer einzelnen Lernepisode mit lokalen und globalen Planungsverfahren gemessen.²² Dabei wurde für die Durchführung einer Planungsaktion ein durchschnittlicher Wert von unter 1 *ms* gemessen.

²¹Details zur Erzeugung von Trainingsdaten für die Validierung siehe in Kap. 6.1.1, S. 154

²²Zur Implementation wurde Java verwendet. Details siehe Anhang C, S. 213 ff. Das Lernverfahren wurde vollständig lokal auf einem Agenten ausgeführt. Der verwendete Computer war ein Intel Core 2 Duo CPU 2.20 GHz mit 2048 MB RAM.

Die konstante und geringe Zeit zur Durchführung eines Änderungsplanungsverfahrens²³ ist mit dessen Laufzeit von $O(n)$ zu begründen. Die Anzahl der Planungsperioden bestimmt dabei den Wert von n , da mit steigender Anzahl der Planungsperioden der Berechnungsaufwand der Verfahren linear steigt. Drei der betrachteten Verfahren²⁴ wiesen eine Laufzeit von $O(n^2)$ auf, da dort zusätzlich eine Schleife durchlaufen werden musste, die die Laufzeit je Periode verzögert. Die effiziente Ausführbarkeit von Änderungsplanungsverfahren unterstützt die Effizienz des Lernverfahrens.

Tabelle 6.5.: Zuordnungsdauer von Plan zu Cluster

Anzahl Cluster	Zuordnungsdauer [ms]
500	5.5 ms
1000	11 ms
5000	56 ms
10000	113 ms

Zum Laden von 1500 Clustern in die im Programm verwendete Datenstruktur bzw. den Programmspeicher wurde eine Zeit von 3237 ms oder rund 3 Sekunden benötigt. Für die Auswahl eines Clusters zu einem Plan wurden die in Tabelle 6.5 dargestellten Zeiten in [ms] gemessen. Die Messungen wurden mit dem in Kapitel 6.1.2 empfohlenen Wert von 500 Clustern begonnen und sukzessive gesteigert. Die Auswertung in Tabelle 6.5 zeigt, dass die Zuweisungszeit für Pläne zu Clustern mit steigender Anzahl der Cluster linear wächst und gut lösbar ist. Für den in Kapitel 6.4 ermittelten Break-even-Punkt bei 500 Clustern ist die Zuweisungszeit von rund 5.5 ms akzeptabel.

Die bis zur Konvergenz der Q-Werte erforderliche Anzahl an Q-Updates wurde über 50 Lernepisoden unter Verwendung von 5 verschiedenen Planungsverfahren ermittelt. Abbildung 6.14 zeigt die bis zur Konvergenz im Trainingsprozess benötigte Anzahl der Q-Updates in einem Histogramm. Die Werte wurden zur Auswertung über die verwendeten Planungsverfahren je Trainingslauf gemittelt und im Histogramm aufgetragen. Für die Anzahl benötigter Q-Updates bis zur Konvergenz des Q-Werte wurde im Mittel ein Wert von 524 Lernschritte berechnet. Die Konvergenz war in diesem Experiment dann erreicht, wenn keine messbare Änderung der Q-Werte mehr zu verzeichnen war.²⁵

²³Siehe in [Hei06], Anhang B als Referenz einer umfassenden Anzahl möglicher Änderungsplanungsverfahren.

²⁴Ebd.

²⁵In der Implementation wurde zur Verarbeitung der Q-Werte mit doppelter Genauigkeit (Datentyp *DOUBLE*) gearbeitet.

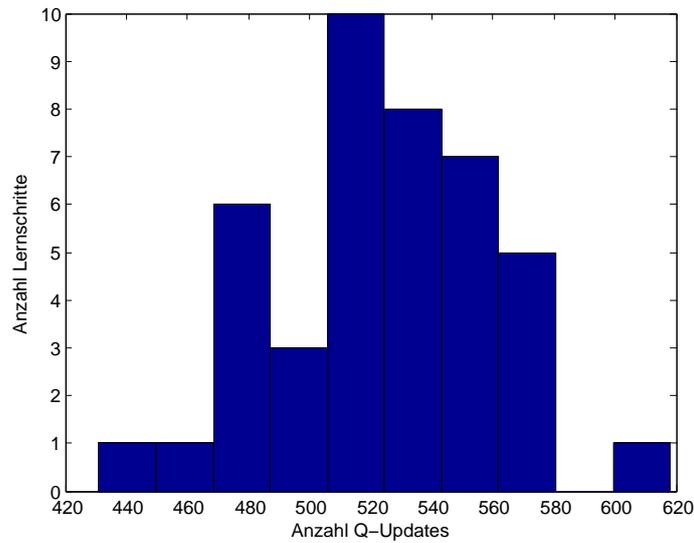


Abbildung 6.14.: Benötigte Lernschritte im Training

6.2.3.2. Dauer des Trainings bei variierender Zustandsraumgröße

Die Trainingsdauer vervielfacht sich durch die Multiplikation verschiedener Faktoren, deren Zusammenwirken Einfluss auf die Trainingszeit hat. Diese sind:

- Anzahl verfügbarer Aktionen
- Anzahl der Objektknoten im Produktionsnetzwerk
- Zeit für die Durchführung einer Änderungsplanung
- Anzahl der durchschnittlich benötigten Lernepisoden bis zur Konvergenz der Q-Werte

Aus der Anzahl verfügbarer Aktionen und der Anzahl der Objektknoten ergibt sich die maximale Anzahl der Q-Werte, die im Training verarbeitet werden. Q-Werte werden während des Trainings für jede ausgewählte und durchgeführte Aktion je Cluster erzeugt. Die Anzahl erzeugter Q-Werte bestimmt die Anzahl verfügbarer Regeln nach Abschluss des Trainings. Je mehr Aktionen im Training zur Verfügung stehen und je mehr Objektknoten im Training verwendet werden, desto länger ist die Trainingsdauer.

Abbildung 6.15 zeigt eine Abschätzung der Trainingsdauer bis zur Konvergenz der Q-Werte. Zur Bestimmung der Anzahl der benötigten Lernepisoden wurde die ermittelte Durchschnittsdauer einer Lernepisode aus Kapitel 6.2.3.1 verwendet. Der Berechnung wurden 22 verfügbare Aktionen zugrunde gelegt. Die benötigte Trainingsdauer wurde dann für jeweils 1, 5 und 10 Objektknoten bei unterschiedlich hoher Clusteranzahl ermittelt.

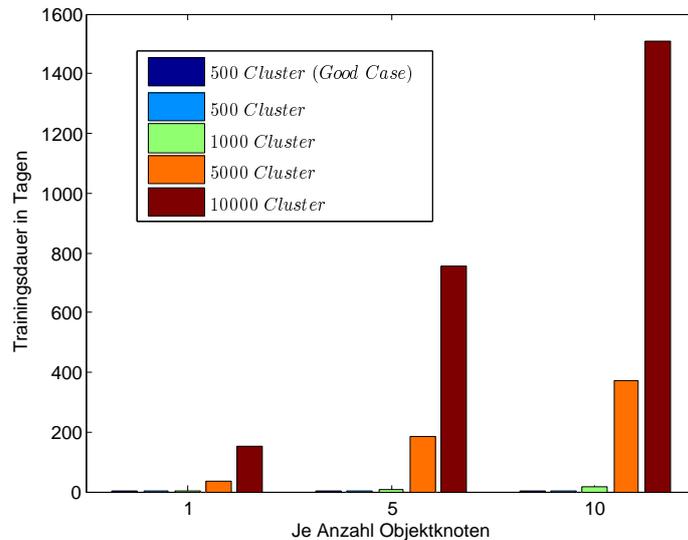


Abbildung 6.15.: Laufzeitabschätzung des Trainings bei verschiedenen Problemgrößen
(1)

Dem Balkendiagramm aus Abbildung 6.15 ist zu entnehmen, dass die ermittelte Trainingsdauer mit zunehmender Anzahl Objektknoten ansteigt. Die Anzahl der verwendeten Cluster wirkt sich verlängernd auf die Trainingsdauer aus. Während bei 500 Clustern und 10 Objektknoten²⁶ die Trainingszeit rund 88 Stunden oder ca. 3,5 Tage beträgt, so steigt sie bei 1.000 Clustern und 10 Objektknoten auf einen Wert von rund 14 Tagen an. Während bei 5.000 Clustern ca. ein Jahr zum Training benötigt wird, so werden bei 10.000 Clustern und gleicher Anzahl Objektknoten knapp 4 Jahre zum Lernen der Regeln *aller* Q-Werte benötigt. Der starke Anstieg ist mit der erhöhten Zugriffszeit auf die Cluster für eine steigende Clusteranzahl zu begründen. Könnte die Zugriffszeit für 10.000 Cluster auf einen Wert wie bei 5.000 Clustern reduziert werden, so würde die Lernzeit deutlich von 4 Jahren auf ca. 73 Tage sinken.

Die gemittelte Dauer einer Lernepisode mit durchschnittlich 524 Lernschritten kann als Worst-Case-Szenario betrachtet werden. Stichproben haben ergeben, dass sich in den meisten Fällen nach ca. 50 oder weniger Lernschritten im Best-Case ein aussagekräftiger Q-Wert zeigt. Nimmt man hier eine konservativ gewählte Größe von durchschnittlich 100 Lernschritten je Lernepisode bis zum Einschwingen der Q-Werte an, so kann die mittlere Lernzeit als deutlich verkürzt betrachtet werden. Abbildung 6.16 zeigt die Auswertung aus Abbildung 6.15 im Detail.

Für den Good-Case von 10 Objektknoten und der empfohlenen Anzahl von 500 Clustern beträgt die Lernzeit knapp 17 Stunden. Für 5 Objektknoten kann die Zeit auf 8 Stunden und für 1 Objektknoten auf 2 Stunden reduziert werden. Die benötigte Zeit

²⁶Wird hier als angemessene Anzahl von Objektknoten in einem durchschnittlichen Produktionsnetzwerk betrachtet

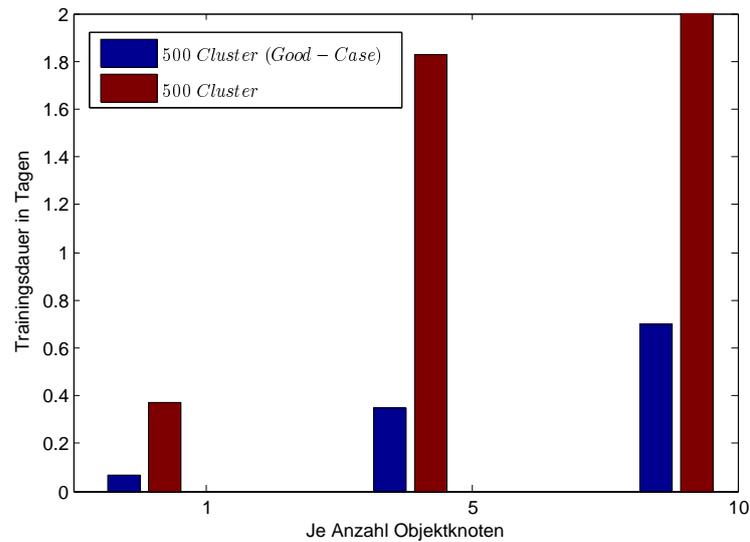


Abbildung 6.16.: Laufzeitabschätzung des Trainings bei verschiedenen Problemgrößen (2)

steigt bzw. sinkt in diesem Fall wegen der konstanten Zugriffszeit von $5,5\text{ ms}$ linear. Die Dauer des Trainings für die Anzahl gelernter Regeln ist im Bezug zur manuellen Formalisierung der Regeln durch einen Planer selbst für die Rechendauer eines Vierteljahres akzeptabel. Bei 500 Clustern werden für einen Objektknoten und 22 Aktionen 11.000 Regeln gelernt. Bei 1.000 Clustern sind es 22.000 und bei 10.000 Clustern sind es 220.000 Regeln, die durch das Verfahren über die Q-Werte gelernt werden. Im hier betrachteten Worst-Case von 10.000 Clustern auf 10 Objektknoten beträgt die Anzahl der verfügbaren Regeln nach Abschluss des Trainings bis zu 2,2 Millionen.

Die in diesem Kapitel dokumentierten Werte der Validierung des Lernverfahrens beschreiben den Fall, bei dem im Training alle Verfahren in einem für sie zugelassenen Cluster angewendet und bewertet werden. Die Auswahl erfolgt dabei bestmöglich durch die in Kapitel 5.3.1.3 vorgestellte Auswahlstrategie. Es wird sichergestellt, dass die richtigen Verfahren im Training verwendet und dadurch die wichtigsten Regeln gelernt werden.

6.2.3.3. Effizienz des Trainingsprozesses

Die im Mittel benötigte Anzahl an Lernepisoden bis zur Konvergenz der Q-Werte ist im Hinblick auf die Dauer des Gesamttrainings unkritisch. Bei der empfohlenen Anzahl von 500 Clustern bei 10.000 Trainingsdaten und bei 10 Objektknoten kann z. B. für bis zu 40 Aktionen²⁷ der Trainingsprozess im Worst-Case innerhalb von ca. 6,5

²⁷Dieses ist die Anzahl der von [Hei06] vorgestellten Änderungsplanungsverfahren.

Tagen und im Good-Case in 30 Stunden mit bis zu 200.000 gelernten Regeln abgeschlossen werden. Nach dem Lernprozess sind insgesamt 10.000 Regeln verfügbar. Verwendet werden je Objektknoten mindestens 500 Regeln, da für jeden charakteristischen Zustand je Objektknoten eine beste Regel zur Auswahl eines Planungsverfahrens nach Trainingsabschluss existiert. Bei sich gleichenden maximalen Q-Werten erhöht sich die Anzahl verfügbarer Regeln.

Der hier ermittelte Wert ist als Abschätzung der durchschnittlichen Trainingszeit zu betrachten, da gemittelte Werte verwendet wurden. Nicht jeder Trainingsprozess benötigt 524 Lernschritte bis zur Konvergenz der Q-Werte. Teilweise liegt der Wert deutlich darunter, da aufgrund der Auswahlstrategie der Planungsverfahren im Training nicht alle Q-Werte ihren maximalen Wert erreichen müssen. Der Trainingsprozess kann über Abbruchregeln zu definierten Zeitpunkten gestoppt werden. Stichproben haben gezeigt, dass die aktualisierten Q-Werte teilweise bei 50 oder weniger Lernschritten nur noch insignifikante Änderungen von weniger als 1% zum vorhergehenden Q-Wert aufwiesen und sich die Abstufung der Q-Werte im Laufe des weiteren Trainings nicht mehr regelwirksam veränderte.²⁸

Unter Berücksichtigung dieser Faktoren handelt es sich bei der hier ermittelten Zeit für das Training des Lernverfahrens mehr um ein Worst-Case-Szenario als um ein Average- oder Good-Case-Szenario. Die ermittelten Trainingszeiten sind zufriedenstellend und ermöglichen die Anwendung des Lernsystems in der Praxis. Das Lernen auf dem abstrakten Zustandsraum konnte für verschiedene Konfigurationen in akzeptabler Zeit ein gutes Lernergebnis erzielen. Die Abstraktion erhöht die Effizienz des Trainingsprozesses. Das intendierte Ziel dieser Forschungsfrage wurde erreicht.

6.2.4. Lernen im Netzwerk

Die verwendeten Szenarien in der Validierung zeigen in nachvollziehbarer Weise, wie das Lernverfahren funktioniert. Das Lernsystem kann effektiv und effizient Regeln zu Steuerung der Änderungsplanung lernen. Es wurde der Lernprozess in der Validierung zwischen zwei Fertigungsstufen analysiert. Die Ergebnisse der Validierung können auf vollständige Produktionsnetzwerke übertragen werden. Der einzige Unterschied liegt darin, dass die Dauer des Lernprozesses ansteigt.²⁹ Eine Lernepisode endet nicht nach einem Lernschritt, sondern sobald alle Objektknoten einen gültigen Plan erreicht haben. Die Q-Werte werden, je nach verwendeten Trainingsverfahren³⁰, aktualisiert.

Durch die Interaktionsmöglichkeiten der einzelnen Partner während der Änderungsplanung schwanken die Q-Werte zu Beginn des Trainings stärker. Durch das typische

²⁸Siehe Szenario 1 in Kap. 6.2.2.1

²⁹Siehe Validierung der Trainingsdauer in Kap. 6.2.3

³⁰Siehe Konzeption der Trainingsverfahren in Kap. 5.3

Verhalten beeinflusst durch die charakteristischen Zustandsmerkmale im Produktionsnetzwerk stabilisieren sie sich während des Trainingsprozesses auf einem aussagekräftigen Niveau. Dieses Niveau spiegelt die Regeln zur Auswahl der Planungsverfahren für das spezifische Netzwerk wider. Der Trainingsprozess konvergiert bzw. erreicht eine Abbruchbedingung und der Lernprozess ist beendet.

6.3. Abschließende Diskussion

Das Clustering wurde auf einem übersichtlichen Szenario ausgewertet, um transparente und nachvollziehbare Ergebnisse zu erzielen. Es wurde festgestellt, dass die konzipierten Distanzfunktionen eine zweckmäßige und anwendungsbezogene Abstraktion von Zuständen zu Clustern vornehmen. Die Ergebnisse der Validierung mündeten in Empfehlungen zur Konfiguration des Clusterings bzgl. der Anzahl zu erzeugender Cluster von 500 oder 5 – 10% der Trainingsdaten, der Anzahl zu verwendender Planverläufe im Clustering von 10.000, und der Anzahl von durchschnittlichen Iterationen von 3-5 bis zur Konvergenz des Verfahrens.

