Methode zur dynamischen Anpassung von Reihenfolgeregeln mit bestärkendem Lernen

Von der Fakultät Wirtschaftswissenschaften der
Leuphana Universität Lüneburg zur Erlangung des Grades

Doktor der Ingenieurwissenschaften
– Dr.-Ing. –

genehmigte Dissertation von Thomas Voß

geboren am 12.02.1990 in Hamburg

Eingereicht am: 21.01.2022

Mündliche Verteidigung am: 06.09.2022

Betreuer und Erstgutachter:	Prof. Dr.-Ing. Jens Heger
Zweitgutachter:	Prof. Dr.-Ing. habil. Matthias Schmidt
Drittgutachter:	Prof. Dr. Burkhardt Funk

Erschienen unter dem Titel: Methode zur dynamischen Anpassung von Reihenfolgeregeln mit bestärkendem Lernen

Druckjahr: 2023

Danksagung

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am Institut für Produkt- und Prozessinnovation.

Bedanken möchte ich mich bei Prof. Dr.-Ing. Jens Heger, meinem Betreuer und Doktorvater, Prof. Dr.-Ing. habil. Matthias Schmidt, meinem Zweitgutachter und Prof. Dr. Burkhard Funk, welcher als dritter Gutachter zur Verfügung stand, für die hilfreichen Anmerkungen und konstruktiven Verbesserungen.

Bei Prof. Dr. Georgiadis und Dr. Nicolas Meier möchte ich mich bedanke, da Sie mich damals zu den Robotinos und damit zum Thema Reihenfolgeplanung gebracht haben. Weiterhin möchte ich mich bei ihnen bedanken für die Unterstützung unseres Teams bei der Teilnahme an der Logistik Liga im Rahmen des RoboCup.

Bei meinen Kolleginnen und Kollegen vom PPI möchte ich mich ebenfalls bedanken. Hasan, Danke für die Gesellschaft an dem ein oder anderem langen Abend. Natürlich geht der Dank auch an Mazhar, Marvin, Simon, Janine, Alex und allen anderen auf dem Flur, für die geführten Diskussionen und viele schöne Stunden. Weiterhin möchte ich mich bei Ingrid und Steffi für ihre Unterstützung bei allen administrativen Prozessen bedanken.

Den Studenten, die ihre Projekte und Abschlussarbeit mit mir geschrieben haben und diese Arbeit beschleunigt haben, bin ich ebenfalls zu Danke verpflichtet. Karl, Sughu, Jonas, Marie und etlichen weiteren, euch gilt mein Dank. Weiterhin danke ich Henrik und Fynn für die Stunden in der Lernfabrik und Joseph, meinem DAAD-Stipendiaten, der das Thema in seinem Praktikum forciert hat.

Meinen Eltern danke ich dafür, dass Sie mich stets ermutigt und mir über die ganze Zeit zur Seite gestanden haben.

Ein besonderer Dank gebührt Maureen. Für die moralische Unterstützung über die Zeit und vor allem für die Geduld auf der Zielgraden. Danke, dass du an mich geglaubt hast!

Und zu guter Letzt: ein Dank all den anonymen Reviewern die meine Beiträge gelesen und für gut befunden haben. Auch wenn ihr mich das ein oder andere Mal fast um den Verstand gebracht habt mit euren Fragen, auch euch ein Dank.

Thomas Voß Hamburg, den 21.01.2022

Deutsche Zusammenfassung

Die Einführung von Industrie 4.0 und der damit verbundene Wandel des Produktionsumfeldes führen zu neuen Herausforderungen, bieten auf der anderen Seite aber auch neue Möglichkeiten für Unternehmen. Ausgehend von den Herausforderungen der Produktionsplanung und Steuerung als zentrales Element der Produktherstellung, z.B. Komplexität, Dynamik und neue Organisationsformen, werden bestehenden Methoden der Reihenfolgeplanung auf ihre Tauglichkeit zur Verwendung hin geprüft. Die Analyse zeigt, dass Aspekte wie die Ableitung von Handlungen und der Transfer von Wissen in unbekannten Situationen zu den größten Herausforderungen für bestehende Verfahren zählen.