Für die Validierung des Lernverfahrens wurde ein nachvollziehbares Szenario verwendet und die Effektivität der Lernfunktion sowie die Effizienz des Lernprozesses wurden validiert. Durch die Validierung des Trainingsverfahrens konnte gezeigt werden, dass durch das Lernen auf Clustern statt auf Zuständen die Effizienz des Trainings erhöht und das Training in polynomialer Zeit durchgeführt werden kann. Durch die Verwendung der problemspezifischen Abstraktion konnte die Problemstellung der Arbeit aufgelöst werden. Die Steigerung der Effizienz durch die Abstraktion wurde durch ungenaue Bewertungen der charakteristischen Zustände im Training hinsichtlich ihrer Originalzustände erkauft. Analysiert man die jeweiligen Q-Werte, so liegen sie beim Lernen mit Clustern mit einer Differenz von kleiner als 0,01 Punkte nahe zusammen. Hinsichtlich einer Abweichung von durchschnittlich 0,03 Punkten bei der Rewardberechnung der einzelnen Clusterpläne ist die Abweichung dieser beiden Werte bisher relativ groß und kann theoretisch zu Interferenzen in der Q-Wert-Berechnung führen. Die Experimente zeigen aber, dass die Q-Werte trotzdem konvergieren. Der Konvergierungsprozess kann durch die Größe von α forciert werden.

Trotz der unvermeidbaren Abweichungen in der Rewardberechnung beim Lernen auf dem charakteristischen Zustandsraum konnte das gewünschte Lernergebnis erzielt werden. Ein Grund dafür war die problemspezifische Abstraktion des Zustandsraumes durch die Clusterfunktion, die Zustände unter Beibehaltung ihrer charakteristischen Merkmale abstrahieren kann. Weiterhin wurde dieses Ziel durch die stark problemspezifische Definition der Rewardfunktion erreicht, welche die anfallenden Strafkosten im Training effektiv ermitteln und verrechnen kann. Durch die Wahl eines hinreichend kleinen Wertes für den Lernfaktor von $\alpha = 0,2$ wurden die entstehenden Abweichungen des Rewards bei fortschreitendem Lernprozess relativiert. Würde das Clustering

verbessert, könnte wahrscheinlich die Anzahl erforderlicher Lernschritte bis zur Konvergenz der Q-Werte weitergehend verringert und so die Effizienz des Verfahrens gesteigert werden.

Insgesamt können das Clustering und die Lernfunktion im Sinne der Forschungsfragen A³¹ und B³² als effektiv und das Training bzw. das Gesamtlernsystem im Sinne der Forschungsfrage C³³ als effizient bezeichnet werden.

³¹Siehe Kap. 2.3.1, S. 41 ff.

³²Siehe Kap. 2.3.2, S. 42 ff.

³³Siehe Kap. 2.3.3, S. 44 ff.

7. Zusammenfassung und Ausblick

Der Mensch ist immer noch
der beste Computer.

(John F. Kennedy, 1958)

Die Zielsetzung der Arbeit war die Entwicklung eines maschinellen Lernverfahrens zur Automatisierung der Steuerung der Änderungsplanung von Produktionsnetzwerken der Serienfertigung. Die Steuerung erfolgt durch maschinell gelernte Regeln, die zur Auflösung von ungültigen Zuständen ein geeignetes Änderungsplanungsverfahren zur Anwendung in der Änderungsplanung empfehlen sollen. Für die Konzeption intelligenter Steuerungssysteme im Bereich der PPS kann das in dieser Arbeit vorgestellte Konzept in seiner Kombination als neuer Ansatz zur Anwendung eines maschinellen Lernsystems aus der Informatik auf ein wirtschaftswissenschaftliches Problem angesehen werden.

7.1. Zusammenfassung

Als Ausgangsbasis dient die von Heidenreich¹ entwickelte kooperative Änderungsplanung für Produktionsnetzwerke der Serienfertigung. Hierfür wurde eine zustandsbasierte Regelsprache definiert, die eine manuelle Konfiguration der Steuerung eines Änderungsplanungssystems ermöglicht. Da Produktionsnetzwerke viele Zustände annehmen können, ist die manuelle Konfiguration eines solchen Regelsystems ein aufwendiger Prozess. Bei der manuellen Regelformalisierung ist die Qualität der Regeln abhängig von der Erfahrung des entsprechenden Planers. Eine effiziente Steuerung des Produktionsnetzwerks durch das Regelsystem ist so manuell nur schwer umsetzbar. In der Umsetzung des maschinellen Lernsystems wurden drei Forschungsfragen beantwortet. Erstens wurde diskutiert, wie trotz des großen Zustandsraumes in endlicher Zeit effizient gelernt werden kann und eine entsprechende problemspezifische

¹[Hei06]

Abstraktionsfunktion spezifiziert. Zweitens wurde untersucht, wie eine effektive Lernfunktion für diese Lernaufgabe umzusetzen ist und diese entsprechend konzipiert. Drittens wurde das Training des Lernsystems so konzipiert, dass es effizient und kontrollierbar durchgeführt werden kann. Das gesamte Lernsystem musste abschließend implementiert und validiert werden.

7.1.1. Reduktion des Zustandsraumes

Die Komplexität des Zustandsraumes von Produktionsnetzwerken musste reduziert werden, um hierin mit dem zustandsbasierten Lernverfahren „Q-Learning“ in endlicher Zeit leistungsfähige Lernergebnisse erzielen zu können. Der reduzierte Zustandsraum sollte skalierbar und in betriebswirtschaftlicher Hinsicht im Sinne der Änderungsplanung interpretierbar bleiben. Als wesentlicher Faktor für den großen Zustandsraum von Produktionsnetzwerken wurden die vielfältigen Ausprägungen der Planverläufe der Objektknoten identifiziert.

Unter Verwendung des *k-means*-Algorithmus wurde eine Distanzfunktion entwickelt, durch die der Zustandsraum eines Produktionsnetzwerkes über dessen diskrete Zustände und deren Planverläufe abstrahiert werden kann. Durch die Anwendung des problemspezifischen Clusterings kann eine signifikante Reduzierung des Zustandsraumes erreicht werden. Durch ein kombiniertes und gewichtetes strukturelles sowie quantitatives Distanzmaß können Pläne verglichen und zu Clustern abstrahiert werden. Das strukturelle Distanzmaß misst den relativen Abstand von Restriktionsverletzungen zweier Planverläufe, während die quantitative Distanz die absoluten Differenzen der Restriktionsverletzungen zweier Planverläufe bewertet. Durch die Minimierung der Distanz eines Planes zum Centroid eines Clusters wird dieser Cluster zu ihm ähnlichen Plänen zugeordnet. Der Centroid des jeweiligen Clusters wird als charakteristischer Planverlauf der zugeordneten Pläne des Clusters bezeichnet.

Das Ergebnis des Clusterings ist eine benutzerdefinierte Anzahl von Clustern. Deren charakteristische Planverläufe dienen als Referenzpläne im Trainingsprozess des Lernverfahrens. Auf diesen wurden im Lernverfahren die Bewertungen mit der Rewardberechnung durchgeführt. Neben der positiv verlaufenen Analyse des Konvergenzverhaltens des Clusterings wurde in der Validierung festgestellt, dass die charakteristischen Planverläufe als Ausgangsdaten für das Lernverfahren geeignet sind.

7.1.2. Konzeption einer Lernfunktion

Als Lernverfahren wurde das Q-Learning-Verfahren ausgewählt. Die primäre Aufgabe bei der Umsetzung eines Q-Learning-Verfahrens ist die Konzeption einer Rewardfunktion, die für diesen Untersuchungsgegenstand als Strafkostenfunktion Pläne hinsicht-

lich entstehender Strafkosten bei der Durchführung einer Änderungsplanung bewertet.

Eine Änderungsplanung wird angestoßen, wenn im Produktionsnetzwerk ein ungültiger Zustand vorliegt. Ein solcher weist in einer oder mehreren Planungsperioden seines Planes Restriktionsverletzungen auf. Der Erfolg eines Änderungsplanungsverfahrens ist stark abhängig von der Art und Höhe der gegebenen Restriktionsverletzungen eines Planes. Aus diesem Grund bewertet die Rewardfunktion die Verbesserung oder Verschlechterung entstehender Strafkosten eines Planes sowohl vor als auch nach erfolgter Änderungsplanung und ermittelt den Reward, der die Verbesserung oder Verschlechterung eines Planes durch die Änderungsplanung misst. Die Rewardfunktion bewertet sowohl für Fertigungsobjektknoten als auch für Kapazitätsobjektknoten Strafkosten für Über- oder Unterschreitungen von Restriktionsgrenzen, wie den Sicherheitsbestand eines Lagers oder den maximalen Leistungsgrad einer Ressource. Sie misst allgemeine Bereitstellungsstrafkosten für Material und Betriebsmittel zur Verwendung in der Produktion.

Da die Änderungsplanung in Produktionsnetzwerken neben lokal begrenzten auch global wirksame Planungsverfahren verwendet, musste die Interaktion von Lieferant und Kunde in der Änderungsplanung analysiert und ein geeignetes Bewertungsschema aufgestellt werden. Hierzu wurden bei der Beschaffung entstehende Strafkosten über den Materialfluss zwischen Lieferant und Kunde bewertet und abhängig vom Erfolg des Beschaffungsprozesses mit lokal entstehenden Strafkosten in Relation gesetzt.

Die lokal und global entstehenden Strafkosten werden zu Gesamtstrafkosten summiert. Bei dem Konzept wurde insbesondere der unvollständigen Informationslage und den dadurch entstehenden Problemen der Bewertung globaler Änderungsplanungsverfahren Rechnung getragen. Die aus lokalen und globalen Strafkosten kombinierte Rewardfunktion kann das Ergebnis lokaler als auch globaler Änderungsplanungsverfahren problemspezifisch bewerten und effektiv im Lernprozess verwenden. Durch Parametrisierung besteht die Möglichkeit, die Rewardfunktion hinsichtlich verschiedener Lernziele anzupassen.

Die Rewardfunktion wurde anschließend in den Trainingsprozess integriert. Die Berechnung des Rewards, wie auch die Aktualisierung des Q-Wertes, wurde über die charakteristischen Planverläufe der Cluster durchgeführt. Dabei wurde vor und nach erfolgter Planung das zugehörige Cluster zum aktuell betrachteten Plan verwendet. Die Konvergenz des Lernverfahrens wurde analytisch hergeleitet.

Die Regeln können durch eine Sortierung der Q-Werte der Cluster erzeugt werden. Je höher der Q-Wert einer Aktion eines charakteristischen Zustandes, desto besser hat sich diese Aktion bewährt. Eine gute Regel zur Auflösung eines ungültigen Zustandes besteht aus der Beschreibung des Musters eines zugeordneten, charakteristischen Planverlaufes und der Aktion mit dem höchsten und somit besten Q-Wert. Um die gelernten Regeln zu Steuerung verwenden zu können, müssen sie nicht explizit erzeugt

werden. Durch die Zuordnung von Plänen zu Clustern können jederzeit der beste Q-Wert des Clusters und eine geeignete Aktion zur Auflösung eines ungültigen Zustandes ermittelt werden. Beide Varianten wurden in der Arbeit diskutiert und algorithmisch beschrieben.

Die Validierung zeigte, dass die gelernten Q-Werte während des Trainings in endlicher Zeit konvergieren. Die Abstufungen der Q-Werte erfüllten die Anforderungen an das betrachtete Szenario. Die Rewardfunktion ermöglichte eine problemspezifische, effektive Bewertung von Zuständen und angewendeten Verfahren in der Änderungsplanung.

7.1.3. Ausgangsdaten und Trainingskonzept

Um die Q-Werte lernen zu können, musste ein adäquates Trainingskonzept für das Clustering und das Lernverfahren entwickelt werden. Untersucht wurde, wie in der verteilten Umgebung eines Produktionsnetzwerkes mit autonom agierenden Partnern effizient über einen strukturierten Lernprozess gelernt werden kann. Die Anforderungen an die Ausgangsdaten zur Unterstützung der Effektivität des Lernprozesses wurden analysiert. Die Datenstruktur und Bereitstellungsmöglichkeiten der Ausgangsdaten wurden diskutiert und ein Lösungsansatz vorgeschlagen. Es wurde ein Konzept zur Gestaltung des Trainingsprozesses vorgestellt, welches auf Realdaten aus ERP-Systemen operieren kann.

Es wurden Parameter etabliert, durch die sowohl die Dauer einer Lernepisode als auch die Dauer des gesamten Trainings konfiguriert werden kann. Durch die Definition einer Grammatik können die Abbruchkriterien anwendungsfallsspezifisch erweitert werden. Die Validierung zeigte, dass der Trainingsprozess für verschiedene realistische Netzwerkgrößen effizient durchführbar ist.

7.1.4. Umsetzung

Das Clustering, das Lernverfahren und alle benötigten Änderungsplanungsverfahren wurden mit der Multiagentenplattform JADE in der Programmiersprache JAVA implementiert. Dabei wurden die in JADE umgesetzten FIPA²-Protokolle an die Anforderungen der kooperativen Änderungsplanung angepasst. Die gesamte Implementation ist modular aufgebaut und durch XML-Dateien weitreichend konfigurierbar. Die Validierung der Konzepte erfolgte auf dem implementierten System.³

²[Fou97]

³Siehe Anhang C

7.2. Grenzen der Arbeit

Diese Arbeit beschränkt sich auf Produktionsnetzwerke der Serienfertigung. Für diesen Untersuchungsgegenstand kann die Änderungsplanung nach dem in dieser Arbeit beschriebenen Prinzip durchgeführt, und es können die Konzepte des Lernsystems angewendet werden. Da für die gegebene Problemstellung keine direkt vergleichbaren Ansätze existieren, versteht sich diese Arbeit als erster Beitrag in der Wirtschaftsinformatik, die Forschung im Bereich intelligenter Steuerungssysteme von Produktionsnetzwerken aufbauend auf diesem Lernkonzept zu vertiefen.

In der aktuellen Forschung, z. B. im Rahmen des Operations Research, oder im industriellen Umfeld treten zum Teil komplexere Planungsprobleme anderer Produktionssysteme auf, die z. B. auf einem detaillierten Planungsmodell aufsetzen können. Aus dort aufkommenden neuen Anforderungen leitet sich zukünftiger Forschungsbedarf ab. Da eine effiziente Änderungsplanung in der Industrie⁴ einen hohen Stellenwert besitzt, kann die weitergehende Forschung in diesem Bereich als sinnvoll betrachtet werden.

7.3. Ausblick

Das vorgestellte Clusteringverfahren lässt sich an einigen Stellen erweitern. Anstatt die abstrahierten Zustände vollständig vor der Anwendung des Lernsystems festzulegen, ist eine Anpassung der Abstraktionsfunktion parallel zum Fortschritt des Lernsystems denkbar. Dadurch ließen sich die Erkenntnisse, die erst während des Lernens gewonnen werden, nachträglich in die Aufteilung des Zustandsraumes einarbeiten. Ein Beispiel ist die Möglichkeit, neue abstrahierte Zustände zu bilden, wenn für den aktuell bearbeiteten kein abstrahierter Zustand gefunden werden kann, der diesem hinreichend ähnlich ist. Es wäre denkbar, die Abstraktion vollständig während der Lernphase durchzuführen. Schwellenwerte legen fest, wann ein Q-Update aufgrund starker Ähnlichkeit unter einem Q-Wert eines bestehenden Clusters verrechnet wird, und wann ein neuer Cluster erzeugt werden muss.

Die Clusterfunktion kann so erweitert werden, dass Restriktionsverletzungen im Centroiden noch besser berücksichtigt werden. Es sollte versucht werden, die Zuordnungsdauer zwischen diskretem Zustand und Cluster zu verringern und unabhängig von der Clustergröße konstant zu halten.

Eine Erweiterung des Lernverfahrens auf eine kooperative Änderungsplanung mit komplexeren Modellen, z. B. mit Berücksichtigung von Rüstzeiten, ist sinnvoll. Das Lernverfahren könnte eingesetzt werden, um die Steuerung der Änderungsplanung zeitge-

⁴Siehe Kap. 1, S. 2

mäßer ERP-Systeme wie SAP APO, d. h. die Sequenz auszuführender Planungsalgorithmen, durch die Regeln zu konfigurieren. Es könnten Regeln erzeugt werden, die sowohl heuristische Änderungsplanungsverfahren als auch Optimierungsalgorithmen berücksichtigen und kombinieren können. In SAP APO kann die bisher vom Menschen festgelegte Ablaufreihenfolge von Planungs- und Optimierungsverfahren durch ein solches Lernverfahren inhärent gelernt werden. Die Systemkonfiguration eines Planungssystems würde fortan maschinell durchgeführt und der manuelle Aufwand reduziert. Zur Umsetzung dieser Vision müsste insbesondere die Strafkostenfunktion zur Rewardberechnung unter Beibehaltung des hier vorgestellten Prinzips auf die erweiterte Problemklasse detailliert und das Planungsmodell verfeinert werden.

Bisher erfolgte die Erzeugung von Regeln nach Abschluss des Lernprozesses über eine gegebene Konfiguration des Produktionsnetzwerkes. Änderungen der Netzwerkkonfiguration führen u. U. zur Ungültigkeit einiger Regeln. Eine Erweiterungsmöglichkeit besteht, in dem Teilmengen oder alle Regeln zur Anwendungszeit des Regelsystems quasi „online“ angepasst bzw. gelernt werden können. Ebenso könnten die Regeln während der Laufzeit eines Änderungsplanungssystems gelernt werden, indem durchgeführte Änderungsplanungen automatisch durch das Lernverfahren im Sinne eines *Lesson-Learned*-Prinzips bewertet werden.

Literaturverzeichnis

- [AC/10] AC/DC - AUTOMOTIVE CHASSIS FOR 5DAYCARS: *EU Projekt AC/DC (Fördernummer 031520)*. <http://www.acdc-project.org>, Laufzeit 2006-2010
- [AKM91] AHA, D. ; KIBLER, D. ; M.ALBERT: Instance-Based Learning Algorithms. In: *Machine Learning* 6 (1991), S. 37–66
- [Bai95] BAIRD, L. C.: Residual Algorithms: Reinforcement Learning with Function Approximation. In: *International Conference on Machine Learning*, 1995, 30-37
- [Bec91] BECKER, B.-D.: *Simulationssystem für Fertigungsprozesse mit Stückgutcharakter - Ein gegenstandsorientiertes System mit parametrisierter Netzwerkmodellierung*. Berlin : Springer Verlag, 1991
- [Ber02] BERKHIN, P.: Survey Of Clustering Data Mining Techniques / Accrue Software. Version: 2002. http://www.accrue.com/products/rp_cluster_review.pdf. San Jose, CA, 2002. – Forschungsbericht
- [BHHB06] BAUMGÄRTEL, H. ; HELLINGRATH, B. ; HOLWEG, M. ; BISCHOFF, J.: Automotive SCM in einem vollständigen Build-to-Order-System. In: *Supply Chain Management* 1 (2006), S. 7–15
- [BKI03] BEIERLE, C. ; KERN-ISBERNER, G.: *Methoden Wissensbasierter Systeme*. 2. Vieweg, 2003
- [BKP05] BICHLER, K. (Hrsg.) ; KROHN, R. (Hrsg.) ; PHILIPPI, P. (Hrsg.): *Gabler Kompakt-Lexikon Logistik*. Wiesbaden : Gabler Verlag, 2005
- [BM95] BOYAN, J. ; MOORE, A.: Generalization in Reinforcement Learning: Safely Approximating the Value Function. In: TESAURO, G. (Hrsg.) ; TOURETZKY, D. S. (Hrsg.) ; LEEN, T. K. (Hrsg.): *Advances in Neural Information Processing Systems 7*. Cambridge, MA. : The MIT Press, 1995
- [Bol03] BOLL, H.: *Evolutionäre Verfahren zur Optimierung von Produktionsplänen mittels impliziter Kooperation*. Aachen, RWTH Aachen, Dissertation, Januar 2003

- [BPR01] BELLIFEMINE, F. ; POGGI, A. ; RIMASSA, G.: A FIPA2000 compliant agent development environment. In: *Proceedings of the 5. International Conference on Autonomous agents*. New York, NY, USA : ACM Press, 2001, S. 216–217
- [Bra99] BRAUSE, R.: *Neuronale Netze. Eine Einführung in die Neuroinformatik*. Teubner Verlag, 1999
- [Bus04] BUSCH, A.: *Kollaborative Änderungsplanung in Unternehmensnetzwerken der Serienfertigung – eine verhandlungsbasierte Konzeption zur interorganisationalen Koordination von Störungen*. Paderborn, Heinz Nixdorf Institut, Universität Paderborn, Dissertation, 2004
- [CA96] CRITES, R. H. ; A.G., Barto: Improving Elevator Performance Using Reinforcement Learning. In: TOURETZKY, D. S. (Hrsg.) ; MOSER, M. C. (Hrsg.) ; HASSELMO, M. E. (Hrsg.): *Advances in Neural Information Processing Systems 8*. MIT Press : MIT Press, 1996, S. 1017–1023
- [CH67] COVER, T. M. ; HART, P. E.: Nearest Neighbour Pattern Classification. In: *IEEE Transactions on Information Theory* 13 (1967), S. 21–27
- [CK91] CHAPMAN, D. ; KAEHLING, L. P.: Input generalization in delayed reinforcement learning: An algorithm and performance comparison. In: *Proceedings of the 1991 International Joint Conference on Artificial Intelligence*, 1991, S. 726–731
- [Cor00] CORSTEN, H.: *Produktionswirtschaft: Einführung in das industrielle Produktionsmanagement*. München : Oldenbourg Wissenschaftsverlag, 2000
- [CS03] CAO, H. ; SMITH, S. F.: A Reinforcement Learning Approach to Production Planning in the Fabrication/Fulfillment Manufacturing Process. In: CHICK, S. (Hrsg.) ; SANCHEZ, P. J. (Hrsg.) ; FERRIN, D. (Hrsg.) ; MORRICE, D. J. (Hrsg.): *Proceedings of the Winter Simulation Conference*, 2003
- [CUGI05] CAPGIMINI ; UNIVERSITY OF TENNESSEE ; GEORGIAN UNIVERSITY ; INTEL: *Collaboration: Enabling Synchronized Supply Chains - Year 2005 Report on Trends and Issues in Logistics and Transportation*. Capgimini, 2005. – Study
- [CYT05] CHA, S.-H. ; YOON, S. ; TAPPERT, C. C.: Enhancing Binary Feature Vector Similarity Measures / Ivan G. Seidenberg School of Computer Science and Information Systems. 2005 (210). – Forschungsbericht
- [Dan99] DANGELMAIER, Wilhelm: *Fertigungsplanung: Planung von Aufbau und Ablauf der Fertigung*. Berlin Heidelberg : Springer, 1999

- [DB02] DANGELMAIER, Wilhelm (Hrsg.) ; BUSCH, Axel (Hrsg.): *Integriertes Supply Chain Management*. Gabler, 2002
- [DDKT07] DANGELMAIER, W. ; DÖRING, A. ; KREBS, W. ; TIMM, T.: Customize-to-Order: Optimized Planning and Control of Global Automotive Supply Networks. In: PILLER, F. T. (Hrsg.) ; MITCHELL, W. J. (Hrsg.) ; T., M. (Hrsg.) ; MCCALAHAN, B. L. (Hrsg.) ; CHIN, R. (Hrsg.) ; MIT, Cambridge M.A. (Veranst.): *Proceedings of the 2007 World Conference on Mass Customization & Personalization 2007* MIT, Cambridge M.A., MIT Press on CD, Oktober 2007
- [DDLT07] DÖRING, A. ; DANGELMAIER, W. ; LAROQUE, Ch. ; TIMM, T.: Simulation-aided process coverage for delivery schedules under short delivery schedules using real-time event based feedback loops. In: *Proceedings of the 6th EUROSIM Congress on Modelling and Simulation* Bd. Vol. 1. Ljubljana, Slovenia, 9-13 Spetemper 2007 2007, S. 122
- [DKT07] DÖRING, A. ; KREBS, W. ; TIMM, T.: Customize-to-Order. In: *PPS Management* 3 (2007), S. 26–29
- [DN07] DIEDRICHSEN, K. ; NICKERL, R. J.: Interview: Intelligenter als das reine Event. In: *Logistik Heute* Dezember (2007), S. 16–18
- [DPR04] DANGELMAIER, W. ; PAPE, U. ; RÜTHER, M.: *Agentensysteme für das Supply Chain Management*. Wiesbaden : Deutscher Universitätsverlag, 2004
- [Dud99] DUDENHAUSEN, H.-M.: *Auftragskoordination in Produktionsnetzwerken der Halbleiterindustrie*. Heimsheim : Jost-Jetter Verlag, 1999
- [DW93] DANGELMAIER, W. ; WIEDEMANN, H.: *Modell der Fertigungssteuerung*. Berlin u. a. : Beuth, 1993
- [DW97a] DANGELMAIER, W. ; WARNECKE, H.-J.: *Fertigungslenkung: Planung und Steuerung des Ablaufs der diskreten Fertigung*. Berlin u. a. : Springer, 1997
- [DW97b] DANGELMAIER, W. ; WIDEMANN, H.: *Modellbasiertes Planen und Steuern der Fertigung*. Beuth, 1997
- [Eis89] EISENFÜHR, F.: *Grundlagen der Produktionswirtschaft - Industriebetriebslehre I.*. Aachen : Verlag Augustinus-Buchhandlung, 1989
- [Erl07] ERLACH, K.: *Wertstromdesign. Der Weg zur schlanken Fabrik*. Springer Berlin, 2007
- [EST⁺95] ERIKSON, H. ; SHAHAR, Y. ; TU, S. W. ; PUERTA, A. R. ; MUSEN, M. A.: Task Modeling with Reusable Problem-Solving Methods. In: *Artificial Intelligence* 79 (1995), S. 293–326

- [Fer01] FERBER, J.: *Multiagentensysteme*. Addison-Wesley, 2001
- [FG96] FRANKLIN, S. ; GRAESSER, A.: Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents. In: *Intelligent Agents III. Agent Theories, Architectures and Languages (ATAL '96)* Bd. 1193. Berlin : Springer, 1996
- [FL99] FINNIN, T. ; LABROU, Y.: *Agent Communication Languages*. 1999. – Tutorial slides
- [Flo98] FLORIAN, M.: Multiagentensysteme für die kooperative Transportdisposition - Das soziotechnische Rationalisierungspotential der Verteilten Künstlichen Intelligenz (VKI) / Technische Universität Hamburg-Harburg. 1998 (RR 1). – Forschungsbericht
- [FM04] FRAUNHOFER GESELLSCHAFT ; MERCER MANAGEMENT CONSULTANTS: *Future Automotive Industry (FAST) 2015*. Studie. Mercer Management Consultants, 2004
- [Fou97] FOUNDATION FOR PHYSICAL INTELLIGENT AGENTS (FIPA): Agent Communication Language (ACL) / FIPA. Geneva, Switzerland, 1997. – Forschungsbericht
- [Fra04] FRANKE, H.: *Ein Methode zur unternehmensübergreifenden Transportdisposition durch synchron und asynchron kommunizierende Agenten*, Universität Paderborn, Dissertation, 2004
- [GK98] GOLLWITZER, M. ; KARL, R.: *Logistik-Controlling*. München : Wirtschaftsverlag Langen Müller/Herbig, 1998
- [GO74] GROSSE-OETRINGHAUS, W. F.: *Fertigungstopologie unter dem Gesichtspunkt der Fertigungsablaufplanung*. Berlin : Duncker & Humboldt Verlag, 1974
- [GRS03] GÖRZ, G. (Hrsg.) ; ROLLINGER, C.-R. (Hrsg.) ; SCHNEEBERGER, J. (Hrsg.): *Handbuch der Künstlichen Intelligenz*. 4. korrigierte Auflage. Oldenbourg Wissenschaftsverlag, 2003
- [Gud04] GUDEHUS, T.: *Logistik*. Springer Berlin Heidelberg, 2004
- [Hac84] HACKSTEIN, R.: *Produktionsplanung und -steuerung (PPS) - Ein Handbuch für die Betriebspraxis*. VDI-Verlag GmbH, 1984
- [Hei06] HEIDENREICH, J.: *Adaptierbare Änderungsplanung der Mengen und Kapazitäten in Produktionsnetzwerken der Serienfertigung*, Universität Paderborn, Heinz Nixdorf Institut, Wirtschaftsinformatik, insbesondere CIM, Dissertation, 2006

- [HM88] HALDER, A. ; MÜLLER, M.: *Philosophisches Wörterbuch*. 3. Freiburg et. al. : Herder Spektrum, 1988
- [HM07] HELLINGRATH, B. ; MANDEL, J.: Das 5-Tage-Auto - Serienfertigung gehört schon bald zum alten Eisen. In: HÄNSCH, T. W. (Hrsg.): *100 Produkte der Zukunft: Wegweisende Ideen, die unser Leben verändern werden*. Düsseldorf : Econ Verlag, 2007, S. 242–243
- [Hol00] HOLTHÖFER, N.: *Regeln in einer Mengenplanung unter Ausbringungsgrenzen*, Universität Paderborn, Dissertation, 2000
- [HR90] HAYES-ROTH, B.: An architecture for adaptive intelligent systems. In: *Proc. of the Workshop on Innovative Approaches to Planning*. San Diego, CA, 1990, S. 422–432
- [HRB⁺99] HUANG, Y. ; RAMAKRISHMA, R. ; BEGUE, C. ; BAKKALBASI, O. ; CHAN, L. M. A. ; FEDERGRUEN, A. ; KRASINSKI, R. J. ; BOEY, P.: Descision Support System for the Management of an agile Supply Chain. In: *United States Patent Patent Number 5,953,707* (1999)
- [HSA99] HEINSOHN, J. ; SOCHER-AMBROSIUS, R.: *Wissensverarbeitung*. Spektrum Akademischer Verlag, 1999
- [HW79] HARTIGAN, J. A. ; WONG, M. A.: A K-Means Clustering Algorithm. In: *Applied Statistics* 28 (1979), 100–108. <http://www.jstor.org/view/00359254/di993342/99p04867/0>
- [HW98] HU, J. ; WELLMANN, M. P.: Online Learning about Other Agents in a Dynamic Multiagent System. In: *Proceedings of the Second International Conference on Autonomous Agents (Agents-98)*. Minneapolis, 1998
- [IA00] INTELLIGENT AGENTS, Foundation for: FIPA Policies and Domains Specification / Foundation for Intelligent Agents (FIPA). Geneva, Switzerland, 2000. – Forschungsbericht
- [Jec04] JECKLE, M.: UML 2.0 Schwerpunkt. In: *Objektspektrum* 3 (2004), März, S. 12–34
- [JMF99] JAIN, A. K. ; MURTY, M. N. ; FLYNN, P. J.: Data clustering: a review. In: *ACM Computing Surveys* 31 (1999), Nr. 3, 264–323. <http://dx.doi.org/10.1145/331499.331504>. – DOI 10.1145/331499.331504
- [KB75] KLAUS, G. ; BUHR, M.: *Philosophisches Wörterbuch - Band 2*. 11. Leipzig : Verlag Enzyklopädie, 1975
- [KB87] KLAUS, G. ; BUHR, M.: *Philosophisches Wörterbuch - Band 1*. 12. Westberlin : deb - Verlag das europäische Buch, 1987

- [KH02] KUHN, A. ; HELLINGRATH, H.: *Supply Chain Management - Optimierte Zusammenarbeit in der Wertschöpfungskette*. Springer Berlin Heidelberg, 2002
- [KK04] KLAUS, P. (Hrsg.) ; KRIEGER, W. (Hrsg.): *Gabler Lexikon Logistik*. 3. Auflage. Wiesbaden : Gabler Verlag, 2004
- [KL90] KARBACH, W. ; LINSTER, M.: *Wissensakquisition für Expertensysteme*. München-Wien : Hanser-Verlag, 1990
- [Kla99] KLAHOLD, R. F.: *Dimensionierung komplexer Produktionsnetzwerke*, GH-Universität Paderborn, Dissertation, 1999
- [Kle02] KLEIN, R.: Genetic Algorithm. In: STADTLER, H. (Hrsg.) ; KILGER, C. (Hrsg.): *Supply Chain Management and Advanced Planning - Concepts, Models, Software and Case Studies*. Berlin : Springer Verlag, 2002, S. 403–410
- [KLM96] KEABELING, L. P. ; LITTMANN, M. L. ; MOORE, A. W.: Reinforcement Learning: A Survey. In: *Journal of Artificial Intelligence* 4 (1996), S. 237–285
- [Kuh99] KUHN, A.: *Referenzmodelle für Produktionsprozesse zur Untersuchung und Gestaltung von PPS-Aufgaben*. Paderborn, Universität Paderborn, Dissertation, 1999
- [Kur05] KURBEL, K.: *Produktionsplanung und -steuerung im Enterprise Resource Planning and Supply Chain Management*. 6. Auflage. Oldenburg, 2005
- [Lar05] LAROSE, Daniel T.: *Discovering Knowledge in Data*. Wiley Interscience, 2005
- [Lar07] LAROQUE, C.: *Ein mehrbenutzerfähiges Werkzeug zur Modellierung und richtungsoffenen Simulation von wahlweise objekt- und funktionsorientiert gegliederten Fertigungssystemen*, Universität Paderborn, Dissertation, 2007
- [LHNH00] LAWRENZ, O. ; HILDEBRAND, K. ; NENNINGER, M. ; HILLEK, T.: *Supply Chain Management*. Vieweg Business Computing, 2000
- [Loc06] LOCKEMANN, P. C.: Agents. In: KIRN, St. (Hrsg.) ; HERZOG, O. (Hrsg.) ; LOCKEMANN, P. (Hrsg.) ; SPANIOL, O. (Hrsg.): *Multiagent Engineering: Theory and Applications in Enterprises (International Handbooks on Information Systems)*. Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2006, Kapitel 1, S. 17–34