Die in der Arbeit neu entwickelte Methode zur dynamischen Auswahl und Anpassung von Reihenfolgeregeln in komplexen Fertigungssystemen mit bestärkendem Lernen greift diese Herausforderungen auf und untersucht mögliche Lösungsstrategien. Die im Rahmen der Arbeit neu entwickelte Methode wird über ein Spektrum an unterschiedlichsten Szenarien evaluiert und mit anderen Methoden verglichen. Dabei werden verschiedene Ausprägungen und Komplexitäts-Niveaus von Handlungen, der Beobachtungsraum und die Mengen an benötigten Daten detailliert analysiert. Schlussendlich zeigt sich, dass die neue Methode in der Lage ist, die Anforderungen an die Produktionsplanung- und Steuerung zu erfüllen und in bekannten wie in unbekannten Szenarien gut Leistung zu erbringen. Zusätzlich ist die Methode in der Lage menschenähnliche Leistungen zu bringen und kann in einem realen Anwendungsfall zur Unterstützung der Produktionsplanung und -Steuerung genutzt werden.

English abstract

Due to the fourth industrial revolution, the central element of product manufacturing - production planning and control – has been subject to various new challenges as well as opportunities. The literature review shows that topics like complexity, dynamics, and new organizational forms are already being focused on. Still, aspects such as derivation of actions and the transfer of knowledge in unknown situations are among the greatest challenges for existing methods.

The method developed in this thesis addresses these challenges and investigates possible solution strategies. Additionally, the new method is evaluated over a variety of scenarios and compared to other methods. Different characteristics and complexity levels of actions, the observation space, and the amounts of data required for successful training are analyzed. Finally, it is shown that the new method can fulfill the requirements of production planning and control and perform well in unknown scenarios. Additionally, the method is capable of human-like performance and can be used in a real-world scenario.

Inhaltsverzeichnis

Danksagung iii
Deutsche Zusammenfassung iv
English abstract iv
I. Abbildungen viii
II. Tabellen xi
1. Einleitung 1
1.1. Motivation 1
1.2. Zielsetzung 2
1.3. Struktur der Arbeit 3
2. Herausforderungen in der Produktionsplanung und -Steuerung 5
2.1. Einordnung im Kontext der Produktionsplanung- und Steuerung 5
2.2. Logistische Zielgrößen 7
2.3. Organisationsformen der Fertigung 8
2.4. Komplexität in der Reihenfolgeplanung 12
2.5. Dynamische Anforderungen an die Reihenfolgeplanung 14
2.6. Dezentrale Reihenfolgebildung in der Fertigung 15
2.7. Zusammenfassung der Herausforderung 16
3. Methoden der Reihenfolgeplanung 18
3.1. Exakte Verfahren zur Reihenfolgeplanung 18
3.2. Heuristiken zur Reihenfolgeplanung 20
3.2.1. Genetische Algorithmen 21
3.2.2. Einfache Prioritätsregeln 22
3.2.3. Kombinierte Prioritätsregeln für die Reihenfolgebildung 25
3.2.4. Generierte Regeln 27
3.3. Hyperheuristik zur dynamischen Reihenfolgebildung 29
3.4. Einführung ins maschinelle Lernen 31
3.4.1. Entscheidungsbäume 33
3.4.2. Künstliche Neuronale Netze 35
3.4.3. Bestärkendes Lernen 38
3.5. Simulation von Produktionssystemen 46
3.6. Zusammenfassung Stand der Technik 49
4. Handlungsbedarf 51