- [LR02] LAUER, M. ; RIEDMILLER, M.: Generalisation in Reinforcement Learning and the Use of Observation-Based Learning. In: KOKAI, G. (Hrsg.) ; ZEIDLER, J. (Hrsg.): *Proceedings of the FGML Workshop, 2002*, S. 100–107
- [LW00] LUTZ, S. ; WIENDAHL, H.-P.: Monitoring und Controlling in Produktionsnetzwerken. In: *wt-Werkstatttechnik* (2000), Nr. 90, S. 193–195
- [Man97] MANNMEUSEL, T.: *Dezentrale Produktionslenkung unter Nutzung verhandlungsbasierter Koordinationsformen*. Wiesbaden : Deutscher Universitätsverlag, 1997
- [MC92] MAHADEVAN, S. ; CONNELL, J.: Automatic programming of behavior-based robots using reinforcement learning. In: *Artificial Intelligence* 55 (1992), Nr. 2-3, S. 311–365. [http://dx.doi.org/10.1016/0004-3702\(92\)90058-6](http://dx.doi.org/10.1016/0004-3702(92)90058-6). – DOI 10.1016/0004-3702(92)90058-6
- [McC95] MCCALLUM, A. K.: *Reinforcement Learning with Selective Perception and Hidden State*. Rochester, NY, USA, University of Rochester, Department of Computer Science, Dissertation, 1995
- [Mig97] MIGDALAS, A.: *Parallel computing in optimization*. Kluwer Academic, 1997
- [Mit97] MITCHELL, Tom M.: *Machine Learning*. McGraw-Hill Book Co, 1997
- [MMDG97] MAHADEVAN, S. ; MARCHALLECK, N. ; DAS, T. K. ; GOSAVI, A.: Self-improving Factory Simulation using Continuous-time Average-Reward Reinforcement Learning / Department of Computer Science and Engineering University Of Florida. Tampa, Florida, 1997 (IRI-9501852). – Resarch Report granted by NSF CAREER Award
- [Mor03] MORALES, Eduardo F.: Scaling up reinforcement learning with a relational representation. In: *Workshop on Adaptability in Multi-Agent Systems*, 2003
- [Nis97] NISSEN, V.: *Einführung in evolutionäre Algorithmen*. Vieweg, 1997
- [Pap06] PAPE, U.: *Agentenbasierte Umsetzung eines SCM-Konzeptes zum Liefermanagement in Liefernetzwerken der Serienfertigung*. Paderborn, Heinz Nixdorf Institut, Universität Paderborn, Dissertation, Januar 2006
- [Pat01] PATIG, S.: *Flexible Produktionsfeinplanung mit Hilfe von Planungsschritten - Ein Planungsansatz zum Umgang mit Störungen bei der Produktion*. Magdeburg, Fakultät für Informatik der Otto-von-Guericke-Universität Magdeburg, Dissertation, 2001
- [PGPB96] PUPPE, F. ; GAPPA, U. ; POECK, K. ; BAMBERGER, S.: *Wissensbasierte Diagnose- und Informationssysteme*. Berlin : Springer, 1996

- [PSD01] PRECUP, D. ; SUTTON, R. S. ; DASGUPTA, S.: Off-Policy Temporal-Difference Learning with Function Approximation. In: *Proceedings of the 18th International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, CA, 2001, 417–424
- [PSS03] PUPPE, F. ; STOYAN, H. ; STUDER, R.: Knowledge Engineering. In: GÖRZ, G. (Hrsg.) ; ROLLINGER, C.-R. (Hrsg.) ; SCHNEEBERGER, J. (Hrsg.): *Handbuch der Künstlichen Intelligenz*. Oldenbourg, 2003, Kapitel 15, S. 600–641
- [Qui90] QUINLAIN, J. R.: Learning logical definitions from relations. In: *Machine Learning* 5 (1990), S. 239–266
- [REF84] REFA (Hrsg.): *Methodenlehre des Arbeitsstudiums Teil 1. Grundlagen*. 7. Auflage. München : Hanser Verlag, 1984
- [REF91] REFA (Hrsg.): *Methodenlehre der Betriebsorganisation*. 7. Auflage. München : Hanser Verlag, 1991
- [RHW86] RUMELHART, D. E. ; HINTON, G. E. ; WILLIAMS, R. J.: Learning internal representations by error propagation. In: RUMELHART, D. E. (Hrsg.) ; MCCLELLAND, J. (Hrsg.): *Parallel distributed processing* Bd. 1. Cambridge, MA, USA : MIT Press, 1986, S. 318–362
- [RN03] RUSSELL, S. ; NORVIG, P.: *Artificial Intelligence - A Modern Approach*. 2. Auflage. Prentice Hall, 2003
- [RR99] RIEDMILLER, S. C. ; RIEDMILLER, M. A.: A neural reinforcement learning approach to learn local dispatching policies in production scheduling. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'99)*. Stockholm, Schweden, 1999, S. 764–771
- [Rüt05] RÜTHER, M.: *Ein Beitrag zur klassifizierenden Modularisierung von Verfahren für die Produktionsplanung*, Universität Paderborn, Dissertation, 2005
- [SAA⁺99] SCHREIBER, A. T. ; AKKERMANS, H. ; ANJEWERDEN, A. ; HOOG, R. de ; SHADBOLT, N. ; VELDE, W. van d. ; WIELINGA, B.: *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press, 1999
- [Sab93] SABES, P. N.: Approximating Q-Values with Basis Function Representations. In: *Proceedings of the 1993 Connectionist Models Summer School*. Hillsdale, USA : Erlbaum, 1993

- [SB97] SINGH, S. P. ; BERTSEKAS, D.: Reinforcement Learning for dynamic channel allocation in cellular telephone systems. In: *Advances in Neural Information Processing Systems: Proceedings of the 1996 Conference*. Cambridge, MA, , MIT Press, 1997
- [SB98] SUTTON, R. S. ; BARTO, A. G.: *Reinforcement Learning: An Introduction*. Bradford Book - MIT Press, 1998
- [Sch96] SCHNEIDER, U.: *Ein formales Modell und eine Klassifikation für die Fertigungssteuerung*. Paderborn, Universität-GH Paderborn, Dissertation, 1996
- [Sch97] SCHNEIDER, H.-J. (Hrsg.): *Lexikon der Informatik und Datenverarbeitung*. München : Oldenbourg Verlag, 1997
- [Sch99a] SCHEKELMANN, A.: *Materialflusssteuerung auf der Basis des Wissens mehrerer Experten*, GH-Universität Paderborn, Dissertation, 1999
- [Sch99b] In: SCHOTTEN, M.: *Grundlagen der Produktionsplanung und -steuerung*. Springer Verlag Berlin, 1999, S. 9–258
- [Sch00] SCHÖNING, U.: *Logik für Informatiker*. 5. Heidelberg et. al. : Spektrum Akademischer Verlag, 2000
- [Sch02] SCHALLNER, H.: *Eine generative Koordinationsplattform für dynamische Produktionsnetzwerke*. Shaker Verlag, 2002
- [Sch05] SCHULTE, Ch.: *Logistik - Wege zur Optimierung der Supply Chain*. 4. Verlag Vahlen, 2005
- [SG00] SEEMANN, J. ; GUDENBERG, J. W.: *Softwareentwurf in UML*. Springer Berlin Heidelberg, 2000
- [SJJ95] SINGH, S. P. ; JAAKKOLA, T. ; JORDAN, M. I.: Reinforcement Learning with Soft State Aggregation. In: TESAURO, G. (Hrsg.) ; TOURETZKY, D. (Hrsg.) ; LEEN, T. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 7, The MIT Press, 1995, 361–368
- [SM83] SALTON, G. ; MCGILL, M. J.: *Introduction to Modern Information Retrieval*. New York : McGraw-Hill, 1983
- [SM06] SUHL, L. ; MELLOULI, T.: *Optimierungssysteme. Modelle, Verfahren, Software, Anwendungen*. Berlin Heidelberg New York : Springer, 2006 (1. Auflage)
- [SNSS04] STOCKHEIM, T. ; NIMIS, J. ; SCHOLZ, T. ; STEHLI, M.: How to build a Multi-Multi-Agent System – The Agent.Enterprise Approach. In: *6th International Conference on Enterprise Information Systems (ICEIS '04)*, 2004

- [SRHGB05] SCHOLZ-REITER, B. ; HAMANN, T. ; GORNAU, N. ; BOGEN, J.: Fallbasierte neuronale Produktionsregelung: Nutzung des Case-Based Reasoning zur Produktionsregelung mit neuronalen Netzen. In: *wt Werkstatt-technik online* (2005), Nr. 4, S. 293–298
- [SRRF06] SCHOLZ-REITER, B. ; REKERSBRINK, H. ; FREITAG, M.: Kooperierende Routingprotokolle zur Selbststeuerung von Transport. In: *Industrie Management* 22 (2006), Nr. 3, S. 7–10
- [SS00] SUHL, U. ; SUHL, L.: Mathematical Optimization System. In: *OR News* 8 (2000), S. 11–16. – ngl
- [SSK03] STOCKHEIM, T. ; SCHWIND, M. ; KOENIG, W.: A Reinforcement Learning Approach for Supply Chain Management. In: *1st European Workshop on Multi-Agent Systems*. St Catherine’s College, Oxford, UK, 2003
- [Ste00] STEGHERR, T.: *Reinforcement-Learning zur dispositiven Auftragssteuerung in der Variantenreihenproduktion*. München : Herbert Utz Verlag, 2000
- [Sti98] STIEFBOLD, O.: *Konzeption eines reaktionsschnellen Planungssystems für Logistikketten auf Basis von Softwareagenten*, Universität Karlsruhe, Diss., Juli 1998
- [Sut96] SUTTON, R. S.: Generalization in reinforcement learning: Successful Examples using sparse coarse coding. In: *Advances in Neural Information Processing Systems* 8 (1996), S. 1038–1044
- [SW01] SCHNEIDER, U. ; WERNER, D.: *Taschenbuch der Informatik*. 4. Fachbuchverlag Leipzig, 2001
- [TB04] TONG, Hui ; BROWN, Timothy: Reinforcement Learning for Call Admission Control and Routing under Quality of Service Constraints in Multimedia Networks. In: *Machine Learning* 49 (2004), November, Nr. 2-3, S. 111–139. <http://dx.doi.org/10.1023/A:1017924227920>. – DOI 10.1023/A:1017924227920
- [Tem06] TEMPELMEIER, Horst: *Material-Logistik*. Bd. 6. Springer, 2006
- [Tes95] TESAURO, G.: Temporal Difference Learning and TD-Gammon. In: *Communications of the ACM* 38 (1995), Nr. 3, S. 58–68
- [Til] TILAB: *JADE*. <http://jade.tilab.com/>,
- [TR96] TSITSIKALIS, J. N. ; ROY, B. van: Feature-based methods for large scale dynamic programming. In: *Machine Learning* 22 (1996), March, Nr. 1-3, S. 59–94

- [TS93] THRUN, S. ; SCHWARTZ, A.: Issues in Using Function Approximation for Reinforcement Learning. In: MOZER, M. (Hrsg.) ; SMOLENSKY, P. (Hrsg.) ; TOURETZKY, D. (Hrsg.) ; ELMAN, J. (Hrsg.) ; WEIGEND, A. (Hrsg.): *Proceedings of the 1993 Connectionist Models Summer School*. Hillsdale, NJ : Lawrence Erlbaum, 1993
- [UV98] UTHER, W. T. B. ; VELOSO, M. M.: Tree based discretization for continuous state space reinforcement learning. In: *IAAI '98: Proceedings of the tenth conference on Innovative Applications of Artificial Intelligence*. Menlo Park, CA, USA : American Association for Artificial Intelligence, 1998, S. 769–774
- [VDI93] VDI: *VDI Richtlinie 3633: Simulation von Logistik-, Materialfluss- und Produktionssystemen - Grundlagen*. VDI Verlag Düsseldorf, 1993
- [VHL03] VIERA, G. E. ; HERRMANN, J. W. ; LIN, E.: Rescheduling manufacturing systems: a framework of strategies, policies and methods / University of Maryland. 2003 (Funded by CAPES Brazil, grant number 3135/95-3). – Research Report
- [WA99] WALKER, W. T. ; ALBER, K. L.: Understanding Supply Chain Management. In: *APICS – The Performance Advantage*, 9 (1999), Nr. 1, S. 38–43
- [Wal03] WALTER, T.: *Grundlagen der Informatik: Informationsverarbeitung mit der Maschine – vom Algorithmus zum Programm*. München : Hanser Verlag, 2003
- [Wat89] WATKINS, C. J.: *Learning from Delayed Rewards*, University of Cambridge, Diss., 1989
- [WD92] WATKINS, C. J. C. H. ; DAYAN, P.: Q-Learning. In: *Machine Learning* 8 (1992), Nr. 3, S. 270–292
- [WD99] WANG, X. ; DIETTERICH, T. G.: Efficient Value Function Approximation Using Regression Trees. In: *Proceedings of IJCAI-99 Workshop on Statistical Machine Learning for Large-Scale Optimization*. Stockholm, Schweden, 1999
- [Web91] WEBER, W.: *Einführung in die Betriebswirtschaftslehre*. Wiesbaden : Gabler, 1991
- [Wei02] WEIKER, K.: *Evolutionäre Algorithmen*. 1. Auflage. Wiesbaden : Teubner Verlag, 2002
- [Wer00] WERNER, Hartmut: *Supply Chain Management*. Gabler, 2000
- [Wil97] WILDEMANN, H.: Koordination in Unternehmensnetzwerken. In: *ZfB - Zeitschrift für Betriebswirtschaft* 67 (1997), Nr. 4, S. 417–439

- [Woo02] WOOLRIDGE, M.: *An Introduction to Multiagent Systems*. Wiley, 2002
- [ZD95] ZHANG, W. ; DIETTERICH, T. G.: A Reinforcement Learning Approach to Job-Shop Scheduling. In: *Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann, 1995, S. 176–184
- [Zäp98] In: ZÄPFEL, G.: *Grundlagen und Möglichkeiten der Gestaltung dezentraler PPS-Systeme*. Stuttgart : Kohlhammer, 1998, S. S. 11–53

Anhang

A. Liste Planungsverfahren und Varianten

In diesem Kapitel werden die Änderungsplanungsverfahren von Heidenreich¹, die im Lernverfahren verwendet wurde, tabellarisch gruppiert.

Tabelle A.1.: Planverfahren bei Änderung der Restriktionen am FOK

Aufgaben	Verfahren	Variante
Reduzierung zur Defizitbeseitigung	Sicherheitsbestandsreduzierung	Absolut
		Prozentual
		Dynamisch entsprechend Defizit
Erhöhung zur Überschussbeseitigung	Maximalbestands-erhöhung	Absolut
		Prozentual
		Dynamisch entsprechend Defizit

¹Siehe [Hei06]

A. Liste Planungsverfahren und Varianten

Tabelle A.2.: Elementare Planungsverfahren am KOK

Aufgaben	Verfahren	Variante
Defizitbeseitigung	Leistungsgraderhöhung	Absolut
		Prozentual
		Dynamisch entsprechend Defizit
Einlastung	Nettoangebotserhöhung	Zeitabschnittsfixiert
		Mengenfixiert ohne Splittung Vorwärts terminiert
		Mengenfixiert mit Splittung Vorwärts terminiert
		Mengenfixiert ohne Splittung Rückwärts terminiert
		Mengenfixiert mit Splittung Rückwärts terminiert
		Vorwärts terminierte Verdichtung
		Rückwärts terminierte Verdichtung

Tabelle A.3.: Bedarfsseitige elementare Planungsverfahren am FOK

Aufgaben	Verfahren	Variante
Erhöhung zur Defizitbeseitigung	Ohne Losgruppierung	Periodensynchrone Erhöhung
	Losgruppierung nach Bestellzyklus	Vorwärts terminierte Loserhöhung
		Rückwärts terminierte Loserhöhung
	Losgruppierung nach Bestellpunkt	Vorziehen eines Loszuganges
Erhöhung bis zum Bestandsmaximum	Ohne Losgruppierung	Zeitabschnittsfixierte Erhöhung
		Mengenfixierte Erhöhung ohne Splitting
		Mengenfixierte Erhöhung mit Splitting
	Losgruppierung nach Bestellzyklus	Mengenfixierte Erhöhung ohne Splitting
		Mengenfixierte Erhöhung mit Splitting
Reduzierung bis zum Bestandsminimum	Losgruppierung nach Bestellzyklus	Periodensynchrone Reduzierung
		Vorwärts terminierte Losreduzierung
		Rückwärts terminierte Losreduzierung
	Losgruppierung nach Bestellpunkt	Verzögern eines Loszuganges
	Reduzierung zur Überschussbeseitigung	Losgruppierung nach Bestellzyklus
Vorwärts terminierte Losreduzierung		
Rückwärts terminierte Losreduzierung		
Losgruppierung nach Bestellpunkt		Verzögern des Loszuganges
Reduzierung bis zum Bestandsminimum		Losgruppierung nach Bestellzyklus
	Mengenfixierte Reduzierung ohne Splitting	
	Mengenfixierte Reduzierung mit Splitting	
	Losgruppierung nach Bestellpunkt	Verzögern des Loszuganges

A. Liste Planungsverfahren und Varianten

Tabelle A.4.: Angebotsseitige elementare Planungsverfahren am FOK

Aufgaben	Verfahren	Variante
Erhöhung Überschuss- beseitigung	zur Ohne Losgruppierung	Periodensynchrone Erhöhung
	Losgruppierung nach Bestellzyklus	Vorwärts terminierte Loserhöhung
		Rückwärts terminierte Loserhöhung
	Losgruppierung nach Bestellpunkt	Vorziehen eines Loszuganges
Erhöhung bis zum Bestandsminimum	Ohne Losgruppierung	Zeitabschnittsfixierte Erhöhung
		Mengenfixierte Erhöhung ohne Splitting
		Mengenfixierte Erhöhung mit Splitting
	Losgruppierung nach Bestellzyklus	Mengenfixierte Erhöhung ohne Splitting
		Mengenfixierte Erhöhung mit Splitting
Reduzierung bis zum Bestandsminimum	Losgruppierung nach Bestellzyklus	Periodensynchrone Reduzierung
		Vorwärts terminierte Losreduzierung
		Rückwärts terminierte Losreduzierung
	Losgruppierung nach Bestellpunkt	Verzögern eines Loszuganges
	Reduzierung zur Defizitbeseitigung	Losgruppierung Bestellzyklus
Vorwärts terminierte Losreduzierung		
Rückwärts terminierte Losreduzierung		
Losgruppierung nach Bestellpunkt		Verzögern des Loszuganges
Reduzierung bis zum Bestandsmaximum		Losgruppierung Bestellzyklus
	Mengenfixierte Reduzierung ohne Splitting	
	Mengenfixierte Reduzierung mit Splitting	
	Losgruppierung nach Bestellpunkt	Verzögern des Loszuganges

B. Generieren von Trainingsdaten

Sowohl für das Erlernen der charakteristischen Planverläufe als auch für das Training des Lernverfahrens ist eine große Menge an Ausgangsdaten erforderlich. Liegen zum Erzeugen der Trainingszustände Realdaten, z. B. in Form von Vergangenheitsdaten oder Prognosedaten, vor, so sollten diese als Ausgangsdaten für das Training des Clustering verwendet werden.¹ Alternativ können Trainingsdaten, ähnlich wie bei einem Simulationsexperiment² in einer Simulationsstudie³, automatisch mit vorgegebener realitätsnaher Parametrisierung eines Datengenerators erzeugt und zum Clustering und Training des Lernverfahrens verwendet werden.

Die Parameter des Datengenerators müssen so gewählt sein, dass die erzeugten Planverläufe, wie im Simulationsexperiment, eine realitätsnahe Abbildung möglicher realer Planverläufe des Produktionsnetzwerkes repräsentieren. Es werden die definierten Merkmale der Zustände von Objektknoten sowie Restriktionsgrenzen und Leistungsvereinbarungen berücksichtigt und durch die Verwendung von typischen Zugängen, Abgängen und Beständen⁴ der einzelnen Objektknoten spezifische Planverläufe erzeugt. In Tabelle B.1 ist ein Parametersatz zur Datengenerierung dargestellt.

¹Siehe Kap. 5.1.3.8, S. 94

²Ein *Simulationsexperiment* ist ein „systematischer Plan zur Ausführung einer Menge von Simulationsläufen mit unterschiedlichen Anfangszuständen und Parametereinstellungen zur effizienten Untersuchung eines Modellverhaltens.“ (Siehe [VDI93])

³Eine *Simulationsstudie* ist ein „Projekt zur simulationsgestützten Untersuchung eines Systems. [...] Eine Simulationsstudie kann mehrere Simulationsexperimente umfassen, die ihrerseits aus mehreren Simulationsläufen bestehen können.“ (Ebd.)

⁴Diese können aus Vergangenheitsdaten ermittelt werden oder basieren auf Erfahrungswerten z. B. eines Produktionsplaners.

Tabelle B.1.: Parameter für das Generieren von Trainingsdaten

Parameter	Beschreibung	Wertebereich
n	Anzahl der generierten Planverläufe	\mathbb{N}
PH	Länge des Planungshorizontes für die generierten Planverläufe	\mathbb{N}
min	Mindestbestand für alle generierten Planverläufe	\mathbb{N}
max	Maximalbestand für alle generierten Planverläufe	\mathbb{N}
$init_{\mu}$	Mittelwert für die Auswahl des initialen Bestandes zu Beginn des Planungshorizontes	\mathbb{N}
$init_{\sigma}$	Varianz für die Auswahl des initialen Bestandes zu Beginn des Planungshorizontes	\mathbb{N}
Δ_i	Intervall zwischen Zu-/Abgängen von Fertigungselementen in Planungsintervallen	$\{1, \dots, PH\}$
Z_i^{start}	Erstes Planungsintervall mit einem zyklischen Zu-/Abgang	$\{1, \dots, PH\}$
Z_i^{min}	Minimale Höhe eines zyklischen Zu-/Abgangs	\mathbb{N}
Z_i^{max}	Maximale Höhe eines zyklischen Zu-/Abgangs	\mathbb{N}
w_j	Wahrscheinlichkeit eines punktuellen Zu-/Abgangs in einem Planungsintervall	$[0, 1]$
P_j^{min}	Minimaler punktueller Zu-/Abgang	\mathbb{N}
P_j^{max}	Maximaler punktueller Zu-/Abgang	\mathbb{N}

C. Implementation

In diesem Kapitel wird die Implementation des Lernsystems skizziert. In Kapitel C.1 wird ein Überblick über die gesamte Systemarchitektur und verwendete Komponenten gegeben. In Kapitel C.2 werden Details zur Umsetzung des Clusterings und in Kapitel C.3 Details zur Implementation des maschinellen Lernverfahrens dargestellt. Das System wurde unter dem Projektnamen *RLPP* (Reinforcement Learning in Production Planning) umgesetzt.

C.1. Gesamtsystem

Da das Lernsystem auf einem Produktions*netzwerk* lernt, wurde zur Umsetzung dieses Netzwerkes auf die Multiagententechnologie (MAS¹) zurückgegriffen.² Die einzelnen MFERT-Objektknoten werden dabei jeweils durch ein Agentenobjekt repräsentiert. Die für das Clustering und Training notwendigen Schritte können dann als *Behavior* der Agenten umgesetzt werden.

C.1.1. JADE als Plattform

Als Plattform zur technischen Umsetzung des MAS wurde die Open-Source-API JADE³ verwendet. JADE wird als Open-Source-Projekt kontinuierlich weiter entwickelt.⁴ Des Weiteren folgt JADE bei der Kommunikation zwischen den Agenten dem FIPA⁵-Standard. Der FIPA-Standard vereinheitlicht das Versenden von Nachrichten in MAS auf der Basis einheitlicher Protokolle⁶, mit denen der Datenfluss zwischen den ausgeführten Behavior der einzelnen kommunizierenden Agenten synchronisiert werden kann.

Der Vorteil bei der Verwendung von JADE liegt darin, dass der Programmierer umgesetzte Standards für MAS und zur Kommunikation notwendige Protokolle direkt

¹Details zur Theorie von MAS bspw. in [Fer01]

²Siehe Kap. 3.3.1, S. 64

³[Til]

⁴[BPR01]

⁵[Fou97]

⁶Z. B. ACL in [FL99]

aus der JADE-Bibliothek verwenden kann, ohne diese selbst programmieren zu müssen. Dieses gilt auch für üblicherweise in MAS verwendete Services, wie bspw. der Yellow-Pages-Dienst. Durch die Objektorientierung von JADE können alle umgesetzten Konzepte aus JADE vererbt und somit einfach weiter entwickelt oder angepasst werden. JADE bietet zusätzlich einen Monitor zur Überwachung des gesamten MAS während der Laufzeit.⁷ Dieses erhöht die Effizienz im Implementationsprozess, da das Debugging erleichtert wird. Ebenso kann hierüber der Lernprozess der Agenten überwacht und die Kommunikation zwischen den Agenten verfolgt werden.

Um die Rechenlast während des Trainings auf verschiedene Computer eines Netzwerkes zu verteilen, können mit JADE Agenten auf entfernte Computer migriert und trotzdem weiterhin zentral im Agentenmonitor überwacht werden. JADE selbst ist in der Programmiersprache JAVA implementiert.⁸

C.1.1.1. Klassenhierarchie und Funktion

Abbildung C.2 zeigt den objektorientierten Aufbau des gesamten Lernsystems. Jeder Objektknoten wird durch eine eigene JAVA-Klasse repräsentiert. Da bei den Agenten die funktionale Eigenschaft, d. h. die Ausprägung der Behavior, im Vordergrund steht, und die Behavior⁹ unabhängig vom Typ des Objektknotens im Wesentlichen gleich sind, wurden die Agenten als eigenständige JADE-Agenten umgesetzt.

Die Objektknotenklassen fungieren dabei als Datenbehälter, welche die erforderlichen Datenstrukturen wie

- Restriktionen
- Planverläufe
- Schnittstellen

repräsentieren und die Datenschnittstellen über vereinheitlichte get/set-Methoden für den Zugriff durch die Agenten-Behavior bereitstellen.

MainAgent

Der `MainAgent` wird von der JADE-Plattform gestartet und ist verantwortlich dafür, alle weiteren Objektknotenagenten zu initialisieren. Dazu liest er die Netzwerk- und Objektknotenkonfigurationsdateien ein und startet auf dieser Konfiguration alle erforderlichen Agenten. Der `MainAgent` wartet, bis er von jedem Objektknoten eine Bestätigungsnachricht über die Initialisierung erhalten hat. Außerdem übernimmt der `MainAgent` das Senden der Änderungsanfragen an das Netzwerk. Die Parameter

⁷Siehe Abb. C.1

⁸JAVA Version 1.6, <http://java.sun.com>

⁹Lernschritte im Training, Interaktion

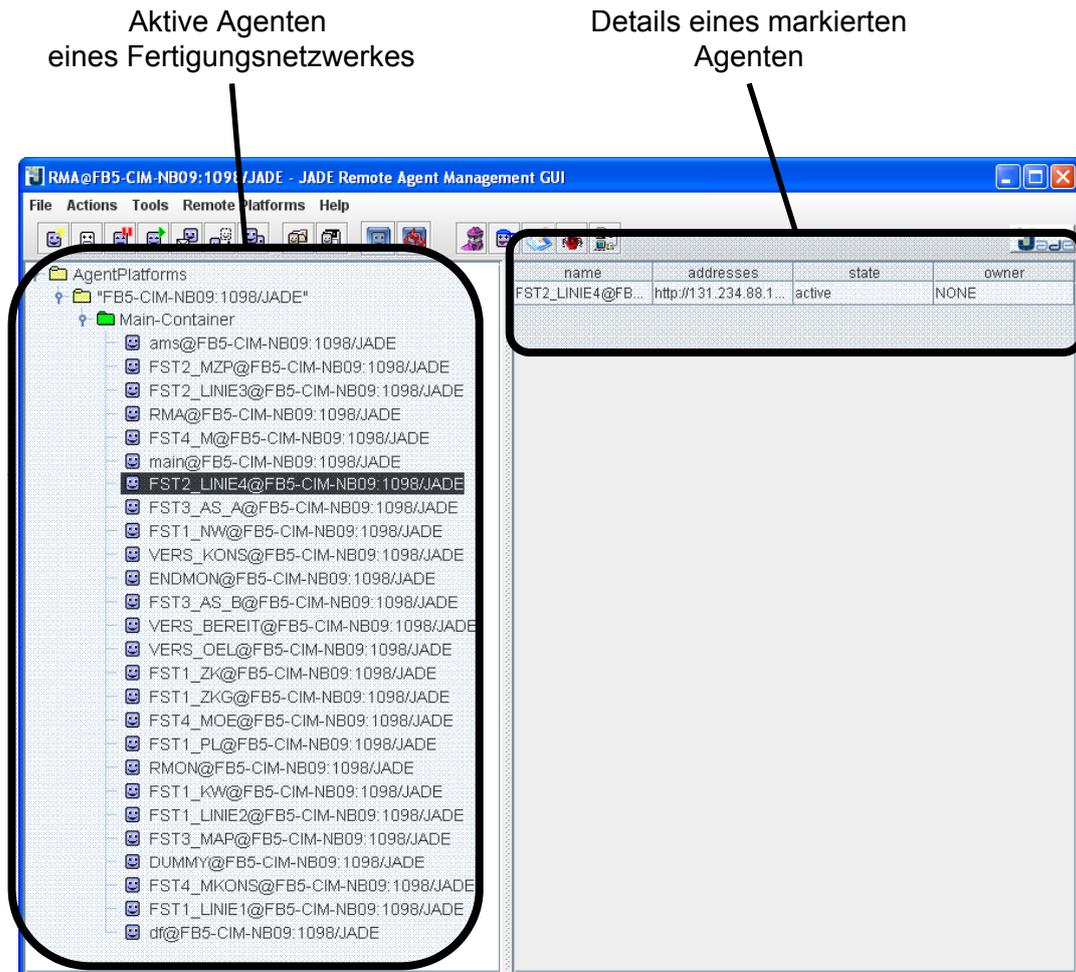


Abbildung C.1.: Screenshot: JADE Agentenmonitor mit initialisiertem Produktionsnetzwerk

C. Implementation

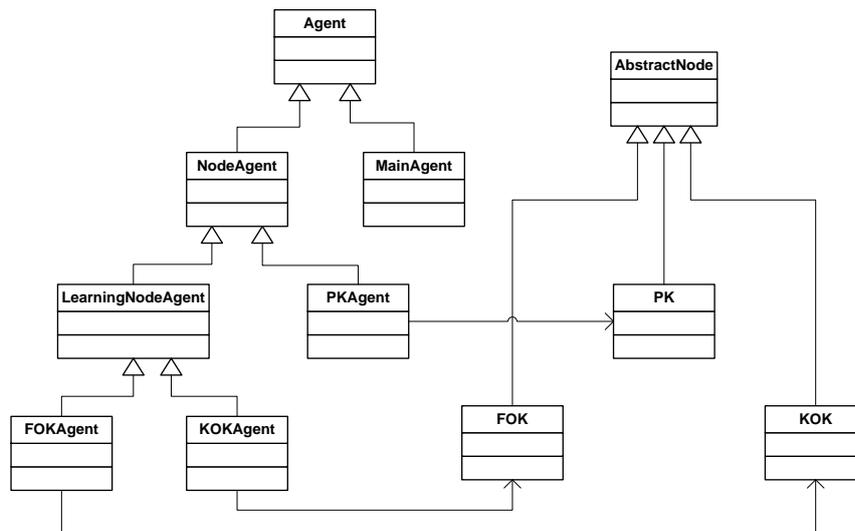


Abbildung C.2.: Klassendiagramm: Vereinfachte Darstellung des Lernsystems

zum Generieren dieser Änderungsanfragen sind als Konstanten in der Agentenklasse festzulegen. Das Generieren von Änderungsanfragen ist im zyklischen Behavior

`GenerateChangeRequestBehavior`

implementiert. Nachdem eine Änderungsanfrage generiert ist, wird dem Agenten eine Instanz des

`InitiateCoordinationBehavior`

hinzugefügt, die das Senden der Anfrage übernimmt und auf die Antwort des Netzwerkes wartet. Während dieser Zeit ist das zyklische

`GenerateChangeRequestBehavior`

blockiert. Erst wenn die Antwort vom Netzwerk empfangen wurde, terminiert das

`InitiateCoordinationBehavior`

und das

`GenerateChangeRequestBehavior`

setzt seine Arbeit fort, indem es prüft, ob nach dieser Koordination eine Neuinitialisierung der Pläne notwendig ist.

Diese Neuinitialisierung wird alle n Koordinationszyklen¹⁰ durchgeführt, das Intervall wird durch die Konstante `REINIT_INTERVAL` festgelegt. Bei einer Neuinitialisierung wird allen Objektknoten eine `REQUEST`-Nachricht mit dem Inhalt „init-plan“ gesendet. Jeder Objektknoten muss die Neuinitialisierung mit einer `INFORM`-Nachricht mit dem Inhalt „init-plan-finished“ quittieren. Diese Nachrichteninhalte sind als Konstanten in der Klasse `MsgIdentifier` hinterlegt.

NodeAgent

Der `NodeAgent` ist die Basisklasse für alle Agenten, die Objektknoten eines Netzwerkes repräsentieren. Dementsprechend beinhaltet dieser Agent solche Methoden, die für alle Objektknoten nützlich sind. Dazu gehören Methoden, um die AID eines Agenten zu einem Objektknoten mit bekannter Objektknoten-ID beim Directory Facilitator zu erfragen und zu cachen. Ebenso gibt es Funktionalitäten, um ACL-Nachrichten der ProductionNetworkOntology in `QuantityChangeRequest`-Objekte zu decodieren und umgekehrt solche Objekte in eine String-Beschreibung zu konvertieren, die als Content einer ACL-Nachricht verwendet werden kann. Weiterhin wird eine Methode geboten, um Basis-ACL-Nachrichten zu erzeugen, die bereits für die Verwendung als Planänderungsanfragen mit den entsprechenden Werten für das Kommunikationsprotokoll in der JADE-Ontologie des RLPP vorkonfiguriert sind.

LearningNodeAgent

Der `LearningNodeAgent` ist die Basisklasse für solche Agenten, die im Rahmen des Q-Learnings Steuerungsregeln lernen. Hier werden insbesondere Datenstrukturen wie die Q-Wert-Tabelle und die Menge der für den zugehörigen Objektknoten anwendbaren Aktionen verwaltet. Außerdem sind hier alle Funktionalitäten implementiert, die zum Loggen des Lernfortschrittes existieren.¹¹

AbstractNode

Der `AbstractNode` ist die abstrakte Oberklasse für alle Objektknoten. Sie definiert eine ID für alle Objektknoten und die abstrakte Methode `initializePlan()`.