5. Konzept & Evaluation 55
5.1. Die flexible Werkstattfertigung mit fahrerlosen Transportfahrzeugen 56
5.1.1. Beschreibung des RoboCup Logisitk Liga Szenarios 56
5.1.2. Mathematische Modellierung 58
5.1.3. Evaluation des mathematischen Modells 60
5.1.4. Zusammenfassung 62
5.2. Zusammenspiel von Maschinen und Fahrzeugen in einer flexiblen Werkstattfertigung 62
5.2.1. Beschreibung der flexiblen Werkstattfertigung 63
5.2.2. Evaluation der Regel-Kombination 65
5.2.3. Regressionsverfahren zur Schätzung des Systemverhaltens 67
5.2.4. Zusammenfassung 69
5.3. Entwicklung der Methode zur dynamischen Anpassung von einfachen Prioritätsregeln 70
5.3.1. Erweiterung der flexiblen Werkstattfertigung 70
5.3.2. Methode zur dynamischen Anpassung einfacher Prioritätsregeln .. 71
5.3.3. Evaluation des Trainings der Methode 73
5.3.4. Evaluation der Methode im Szenario 75
5.3.5. Zusammenfassung 78
5.4. Methode zu Anpassung von kombinierten Prioritätsregeln 78
5.4.1. Szenario einer flexiblen Fließfertigung 79
5.4.2. Anpassung der ATCS-Regel mit bestärkendem Lernen 79
5.4.3. Evaluation der durchschnittlichen Verspätung 81
5.4.4. Evaluation der durchschnittlichen Durchlaufzeit 85
5.4.5. Evaluation der Anpassung mit bestärkendem Lernen 86
5.4.6. Zusammenfassung 91
5.5. Vergleich unterschiedlicher Lernverfahren zur dynamischen Anpassung 91
5.5.1. Entwicklung der drei Regressionsmodelle 93
5.5.2. Evaluation der Regressionsmethoden in mehreren Szenarien 95
5.5.3. Zusammenfassung 98
5.6. Anwendungsszenarien in der Produktion 98
5.6.1. Beschreibung des Lernfabrik Szenarios 99

5.6.2. Methode zur Bestimmung von dynamischen Losgrößen 100
5.6.3. Aufbau der Simulation 102
5.6.4. Dynamische Auswahl der Losgrößen mit bestärkendem Lernen .. 104
5.6.5. Zusammenfassung 106
5.7. Zusammenfassung Konzept und Evaluation 106
6. Fazit und Ausblick 108
Literaturverzeichnis 111

I. Abbildungen

Abbildung 1: Aufgabensicht des Aachener PPS Modell (Schuh, 2006) 6

Abbildung 2: Räumliche Struktur der Werkstattfertigung (Wiendahl, 2010) 9

Abbildung 3: Räumliche Struktur einer Reihenfertigung (Wiendahl, 2010) 10

Abbildung 4: Räumliche Struktur einer Matrixfertigung (eigene Darstellung) 12

Abbildung 5: Ansätze zur Berechnung von Maschinenbelegung und Routenauswahl (Homberger et al., 2019) 18

Abbildung 6: Drei unterschiedliche Vorgangsreihenfolgen bei bekannten Aufträgen (eigene Darstellung) 24

Abbildung 7: Genereller Aufbau zur dynamischen Auswahl und Anpassung von Reihenfolgeregeln (eigene Darstellung) 30

Abbildung 8: Basierend auf einer Eingabe macht die Funktion eine Ausgabe (Schacht und Lanquillon, 2019) 32

Abbildung 9: Eine Funktion als Ausgabe des maschinellen Lernverfahrens (Schacht und Lanquillon, 2019) 32

Abbildung 10: Einfacher Entscheidungsbaum zur Klassifikation (Matzka, 2021) 34

Abbildung 11: Ein neuronales Netz und ein einzelnes Perzeptron (eigene Darstellung) 36

Abbildung 12: Abbildung des RL-Agenten (Sutton und Barto, 2018) 39

Abbildung 13: Belohnungen über verschiedenen Zustands-Handlungspaar (eigene Darstellung) 40

Abbildung 14: Bestimmung der Einschwingphase durch die Betrachtung der Leistung über die Zeit (Gutenschwager 2017) 48

Abbildung 15: Die Simulation kann zur Evaluierung der Optimierung verwendet werden (Suhl und Mellouli, 2013) 49

Abbildung 16: Leerfahrten und blockierte Maschinen können die Fertigstellung verzögern (Poppenborg et al., 2012) 59

Abbildung 17: In ungünstigen Fällen verzögern Blockier-Zeiten die weitere Bearbeitung (Gröflin und Klinkert, 2009) 59

Abbildung 18: Für bis zu 5 Aufträge konnten in vertretbarere Rechenzeit eine optimale Lösung gefunden werden (Heger und Voss, 2017) 61

Abbildung 19: Vergleich der optimalen Lösung und FIFO (Heger und Voss, 2018)..... 61
Abbildung 20: Detaillierter Vergleich der Regel-Kombination unter Verwendung von LWT (Heger und Voß, 2019)..... 66
Abbildung 21: 5 Regelkombinationen bei unterschiedlicher Maschinen- und Fahrzeugauslastung (Heger und Voss, 2019) 68
Abbildung 22: Konzept der dynamischen Anpassung (eigene Darstellung) 72
Abbildung 23: Abhängig vom Szenario muss die Trainingsdauer des Agenten angepasst werden (Heger und Voss, 2020) 74
Abbildung 24: Die beste Regel ist abhängig vom Systemzustand und dem Produktmix (Heger und Voss, 2020)..... 75
Abbildung 25: Der Agent wählt eine passende Regel zur aktuellen Situation (Heger und Voss, 2020)..... 76
Abbildung 26: Die Leistung des Agenten aus dem ersten Training für das bekannte Szenario ist besser als die Referenzregel (Heger und Voss, 2020).... 77
Abbildung 27: Der Agent erreicht in beiden Szenarien eine vergleichbare Leistung (Heger und Voss, 2020)..... 78
Abbildung 28: Der Agent passt die k-Faktoren schrittweise an (eigene Darstellung) 80
Abbildung 29: Kleine $k2$-Werte führen bei einer hohen Auslastung zu einer besseren Leistung (Heger und Voss, 2021) 81
Abbildung 30: Die Verwendung von kleinen $k2$-Werten bei niedriger Auslastung für zu einer besseren Leistung (Heger und Voss, 2021) 82
Abbildung 31: Die Häufigkeit der besten k-Faktoren für 66 Produktmixe (Heger und Voss, 2021) 83
Abbildung 32: Detaillierte Analyse der $k2$-Faktoren bei steigender Auslastung für einen Produktmix (eigene Darstellung)..... 84
Abbildung 33:Der Agent passt den $k2$-Faktor dynamisch an die Situation an (Heger und Voss, 2021)..... 87
Abbildung 34: Der Agent bringt bessere Leistungen als die statischen k-Faktoren (Heger und Voss, 2021)..... 88
Abbildung 35: Der Agent erkennt den Produktmix-Wechsel passt den $k2$-Faktor an (Heger und Voss, 2021)..... 89
Abbildung 36: Der Agent reduziert die durchschnittliche Verspätung um bis zu 5 % (Heger und Voss, 2021) 89

Abbildung 37: Verhalten des Agenten bei unbekannter Zwischenankunftszeit (eigene Darstellung) 90

Abbildung 38: Leistung des RL-Agenten in einem unbekannten Szenario (eigene Darstellung) 90

Abbildung 39: Bei gleicher Einlastung und gleichem k-Faktor zeigen die beiden Produktmixe unterschiedliche Leistung (Heger et al., 2021)..... 95

Abbildung 40: Dynamische Anpassung des niedrigen $k2$-Faktors mit bestärkendem Lernen (Heger et al., 2021)..... 96

Abbildung 41: Vergleich der ML-Methoden zur dynamischen Parameteranpassung in einem statischen Szenario (Heger et al., 2021)..... 97

Abbildung 42: Vergleich der Methoden zur dynamischen Anpassung in einem dynamischen Szenario (Heger et al., 2021)..... 98

Abbildung 43: Montagestruktur in der Lernfabrik (Voß et al., 2021b)..... 99

Abbildung 44: Die einzelnen Elemente für die Entwicklung eines RL-Agenten (Voß et al., 2021b)..... 102

Abbildung 45: Vergleich der Leistung der MTO-Strategie in Simulation und realem System (Voß et al., 2021b)..... 104

Abbildung 46: Profit der verschiedenen Runden über die Zeit (Voß et al., 2021b)..... 105