PkAgent/PK

Jeder PK verwaltet die Fertigung eines Artikels und speichert dazu als Stammdaten die zugehörige Stückliste (BOM)¹², sowie analog dazu eine Liste mit benötigten Ka-

¹⁰Entsprechend der Konfiguration der Lernepisoden

¹¹Loggen der Q-Wert-Entwicklungen und der Zustands-Cluster-Zuordnungen

¹²Engl. *bill of materials*

C. Implementation

pazitäten (BOC)¹³. Die Daten in dieser Liste beziehen sich jeweils auf die benötigten Input-Faktoren bzw. Kapazitäten zum Fertigen einer Einheit des Artikels.

Der `PkAgent` hat im Rahmen der Verhandlungen eine koordinierende Funktion. Für jede Anfrage berechnet er aus den Stücklisten die resultierenden Änderungen und sendet entsprechende Anfragen an die zugehörigen FOK und KOK. Insbesondere für die Angebotskoordinationen ist dies ein größerer Aufwand. Bei der Änderung der Menge eines Input-Faktors wird über die Stückliste ermittelt, um wie viel sich die produzierte Menge ändert. Für jeden weiteren Input-Faktor müssen nun wieder unter Berücksichtigung der Stückliste die geänderten Mengen ermittelt und angefragt werden. Gleichzeitig muss die geänderte Menge auch in Angebotsrichtung weitergegeben werden.

FokAgent/FOK

Im Gegensatz zu den PK haben die FOK ein Planobjekt, das den Planverlauf und die aktuellen Restriktionsgrenzen für jede Periode speichert. Die Methode

```
initializePlan()
```

initialisiert diesen Plan unter Verwendung der Parameter aus der Objektknotenkonfiguration neu. Der `FokAgent` beinhaltet aktuell die Funktionalitäten zum Verarbeiten von Planänderungen und Senden von Änderungsanfragen an benachbarte PK. Dabei wird der Ablauf jeder Koordination vom

```
PlanChangeCoordinationBehavior
```

gesteuert. Dieses Behavior löst nach dem Eintreffen einer Änderungsanfrage eine Planbestandsrechnung aus, wählt eine Aktion aus und startet im Fall einer globalen Aktion ein zusätzliches

```
GlobalActionInitiatorBehavior
```

das die nötigen Änderungsanfragen für die benachbarten PK generiert und entsprechend auf Antworten wartet.

KokAgent/KOK

Die Funktionalitäten des KOK sind im `KokAgent` umgesetzt. Dieser ist analog zum `FokAgent` implementiert.

¹³Engl. *bill of capacities*

Tabelle C.1.: Nachrichtfelder und Inhalte der RLPP-Koordinationsnachrichten

Nachrichtenfeld	Inhalt
Protokoll	rlpp-request
Sprache	fipa-sl
Ontologie	Production- Network- Ontologie
Inhalt	siehe Code unten

C.1.1.2. Technische Umsetzung der Koordination

Die Koordination wird über FIPA-ACL-Nachrichten durchgeführt. Die Art der Koordination, z. B. das Akzeptieren oder Ablehnen einer Angebotsänderung, wird über den Nachrichtentyp der ACL-Nachricht festgelegt. Der Nachrichtentyp löst im sendenden und empfangenden Agenten ein entsprechendes, zur Verarbeitung der Koordinationsart notwendiges Behavior aus, indem die ACL-Nachricht ausgewertet und bearbeitet wird. Die Typen und Ausprägungen der Nachrichten werden über eine Ontologie definiert. JADE unterstützt nativ die Verwendung von Ontologien.

Eine Koordination zwischen den Objektknoten wird durch eine Nachricht vom Typ REQUEST initiiert. Dabei muss diese Nachricht die in Tabelle C.1 aufgelisteten Eigenschaften aufweisen. Auf diese Anfrage reagiert der angefragte Agent mit einer AGREE- oder REFUSE-ACL-Nachricht, je nachdem, ob er die Änderung annehmen kann oder nicht. Sendet ein Agent ein REFUSE, so ist die Koordination für ihn abgeschlossen, er wartet auf keine weiteren Nachrichten mehr. Sendet er hingegen ein AGREE, so wartet er anschließend noch auf eine Bestätigungsnachricht in Form eines CONFIRM oder CANCEL.

Im ersten Fall wird ihm vom anfragenden Agenten signalisiert, dass die angefragte Änderung durchgeführt werden soll. Im Fall eines CANCEL soll die Änderung nicht durchgeführt werden, z. B. weil andere Partizipanten die Änderungsanfrage nicht erfüllen konnten. In diesem Fall wird der Plan wieder auf seinen Ausgangszustand zurückgesetzt. Erst nach Erhalt dieser Bestätigungsnachricht ist die Koordination für den Objektknoten beendet, d. h. erst dann verarbeitet er die nächste Änderungsanfrage, auch wenn diese bereits früher eingetroffen ist. Die Bestätigungsnachricht wird von den Objektknoten sequenziell verarbeitet.

C.1.1.3. Konfiguration

Um die Implementation flexibel zu gestalten, sind viele Funktionen über Konfigurationsdateien an spezielle Anwendungsfälle anpassbar. Dieses gilt insbesondere für die Ausgestaltung des Produktionsnetzwerks und die zum Lernen zugelassenen Änderungsplanungsverfahren. Die Konfiguration erfolgt auf der Basis von XML-Dateien, deren strukturelle Integrität bzw. deren Syntax über korrespondierende XML-Schema-Definitionen abgesichert wird. Die Verwendung von XML erlaubt eine einfache Erweiterung oder Anpassung der Systemkonfigurationsdateien an neue oder geänderte Anforderungen an zentraler Stelle.

Zur Konfiguration des Systems sind im Wesentlichen fünf zentrale Konfigurationsdateien notwendig:

rlpp.properties: Globale Systemkonfiguration. Hier wird z. B. festgelegt, für welche Objektknoten die Clusterabbildungen und die Entwicklung der Q-Werte gespeichert werden sollen. Weiterhin wird die Datenbankverbindung für die Speicherung der Cluster und Q-Werte konfiguriert und alle weiteren notwendigen Einstellungen werden vorgenommen.

<netzwerk>.xml: Struktur des Anwendungsfallproduktionsnetzwerkes. Es kann ebenso eine beliebige Datei konform zum XML-Schema `network.xsd` verwendet werden.

<netzwerk-nodes>.xml: Parameter für die einzelnen Objektknoten des Netzwerkes. Beinhaltet Restriktionsgrenzen sowie Parameter zum generieren von Planverläufen. Es kann ebenso eine beliebige Datei nach dem XML-Schema `node_config.xsd` verwendet werden.

Aktionen-OK.xml: Definition der zur Verfügung stehenden Aktionen an den Objektknoten. Hier wird weiterhin konfiguriert, in welchen Klassen die Planungsverfahren implementiert sind und unter welchen Bedingungen die einzelnen Verfahren anwendbar sind. Ebenso kann eine beliebige Datei nach dem XML-Schema `actions_config.xsd` verwendet werden.

logging.properties: Konfiguration der Logausgaben, d. h. der Log-Handler (Konsole, Dateien) und der Log-Level. Unterstützt das Debugging des Systems bei möglichen Erweiterungen.

Die Pfade der Konfigurationsdateien werden beim Systemstart, wie in der Abbildung C.3 dargestellt, den instanziierten Agenten über die JADE-Plattform übergeben. Die zu instanziiierenden Agenten werden durch die JADE-Plattform über die Netzwerkkonfigurationsdatei ermittelt.

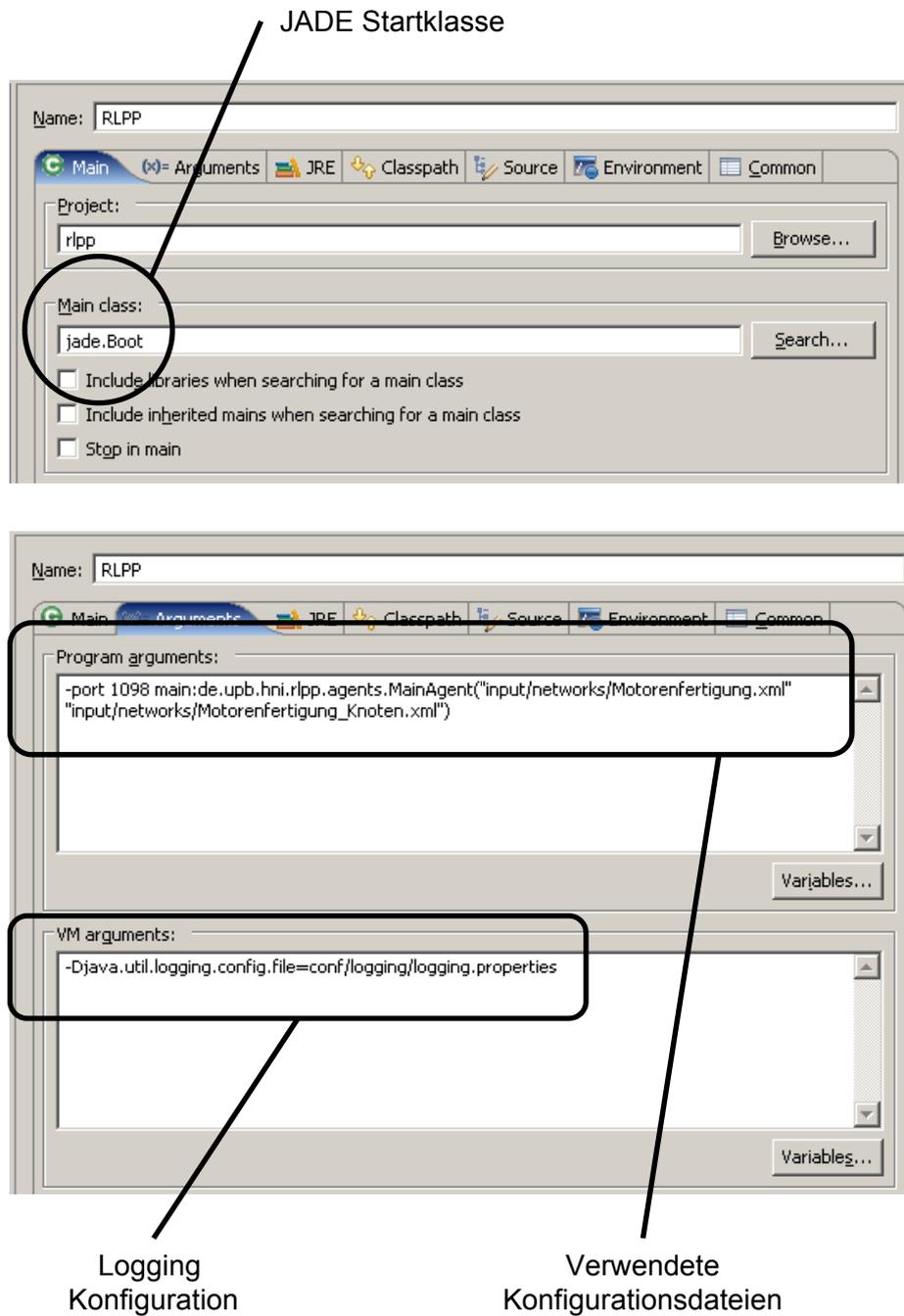


Abbildung C.3.: Screenshot: Konfiguration des RLPP

C.2. Clustering

Für das Erlernen der charakteristischen Planverläufe wurde sowohl das parametrisierte Generieren von Trainingsdaten als auch das k -means-Clustering mit der problem-spezifischen Distanzfunktion implementiert. Die wesentlichen Funktionalitäten des k -means-Algorithmus sind in der Klasse `KMeans` zu finden, während die Distanzfunktion des Clusterings in der Klasse `Distance` implementiert ist. Als Datenstruktur für die Trainingsdaten dienen aus Effizienzgründen zwei-dimensionale Arrays. Diese Datenstrukturen sind jedoch in die speziellen Klassen

`PlanDoubleMatrix`

und

`ClusterTrainingData`

gekapselt, die zusätzlich Methoden für den Zugriff auf die Daten sowie häufig verwendete Operationen wie das Berechnen von gewichteten Summen, Auffinden von Restriktionsverletzungen oder das Anwenden von Planänderungen bereitstellen.

Sowohl für das Generieren der Trainingsdaten wie auch das Clustering sind zahlreiche Möglichkeiten der Konfiguration vorgesehen. Die nötigen Parameter für das Erlernen von charakteristischen Planverläufen für einen Objektknoten lassen sich in speziellen XML-Dateien konfigurieren, die einem dafür definierten Schema folgen müssen. In diesen Konfigurationen¹⁴ werden Parameter zum Erzeugen der Trainingsdaten definiert. Weiterhin sind einige allgemeine Parameter für das Clustering festzulegen, insbesondere die Anzahl der zu erzeugenden Cluster k sowie die Abbruchkriterien für das Clusterverfahren in Form der Konvergenztoleranz und der maximalen Anzahl durchzuführender Iterationen. Um aus den charakteristischen Planverläufen die für das Lernverfahren benötigten abstrahierten Zustände generieren zu können, sind außerdem Angaben über die Bezeichnung des anfragenden Objektknotens sowie die Art der beim Generieren der Planverläufe angewendeten Planänderungen zu machen.

Neben diesen instanzspezifischen Konfigurationen lassen sich weitere Parameter in einer systemweiten Konfiguration festlegen. Hier ist insbesondere die Möglichkeit gegeben, die Konfigurationsdatei für das durchzuführende Clustering anzugeben. Neben der Angabe von einzelnen Konfigurationsdateien ist auch die Möglichkeit für einen Batch-Modus gegeben, indem ein Verzeichnis mit einer beliebigen Anzahl von Konfigurationen spezifiziert wird, um die einzelnen Instanzen sequenziell bearbeiten zu lassen. Damit besteht die Möglichkeit, die abstrahierten Zustände für alle Objektknoten eines Produktionsnetzwerkes mit einem Programmablauf zu generieren, sofern für jeden Objektknoten vorher die nötige Konfiguration der Trainingsdaten und des Clusterings angelegt wurde.

¹⁴Siehe Tab. B.1, 212

C.2.1. Ausgabemöglichkeiten

Um die erlernten abstrahierten Zustände weiterverwenden zu können, sind verschiedene Möglichkeiten der Ausgabe vorgesehen. Diese können in der bereits erwähnten globalen Konfigurationsdatei beliebig kombiniert werden, um alle gewünschten Ausgabeformate zu erhalten. Dabei stehen die folgenden Ausgabemethoden zur Verfügung:

- Ausgabe der vollständigen Clusterdaten im CSV-Format
- grafische Ausgabe der Clusterdaten als Diagramm
- Speichern der abstrahierten Zustände in einer SQL-Datenbank

Die Ausgabe im CSV-Format erzeugt für jedes Cluster eine Datei, in der in der ersten Zeile jeweils der aus diesem Cluster errechnete charakteristische Planverlauf gespeichert ist. Darunter folgen zeilenweise die Planverläufe aus den Trainingsdaten, die diesem Cluster zugewiesen wurden. Die CSV-Ausgabe eignet sich insbesondere für eine quantitative Analyse der erzeugten Cluster.

Um einen Überblick über die Beschaffenheit der Cluster zu erhalten, kann auch eine grafische Darstellung der jeweiligen charakteristischen Planverläufe und zugehörigen Cluster erzeugt werden. Dabei wird das Generieren der Diagramme von dem Programm `gnuplot`¹⁵ übernommen. Nach Beendigung des Clusterings wird für jedes Cluster ein `gnuplot`-Skript generiert, das alle Daten über das Cluster und den zugehörigen charakteristischen Planverlauf enthält. Anschließend wird `gnuplot` aufgerufen und verwendet diese Daten als Eingabe zum Erzeugen eines Diagramms für jedes Cluster. Hierzu können verschiedene Einstellungen bzgl. der Formatierung des Diagramms sowie des erzeugten Dateiformates in der globalen Konfiguration vorgenommen werden. Dort kann auch spezifiziert werden, dass für jedes Cluster ein zusätzliches Diagramm generiert werden soll, das lediglich den charakteristischen Planverlauf für dieses Cluster enthält. Damit ist die Möglichkeit gegeben, sich einen einfachen Überblick über die erlernten charakteristischen Planverläufe zu verschaffen.

Da die eigentliche Verwendung der charakteristischen Planverläufe beim Erzeugen der abstrahierten Zustände für das Lernverfahren liegt, muss auch hierfür eine entsprechende Ausgabe bereitgestellt werden. Zu diesem Zweck lassen sich alle Informationen zu den abstrahierten Zuständen auch in einer SQL-Datenbank ablegen. Diese Informationen können später vom Lernverfahren dazu verwendet werden, die einzelnen Zustände auf einen der abstrahierten Zustände abzubilden. Generell kann die Ausgabe in jede JDBC-konforme Datenbank erfolgen. Getestet wurde die Implementierung mit

¹⁵`gnuplot` ist ein frei verfügbares Programm zur grafischen Darstellung von Daten. Nähere Informationen und Programmversionen für alle gängigen Plattformen sind unter <http://www.gnuplot.info> verfügbar. Im Rahmen dieser Arbeit kam `gnuplot` Version 4.0 zum Einsatz.

einer Microsoft SQL-Datenbank. Alle nötigen Einstellungen zur Datenbankanbindung sind ebenfalls in der globalen Konfiguration vorzunehmen.