II. Tabellen

Tabelle 1: Anzahl der möglichen Routen in Abhängigkeit von der Anzahl der Städte für das Problem des Handlungsreisenden 13

Tabelle 2: Beschreibung der Aufträge mit Auftragstyp, Prozessabfolge und Fälligkeitstermin..... 57

Tabelle 3:Parameter für die Simulationsstudie 63

Tabelle 4: durchschnittliche Durchlaufzeit bei unterschiedlichen Regelkombinationen 65

Tabelle 5: Durchschnittliche Verspätung in Abhängigkeit von Systemzustand und Regelkombination..... 67

Tabelle 6: Zusammenfassung der Simulationsparameter 71

Tabelle 7: Exemplarische Beobachtungen des Agenten im System..... 72

Tabelle 8: Beobachtungen im System mit der ATCS-Regel 80

Tabelle 9: Betrachtung der besten k-Faktoren für zwei ausgewählte Produktmixe..... 85

Tabelle 10: Vergleich der k-Werte für denselben Produktmix unter der Betrachtung unterschiedlicher Leistungsindikatoren 86

Tabelle 11: Szenario der flexiblen Fließfertigung mit 10 Maschinen..... 92

Tabelle 12: Beispiel für die Beschreibung des Systemzustandes..... 92

Tabelle 13: Parameterkonfiguration des neuronalen Netztes 93

Tabelle 14: Vorverarbeitete Beobachtungen für das neuronale Netz. 94

Tabelle 15: Parameterkonfiguration für die Entscheidungsbäume 94

Tabelle 16: Runde 139 erwirtschaftet den höchsten Profit 105

1. Einleitung

Produktionsunternehmen sind mit einer Vielzahl an technischen und strukturellen Herausforderungen konfrontiert. So ist seit 1950 bis heute ein Trend in der Zunahme an Produktvarianten und kürzer werdenden Produktlebenszyklen zu erkennen. Dieser Trend der Produktindividualisierung führt zu einer Reduktion in Produktionsvolumen pro Variante und schlussendlich zur kundenindividuellen Einzelstückfertigung (Koren und Hill, 2010).

Seit 2011 wird in Deutschland zusätzlich vom Aufkommen der vierten industriellen Revolution, auch Industrie 4.0 genannt, gesprochen. Sie schließt an die dritte industrielle Revolution, dem Aufkommen der Elektronik an. Diese beschreibt die Automatisierung von Stationen und sich wiederholenden Arbeitsschritten durch speicherprogrammierbare Steuerungen seit 1970 (Kagermann et al. 2013). Das Konzept von Industrie 4.0 greift sowohl die individuelle Fertigung wie auch die Aspekte der Digitalisierung auf, um eine effiziente Produktion und somit die bestmögliche Erfüllung des Kundenwunschs zu ermöglichen. Dabei spielen verschiedenste technologisch gestützte Aspekte wie Cyber-physische Produktionssysteme (CPPS), die horizontale und vertikale Integration der Wertschöpfungskette und die Auswertung und Reaktion auf Daten in Echtzeit eine entscheidende Rolle. (Bauernhansl et al., 2014; Mittal et al., 2018; ten Hompel et al., 2020)

1.1. Motivation

Die Produktionsplanung und –Steuerung (PPS) nimmt in diesem Zusammenhang bei der Produktion von Gütern eine zentrale Rolle in der Wertschöpfungskette ein. Nicht zuletzt ist bei neuen Konzepten wie dem Matrixproduktionssystem (Greschke et al., 2014) oder bei komplexen Systemen wie der flexiblen Werkstattfertigung die Belegung von Maschinen zum richtigen Zeitpunkt von entscheidender Bedeutung für die Leistung und so auch für die rechtzeitige Erfüllung des Kundenwunschs. (Lödding, 2016)

Ausgehend von einem kontinuierlichen Zufluss an Aufträgen und der Komplexität des Produktionssystems sowie dynamischen Einflüssen, zum Beispiel Unsicherheit durch Maschinenausfall, wurden unterschiedlichste Verfahren für die Maschinenbelegung entwickelt. Diese sind jedoch oft Szenario spezifisch oder nur situationsbedingt geeignet (Ouelhadj und Petrovic, 2009).

Ein viel diskutierter Ansatz, um Systemverhalten vorherzusagen und situationsbedingte Handlungen zu treffen ist die Verwendung von Methoden des

maschinellen Lernens. Dabei wird, ausgehend von einem vorab aufgezeichneten Datensatz, das zukünftige Verhalten vorhergesehen oder klassifiziert. In den letzten Jahren haben vor allem Ansätze unter Einsatz von bestärkendem Lernen gezeigt, dass die trainierten Agenten menschenähnliche Leistung bringen können (Silver et al., 2017).

Eine Herausforderung bei der Entwicklung und Verwendung von solchen Agenten, besonders im Kontext von effizienten und skalierbaren CPPS, stellt sich bei der Betrachtung der autonomen und dezentralen Entscheidungsfindung. Auch wenn die Teilelemente (Menschen und Maschinen) im Rahmen einer Interessengemeinschaft agieren, bestehen komplexe Wechselbeziehungen zwischen den Akteuren (Frazzon et al., 2013; Schuh et al., 2014).

Ausgehend von den neuen Rahmenbedingen wie individuellen Produktionsreihenfolgen und der Entwicklung neuer Technologien wie maschinelles Lernen stellt sich die Frage, wie eben beschriebene Konzepte konkret in der Produktion, mit dem Fokus auf die PPS, ein- und umgesetzt werden können.

1.2. Zielsetzung

Ausgehend von der zentralen Stellung der PPS gilt es, eine Methode zu entwickeln, die im Rahmen der komplexen Fertigungsstrukturen sowohl Materialflussaspekte wie auch aktuelles Systemverhalten bei der Planung der Reihenfolge sowie der schlussendlichen Belegung des Fertigungssystems berücksichtigt. Dabei gilt es die beschriebenen Rahmenbedingungen wie dynamische Änderungen in der Produktion zu erkennen und adäquat darauf zu reagieren. Die Betrachtung von verfügbaren Informationen soll zur dynamischen Anpassung von Planungsverhalten in einem komplexen und flexiblen Produktionsszenario genutzt werden.

Die Verwendung von cyber-physischen Produktionssystemen und die Verfügbarkeit von lokalen Informationen legt die detaillierte Betrachtung von dezentralen Strukturen nah. Um eine solche Methode systematisch entwickeln, die Leistung evaluieren und gegen andere Ansätze vergleichen zu können, sind mehrere Schritte notwendig: Im ersten Schritt muss eine Basiskennlinie geschaffen werden, anhand derer alle verwendeten Methoden evaluiert werden können. Danach muss die neue Methode entwickelt und gegen andere Ansätze getestet werden. Schlussendlich muss die vorgeschlagene Methode in einem realen Anwendungsfall ihre Tauglichkeit demonstrieren.

Ziel dieser Arbeit ist es, unter Betrachtung der beschriebenen Schritte, eine Methode zu entwickeln, die in Anbetracht aktueller Systeminformationen, dynamisch die Maschinenbelegung und Prozessreihenfolge an sich stetig ändernde Systemzustände anpasst.