Neben den beschriebenen Ausgabemöglichkeiten für die Ergebnisse des Clusterings kann zusätzlich veranlasst werden, die zuvor generierten Trainingsdaten ebenfalls im CSV-Format auszugeben und abzuspeichern. Dies wird über einen entsprechenden Eintrag in der globalen Konfiguration gesteuert. Zusammen mit der Möglichkeit, alternativ zu den Trainingsdaten-Parametern eine solche Datei im Rahmen der Cluster-Konfiguration anzugeben, ermöglicht dies die mehrfache Verwendung eines fixen Trainingsdaten-Satzes, was insbesondere für das Testen von verschiedenen Parameter-Konfigurationen hilfreich ist.

C.2.2. Integration in das Lernverfahren

Das Lernverfahren ist im Multiagentensystem umgesetzt, indem für die Objektknoten eines Produktionsnetzwerkes jeweils einzelne Agenten existieren. Die Integration des Clusterverfahrens erfolgt im Wesentlichen über die Datenbank, in der vom Clusterverfahren die Informationen über die abstrahierten Zustände abgelegt werden. Das Lernverfahren verwendet diese Informationen, um einzelne Zustände auf die erlernten abstrahierten Zustände abzubilden.

Zunächst werden mithilfe der beschriebenen Verfahren auf Basis von realen oder generierten Trainingsdaten die charakteristischen Planverläufe gelernt. Die daraus resultierenden abstrahierten Zustände werden für jeden Objektknoten in einer zentralen Datenbank abgelegt. Sobald im Rahmen des Q-Learnings nach dem Ausführen einer Aktion eine Aktualisierung eines Q-Wertes durchgeführt werden soll, ist eine Abbildung des aktuellen Zustandes auf einen abstrahierten Zustand notwendig. Zu diesem Zweck wurde das Lernverfahren derart erweitert, dass eine solche Abbildung auf Basis der in der Datenbank verfügbaren abstrahierten Zustände vorgenommen wird. Nachdem aus der Datenbank alle potenziell passenden abstrahierten Zustände auf Basis der Art der eingetretenen Planänderung und des anfragenden Objektknotens ausgelesen wurden, wird die Distanzfunktion des Clusterings verwendet, um den passenden abstrahierten Zustand mit der größten Ähnlichkeit im Bezug auf den charakteristischen Planverlauf zu identifizieren. Die Aktualisierung des Q-Wertes erfolgt dann zum so gefundenen Cluster.

Abbildung C.4 zeigt die am Ablauf eines Lernschrittes beteiligten Klassen und Methodenaufrufe als Sequenzdiagramm¹⁶. Die abstrakte Klasse `LearningNodeAgent` dient als Basisklasse für die Agenten der FOK und KOK. Zu einem Zustand s , der Informationen über die eingetretene Planänderung und den anfragenden Objektknoten enthält, erfragt der Agent vom `ClusteringDAO` mit der Methode `getClusters(s)`

¹⁶Dabei handelt es sich um eine leicht vereinfachte Darstellung. Einige Aufrufe und Methoden-Parameter wurden zur besseren Übersichtlichkeit nicht explizit dargestellt.

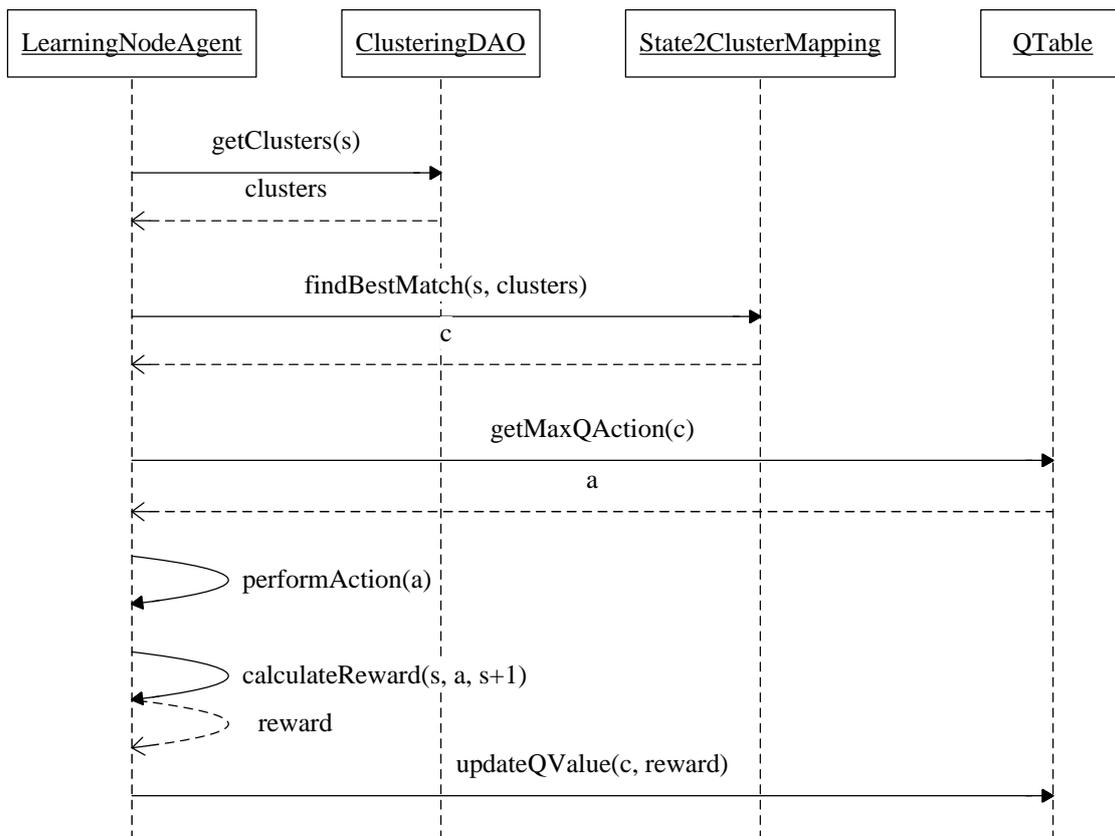


Abbildung C.4.: Sequenz: Ablauf eines Lernschrittes

die Menge der abstrahierten Zustände, die prinzipiell zur Generalisierung des betrachteten Zustandes in Frage kommen. Das `ClusteringDAO` übernimmt die Kommunikation mit der Datenbank und liest alle zum Zustand s passenden abstrahierten Zustände aus.

Um aus dieser Menge von abstrahierten Zuständen den zu finden, der den aktuellen Zustand im Hinblick auf den charakteristischen Planverlauf am besten repräsentiert, wird die statische Methode `findBestMatch(s, clusters)` der Helper-Klasse `State2ClusterMapping` verwendet. Wenn der passende abstrahierte Zustand gefunden ist, kann aus der `QTable` diejenige Aktion ausgelesen werden, die aktuell den maximalen Q-Wert bietet. Nach der Ausführung dieser Aktion werden die Effekte bewertet und in `calculateReward(s, a, s+1)` zu einem numerischen Reward quantifiziert. Dieser wird schließlich verwendet, um den Q-Wert auf Ebene des zuvor identifizierten abstrahierten Zustandes zu aktualisieren.

C.3. Lernfunktion und Training

Das maschinelle Lernverfahren wurde angelehnt an den konzeptionellen Ablauf des Trainings implementiert, dabei aber an die Erfordernisse eines MAS angepasst. Das bedeutet, dass die im Lernverlauf ausgeführten Lernschritte als Behavior der Agenten umgesetzt wurden.

Es ergeben sich zwei Kontrollflüsse, zum einen für lokale und zum anderen für globale Änderungsplanungsverfahren. Der Grund dafür ist, dass lokale Änderungsplanungsverfahren ohne Koordination auskommen, da sie lokale Planänderungen durchführen, während globale Änderungsplanungsverfahren durch die notwendige Koordination komplexere Protokolle zur Abstimmung von Anfragen und Antworten durchlaufen müssen. Zur Umsetzung der Planverläufe im Lernverfahren wird auf die auch im Clustering verwendete `Plan`-Klasse zurückgegriffen, die intern alle Restriktionen und Planwerte je Periode verwaltet. Das Lernverfahren selbst, also die definierten Kostenfunktionen zur Rewardberechnung, werden zentral in der Klasse `QLearner` umgesetzt. Die Aktualisierung des clusterbasierten Q-Wertes wird durch die Methode `qUpdate()` der Klasse `QUpdate` implementiert.

C.3.1. Umsetzung und Überwachung des Trainings

Der Ablauf des Trainings wurde, wie in Abbildung C.5 dargestellt, in sechs Schritte unterteilt und dort durch entsprechende Behavior oder Methoden innerhalb des Behavior implementiert. Die dabei verwendeten Behavior sind:

PCCB: `PlanChangeCoordinationBehavior`, welches implizit die Koordinationsprotokolle umsetzt und daher für alle notwendigen Aktionen im Rahmen der Koordination zwischen Agenten zuständig ist

GAIB: `GlobalActionInitiatorBehavior`, welches zusätzlich zum PCCB bei alternativen oder komplementären Anfragen oder Antworten die entsprechende Überwachung und Sequenzierung beim Einsammeln, Auswerten und Beantworten von Nachrichten übernimmt.

Das PCCB ist im Prinzip vom JADE-Behavior `SerialBehavior` abgeleitet, welches eine serielle Bearbeitung aller abgeleiteten Behavior sicherstellt. Das GAIB ist ebenfalls von dem JADE-`SerialBehavior` abgeleitet, da hier zeitlich versetzt eintreffende Nachrichten gesammelt und sequenziell abgearbeitet werden müssen.

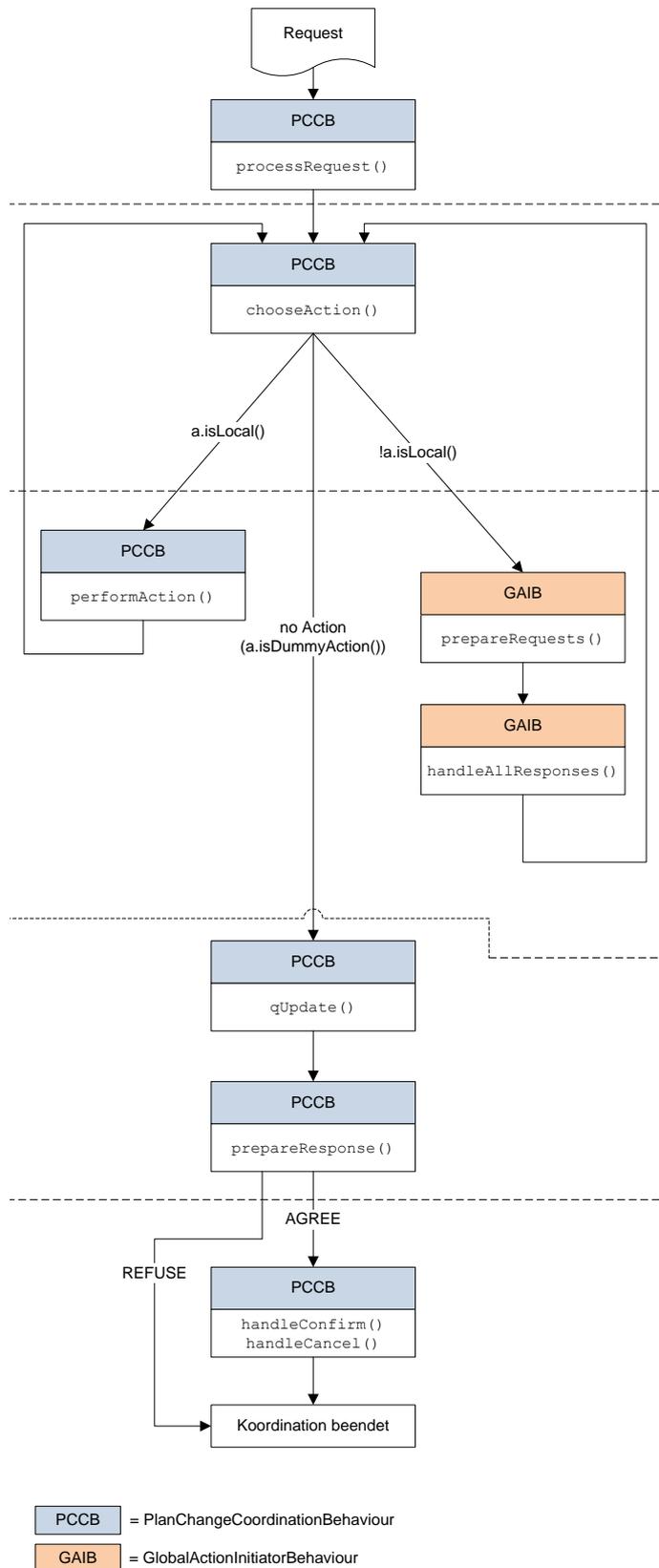


Abbildung C.5.: Technischer Ablauf einer Koordination im Lernsystem

Start einer Koordination und Lernepisode durch einen Request

In Schritt 1 wird eine Anfrage des Initiators der Koordination als Request-Nachricht an den Partizipanten gesendet.¹⁷ Der Request wird beim Empfänger durch die Methode

```
processRequest ()
```

decodiert. Weiterhin wird der aktuelle Plan des partizipierenden Objektknotens zur späteren Rewardberechnung gespeichert. Danach wird eine Planbestandsrechnung auf Basis des Requests durchgeführt und im nächsten Prozessschritt im Falle von Restriktionsverletzungen so der Bedarf einer Änderungsplanung festgestellt.

Restriktionsverletzung und Änderungsplanung

Die Methode

```
chooseAction ()
```

prüft einen Plan auf Zulässigkeit durch erkennen etwaiger Restriktionsverletzungen. Falls keine Restriktionsverletzung vorliegt, ist auch keine Änderungsplanung erforderlich. Über die Methode

```
prepareResponse ()
```

und

```
handleConfirm ()
```

wird der Request bestätigt.

Lokales Lernen

Bei einem unzulässigen Plan mit Restriktionsverletzungen wird der Planverlauf auf einen durch das Clustering abstrahierten Zustand abgebildet und zu diesem Zustand die Aktion a mit dem maximalen Q -Wert gesucht und im DataStore gespeichert. Während der Änderungsplanung ausgeführte Aktionen werden nur einmal ausgewählt. Alternativ kann eine Aktion auch über die ϵ -greedy-Strategie ausgewählt werden.¹⁸ Handelt es sich um eine globale Änderungsplanungsaktion, so wird anstelle ihrer direkten Ausführung durch `performAction` für die spätere Verarbeitung eine Instanz des `GlobalActionInitiatorBehavior` registriert.

Handelt es sich um eine lokale Änderungsplanungsaktion, so wird im nächsten Schritt über

¹⁷Details zu den Konzepten Initiator und Partizipant siehe [Hei06]

¹⁸Siehe Kap. 5.3.1.3

`performAction()`

eine Änderungsplanung auf dem vormals kopierten Plan durchgeführt. Anschließend wird in der Methode

`qUpdate()`

der Reward berechnet und der Q-Wert der ausgeführten Änderungsplanungsaktion im aktuellen Cluster aktualisiert.

Globales Lernen

Bei einer globalen Aktion wird durch das neu registrierte Behavior eine Nachricht an den Empfänger der Anfrage gesendet und diese dort verarbeitet. Durch die Methode

`prepareRequests()`

werden die notwendigen ACL-Nachrichten präpariert und dabei durch Anwendung des globalen Änderungsplanungsverfahrens die entsprechenden anzufragenden Änderungswerte (Bedarfe, Angebote) berechnet. Durch

`handleAllResponses()`

wird geprüft, ob alle angefragten Agenten geantwortet haben. Ist dieses der Fall, werden alle Änderungen in den Plan eingerechnet. Wurde mindestens eine Ablehnung gesendet, wird die Anfrage an die bestätigenden Agenten durch eine CANCEL-Nachricht storniert.

Nachdem die Koordination abgeschlossen wurde, wird im vierten Schritt in der Methode `qUpdate()` der Reward berechnet und der Q-Wert der ausgeführten Änderungsplanungsaktion im Cluster aktualisiert. Weiterhin muss der partizipierende Agent durch

`prepareResponse()`

eine Nachricht an den ursprünglichen Initiator der Anfrage senden. Diese ist, entsprechend der Gültigkeit des Planes, eine Ablehnung der Anfrage REFUSE oder eine Bestätigung der Anfrage AGREE.

Abschluss einer globalen Koordination

Wurde im vierten Schritt die Planänderung vom Partizipanten abgelehnt (REFUSE), so endet im fünften Schritt unmittelbar die Koordination zwischen den Agenten. Wurde

eine Bestätigung (AGREE) vom Partizipanten gesendet, so wird auf ggf. noch ausstehende Nachrichten gewartet. Dieses kann z. B. eintreten, wenn bei der globalen Änderungsplanung als Aktion Bedarfe auf mehrere alternative Lieferanten verteilt werden müssen. Hat der Initiator eine Bestätigung aller angefragten Agenten empfangen, so wird die berechnete Änderung in den Plan eingerechnet. Im Falle mindestens einer Ablehnung wird der gesicherte Backup-Plan wiederhergestellt.