1.3. Struktur der Arbeit

In Kapitel 1 werden Motivation sowie Zielsetzung und Struktur der Arbeit aufgezeigt. In Kapitel 2 werden die Herausforderungen der Fertigungsplanung detaillierter betrachtet. Dabei handelt es sich neben der Einordnung des Themas in den Kontext auch um die Betrachtung der gängigen Organisationsformen und dem daraus resultierenden Komplexitätsniveaus. Weiterhin werden die Unterschiede zwischen zentralen und dezentralen Ansätzen sowie statischen und dynamischen Anforderungen betrachtet. Im dritten Kapitel werden mehrere gängige Methoden der Reihenfolgeplanung vorgestellt. Neben der Beschreibung der optimalen Berechnung von Reihenfolgeplänen mit Hilfe mathematischer Modelle werden zentrale und dezentrale heuristische Ansätze aufgezeigt. Eine gängige Methode zur dezentralen Reihenfolgebildung, die in der Industrie aufgrund ihrer Geschwindigkeit und Einfachheit häufig verwendet wird, sind Prioritätsregeln. Um die in Kapitel 2 aufgezeigten Herausforderungen zu lösen, wird die Verwendung von maschinellem Lernen in Kombination mit dezentralen Heuristiken, mit besonderem Fokus auf der Verwendung von Prioritätsregeln zur Reihenfolgebildung, im Anschluss beschrieben. Weiterhin wird die ereignisdiskrete Simulation als Evaluationsmethode von komplexen Sachverhalten vorgestellt. In Kapitel 4 wird der Handlungsbedarf beschrieben, der sich aus den in Kapitel 2 gezeigten Herausforderungen sowie den in Kapitel 3 gezeigten Methoden ergibt. Weiterhin wird in Kapitel 4 beschrieben, welche Schritte benötigt werden, um eine mögliche Lösung zu erarbeiten. Kapitel 5 erörtert die in Kapitel 4 gezeigten Handlungsschritte, von der Berechnung des optimalen Plans bis zur realen Anwendung einer neuen, dezentralen und autonomen Heuristik zur dynamischen Reihenfolgeplanung. In mehreren Szenarien wird aufgezeigt, inwieweit die Berechnung einer optimalen Lösung möglich ist und wie sich die Interaktion aus Reihenfolge- und die Routenregeln auswirkt. Weiterhin wird beschrieben, wie Regressionsverfahren verwendet werden können, um das Systemverhalten bei unbekannten Situationen zu schätzen. Zusätzlich wird erläutert, wie dieses Wissen genutzt werden kann um mit Hilfe eines dynamischen Umschaltens zwischen mehreren Regeln die Systemleistung zu verbessern. Die Entwicklung der neuen Methode zur dynamischen Anpassung von einfachen und kombinierten Prioritätsregeln zur

Reihenfolgebildung mit Hilfe von RL wird vorgestellt und ausführlich evaluiert. Schlussendlich wird im Rahmen eines realen Anwendungsfalls gezeigt, wie die komplette Prozesskette von der Datenauswertung über die Erstellung der Simulation bis zum Training der neuen Heuristik und deren Bereitstellung durchgeführt werden kann. Im letzten Kapitel werden die Ergebnisse zusammengefasst und ein Ausblick auf weitere Entwicklungsmöglichkeiten gegeben.

2. Herausforderungen in der Produktionsplanung und -Steuerung

Die Grundlage vieler produzierender Unternehmen sind Mangelempfinden und Bedürfnisse des Menschen und der damit verbundene Wunsch nach Befriedigung. Bei vorhandener Kaufkraft wird aus Bedürfnissen ein Bedarf, was auch als Nachfrage bezeichnet werden kann. Der Mensch versucht dann, seinen Bedarf unter minimalem Mitteleinsatz am Markt durch Erwerb oder Gebrauch eines Gutes zu befriedigen. An dieser Stelle stehen die Unternehmen bereit, die aus verschiedenen Ressourcen Güter herstellen und diese am Markt mit einer Gewinnabsicht absetzen. (Wiswede, 1973)

2.1. Einordnung im Kontext der Produktionsplanung- und Steuerung

Zur langfristigen Sicherung der Wettbewerbsfähigkeit müssen Unternehmen einem Spannungsfeld von mehreren Faktoren, zum Beispiel Liefertermintreue und Kapitalbindungskosten sowie Lieferzeit und Prozesskosten, wirtschaftlich agieren und gefertigte Produkte absetzen. Die logistischen Leistungsindikatoren wie Lieferzeit und Liefertreue gewinnen dabei an Bedeutung, müssen aber gegen geringe Bestände und hohe Auslastung abgewogen werden. Als Ziel der Produktionslogistik kann das Streben nach hoher Lieferfähigkeit und -treue bei geringstmöglichen Logistik- und Produktionskosten definiert werden. Dabei nimmt die PPS, als Teil des komplexen Gesamtkonstrukts, einen entscheidenden Platz ein. (Wiendahl, 2010)

Eine ganzheitliche Betrachtungsweise aller relevanten, teilweise durch Abstraktion vereinfachten Zusammenhänge bietet das Aachener PPS-Modell. In Abbildung 1 ist die Aufgabensicht des Aachener PPS-Modells aufgezeigt. Diese Ansicht stellt die unterschiedlichen Aufgaben spezifiziert und detailliert dar. Typischerweise beinhaltet die Aufgabensicht drei Spalten, die Netzwerkaufgaben, die Kernaufgaben sowie die Querschnittsaufgaben.