Abschluss der Lernepisode

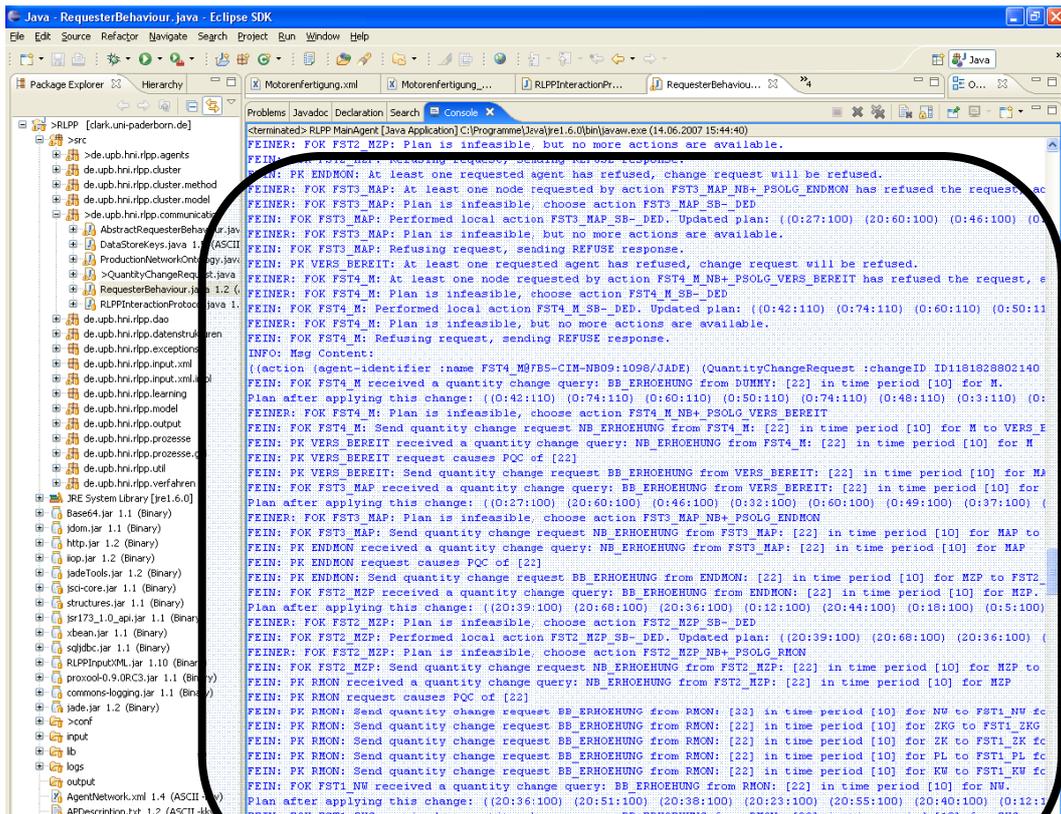
Die nächste Lernepisode beginnt auf dem aktuellen Zustand einer erneuten Änderungsanfrage. Alternativ kann ein neuer Ausgangszustand generiert werden. Dieser Vorgang wiederholt sich, bis ein benutzerdefinierter Abbruchpunkt des Trainings erreicht wurde.

Trainingsüberwachung

Die Überwachung des Trainings findet auf der Basis eines Protokolls statt. Dieses Protokoll, welches in seiner Detaillierung konfiguriert werden kann, wird in der Konsole, z. B. in Eclipse dargestellt in Abbildung C.6, ausgegeben. Hierüber kann der Benutzer über die protokollierte Koordination das Fortschreiten des Trainingsprozesses überwachen.

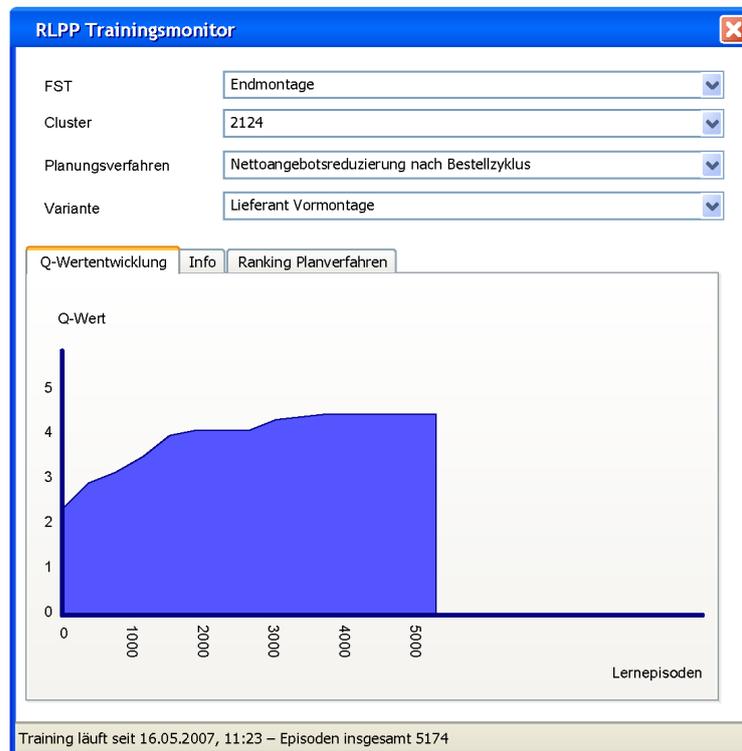
Um die Überwachung benutzerfreundlicher zu gestalten, kann die in Abbildung C.7 dargestellte grafische Oberfläche umgesetzt werden, in der die Entwicklung der Q-Werte je Objektknoten, Cluster und zugelassener Aktion direkt überwacht werden kann. In Abbildung C.7(a) ist unter dem Reiter „Q-Wertentwicklung“ dargestellt, wie sich der Q-Wert einer Änderungsplanungsvarianten eines Clusters im Objektknoten der FST „Endmontage“ während des bisherigen Trainings entwickelt hat. In Abbildung C.7(b) sind weitere Informationen zum aktuellen Cluster und Fortschritt des Trainings einsehbar. Es wird dargestellt, wie viele Lernepisoden und entsprechende Q-Updates auf diesem Cluster durchgeführt wurden. Weiterhin werden der charakteristische Planverlauf oder auch der Centroid des ausgewählten Clusters und seine assoziierten Planverläufe als Graph über die Planungsperiode und Menge angezeigt. Auf dem nicht dargestellten Reiter „Ranking Planverfahren“ kann für den Benutzer die aktuelle Sortierung der Q-Werte der Planungsverfahren aufgelistet werden, sodass das aktuell am besten bewertete Verfahren abgelesen werden kann. Diese Liste bestimmt dann auch direkt die Reihenfolgepriorität der zu generierenden Regeln.

Grundsätzlich können alle gelernten Q-Werte als CSV-Datei zur Weiterverarbeitung in Tabellenkalkulationsprogramme ein- oder zum Ausdruck ausgegeben werden. Die Q-Werte werden ebenso in einer Datenbank zur Weiterverarbeitung gespeichert.

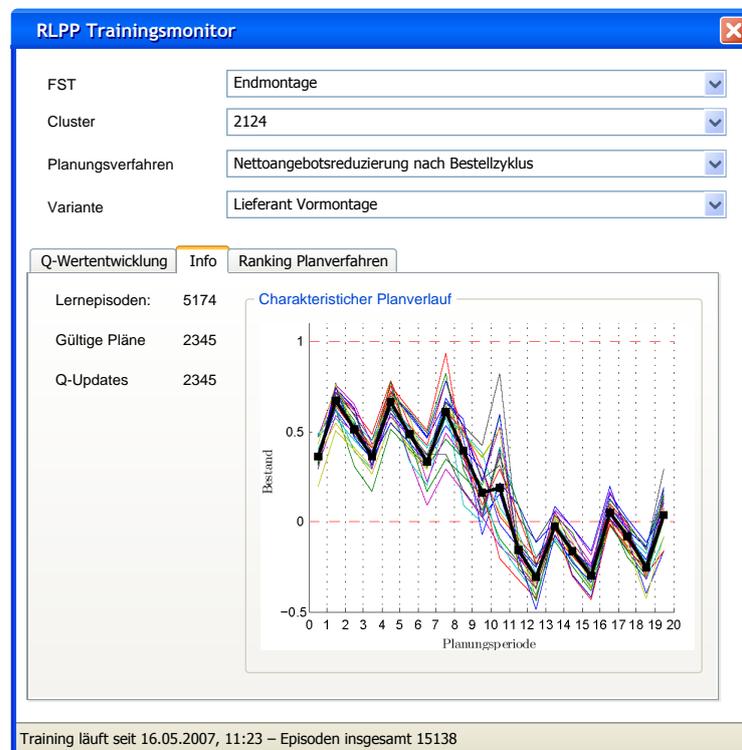


Koordinationsprotokoll zwischen den aktiven Agenten

Abbildung C.6.: Screenshot: Trainingsprotokoll mit Koordinationen in Eclipse



(a) Ansicht: Q-Wert-Entwicklung



(b) Ansicht: Weitere Informationen

Abbildung C.7.: Screenshot: UI-Konzept eines Überwachungsmonitors für das Lernverfahren

Index

- γ , 40
- $\max(Q)$, 40
- Ähnlichkeit
 - Pläne, 83
- Ähnlichkeitsmaß, 52, 54, 58
- Änderungsplanung, 3, 27
 - Entscheidungsprozess, 22
 - global, 26
 - Initiator, 100
 - kooperative, 5, 187
 - lokal, 25
 - Partizipanten, 100
 - Vorlaufzeitverschiebung, 17
- Änderungsplanungsverfahren, 3
 - Parameter, 24
 - Priorität, 23
 - Variante, 24
 - Zustand, 24
- Abstraktion, 60
 - Effektiv, 41
 - Effizient, 41
- Abstraktionsfunktion, 60
- Abstraktionsverfahren, 47
 - explizit, 53
 - implizit, 51
- Agent, 34
- Aktion, 34, 37
- Anfrage
 - Ablehnung, 25
 - Bestätigung, 25
 - geänderte Bestätigung, 25
 - Gegenvorschlag, 21
 - global, 25
 - Kompensation, 21
 - Weitergabe, 21
- Angebot, 3, 15
 - Brutto-, 15
 - Netto-, 15
- Approximation, 47
- Ausgangsdaten, 40
- automatisiert, 27
- Automatisierte Systeme, 28
- Automatisierung, 4
- Bedarf, 3
 - Brutto-, 16
 - Netto-, 16
- Beschaffungssteuerung
 - Bestellpunktverfahren, 17
 - Bestellzyklusverfahren, 17
- Betriebsmittel, 3
- Bewertungsfunktion, 30
- Cluster, 74
- Clusteranzahl
 - Validierung, 157
- Clustering, 74, 167, 174
 - k -Means-Algorithmus, 143
 - Clustermittelpunkt, 92
 - Anfragender OK, 77
 - Art der Planänderung, 77
 - Beispiel Distanzberechnung, 89
 - Effektivität, 161
 - Effizienz, 162
 - harte Kriterien, 76, 79
 - harte Merkmale, 79
 - Kombinierte Distanz, 88
 - Normierung, 77
 - Quantitative Distanz, 88

- Restriktionsgrenzen, 77
- strukturelle Distanz, 85
- Terminierung, 98
- Training, 94
- Trainingsdaten, 211
- Validierung, 153
- weiche Kriterien, 76, 79
- Clusteringverfahren, 52
 - dynamisches, 52
- Descison-Support-System, 23
- Diskontfaktor, 40, 106
- Distanzfunktion, 159
 - Quantitative Distanz, 88
 - strukturelle Distanz, 85
- Effizienz, 7, 30
- Entscheidungsbaum, 52
- Entscheidungsunterstützung, 28
- Entscheidungsverhalten, 31
- Ereignis, 2
- Ereignistyp, 24
- Erfahrung, 22
- Erfahrungswissen, 22
- Fertigungsstufe, 11
- Formalziel, 10, 19
- Forschungsfragen, 187
- Genetische Algorithmen, 33
- Informationen, 25
- Informationskapital, 22
- Initialzustand, 134
- Initiator, 4, 24
- Job-Scheduling, 51
- Künstliches Neuronales Netz, 33, 61
- Klassifikationsschema, 11
- KNN, 61
- Knowledge-Engineering, 23, 25, 27
- Konvergenz, 59
- Kostenfunktionen, 37
- Kunde, 14
- Leistungsvereinbarungen, 4, 26
- Lernaufwand, 159
- Lernen
 - Ablauf Lernepisode, 135
 - Ablauf Lernschritt, 135
 - Ablauf Training, 135
 - Aufgabe, 30
 - Leistungsmaß, 30
 - maschinelles Lernverfahren, 30
 - Trainingserfahrung, 30
- Lernepisode, 44, 133
- Lernfunktion, 188
 - Effektive, 42
 - Effektivität, 167
 - Effiziente, 43
 - Effizienz, 165, 173
- Lernproblem, 34
 - des Untersuchungsgegenstandes, 30
 - maschinelles, 29
- Lernschritt, 133, 135
- Lernsystem, 100
- Lernverfahren
 - Ausgangszustand, 37
 - Cluster, 102
 - Endzustand, 37
 - Folgezustand, 37
 - maschinelles, 32
 - Q-Wert, 102
 - Reward, 102
 - Rewardfunktion, 103
- Lieferant, 14
- Marcov-Decision-Process, 36
- Material, 3, 13
- Materialfluss, 18
- MDP, 36
- Merkmale, 76
 - charakteristisch, 74
 - charakteristische, 41, 49, 53
- Multiagentensystem, 46
- Multiagentensysteme
 - Agent.Enterprise, 65

- Agile Agent Control Environment, 64
- COAGENS, 65
- MASCOPP, 66
- Planes AS, 65
- X-CITTIC, 65
- NP-vollständig, 29
- Objektknoten
 - Fertigungsobjektknoten, 13
 - Kapazitätsobjektknoten, 13
- Onlineanpassung, 192
- Operations Research, 27, 28
- Partitionierung, 49
- Partizipant, 4
- Plan, 14, 18
 - charakteristischer, 80, 81, 83
 - gültiger, 18
 - ungültiger, 18
- Planbestandsrechnung, 37
- Planung
 - deterministisch, 25
 - nicht deterministisch, 26
 - Planbestandsrechnung, 37
 - verhandlungsbasiert, 20
- Planungsaufgabe, 19
- Planungsstrategie, 3, 20, 37
 - Gegenvorschlag, 21
 - global, 3
 - Kompensation, 21
 - lokal, 3
 - Weitergabe, 21
- Planungsverfahren, 24
- Policy, 34
- PPS
 - Änderungsplanung, 19
 - Neuplanung, 19
 - Steuerung, 19
- Problemraum, 33
- Produktion, 10
 - Produktionsfaktoren, 10
 - Produktionsnetzwerk, 10
 - Produktionssystem, 11
 - Produktionsnetzwerk, 181, 187
 - Produktionsunternehmen, 9
 - Prozess
 - nicht-deterministisch, 26
 - Prozessknoten, 13
- Q-Learning, 33, 35, 52, 75
 - Q-Update, 35
 - Rewardfunktion, 100
- Q-Update, 38
- Quadrierte Distanz, 158
- Rechenleistung, 32
- Regel, 30
 - einfaches Beispiel, 40
- Regelerstellung
 - Heuristik, 27
 - maschinelle Lernverfahren, 27
 - Mathematische Analyse, 27
 - Monte-Carlo-Simulation, 29
 - Operations Research, 27
 - Reinforcement-Learning, 29
 - Simulation, 27
- Regeln, 23, 24
 - gelernt, 171, 173
 - globale, 25
 - lokale, 25
- Regelsprache, 5, 23
- Regressionsbäume, 50
- Relationale Zustandsbeschreibung, 47
- Relationale Zustandsbeschreibungen, 50
- Restriktionsverletzung, 18
- Restriktionsverletzungen, 86
- Reward, 34, 37, 38
- Rewardberechnung, 43
- Rewardfunktion, 37, 100
 - Bereitstellungskosten, 108
 - Beschaffungsstrafkosten, 119
 - Bestellpunkt, 120
 - Bestellzyklus, 120
 - Diskontfaktor, 121
 - Restriktionsverletzung, 107

- Strafkosten, 105
- Sachziel, 10, 19
- Simulation, 27
 - Simualtionsexperiment, 211
 - Simulationsstudie, 211
- SMART-Algorithmus, 62
- Steuerungsregeln, 5
- Strafkosten, 105
 - Bereitstellungsstrafkosten, 38, 104
 - Beschaffungsstrafkosten, 38, 104
 - Betriebsmittelstrafkosten, 38, 104
 - global, 189
 - lokal, 189
 - Restriktionsverletzungen, 38, 104
- Strafkostenfunktionen, 37
- Supply Chain Management, 1
- System, 11

- TD-Gammon, 51
- Training, 31, 133, 139, 178
 - Konvergenzherleitung, 144
- Trainingskonzept, 190
- Trainingsprozess, 30, 189
 - Konvergenz, 178

- Umwelt, 34
- Unsicherheit, 53

- Validierung
 - Clustering, 154
 - Dauer Trainingsprozess, 180
 - Gewichte Distanzfunktion, 159
 - Konvergenztoleranz, 160
 - Lernfunktion, 167
 - Lernverfahren, 165
 - Parametereinstellungen Clustering, 157
 - Trainingsdaten, 154
 - Trainingsprozess, 178
- Value-Funktion, 34, 48
- Verfahren, 10

- Zustand, 34, 37, 38

- Zustandsabstraktion
 - k*-means-Clustering, 54
 - Age Replacement, 63
 - Coefficient of Operational Readiness, 63