Die Kernaufgaben, in der Abbildung auf der linken Seite, umfassen dabei die Produktionsprogrammplanung, Produktionsbedarfsplanung sowie Fremdbezugs- und Eigenfertigungsplanung und –steuerung. Die Produktionsprogrammplanung definiert die Erzeugnisse für die folgenden Planungsperioden. Die Produktionsbedarfsplanung ermittelt basierend darauf den Sekundärbedarf an Komponenten und Teilen und terminiert diese, in Abhängigkeit von der Kapazität. Fremdbezogene Artikel werden anschließend bezogen. Besonderes Augenmerk ist an dieser Stelle auf die Eigenfertigungsplanung und –steuerung zu legen, welche sowohl Losgrößenrechnung, Feinterminierung wie auch Reihenfolgeplanung beinhaltet.

Die Querschnittsaufgaben unterstützen bei der Abwicklung der Produktion. (Schuh, 2006)

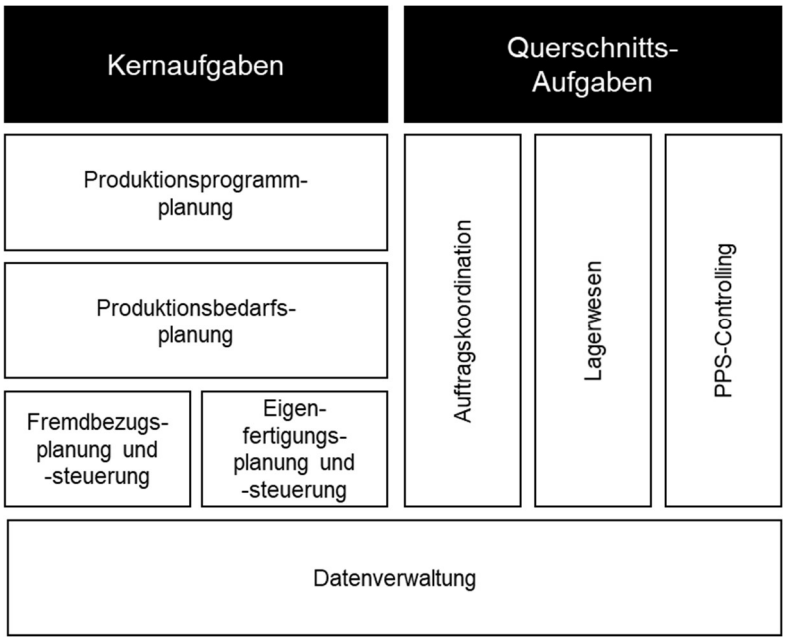


Abbildung 1: Aufgabensicht des Aachener PPS Modell (Schuh, 2006)

Von entscheidender Bedeutung für diese Arbeit ist die kurzfristige Planung und Steuerung der Fertigungsprozesse, welche ein Teil der gezeigten Eigenfertigungsplanung und –steuerung ist. Vor allem die Teilaspekte der Reihenfolgeplanung und Reihenfolgebildung werden im Kontext dieser Arbeit betrachtet. In diesem Kontext erstreckt sich der Betrachtungszeitraum von einigen Tagen bis einige Wochen. Die konkreten, dazugehörigen Aufgaben umfassen die Zuweisung, Überwachung und Anpassung von Aufträgen in der Eigenfertigung. (Hackstein, 1984)

Ausgehend von den freigegebenen Produktionsaufträgen, sowie auftragsbezogene Informationen (Reihenfolge der Arbeitsvorgänge, Plan-Fertigstellungstermin, etc.) und Kapazitätsrestriktionen ist es Aufgabe der Reihenfolgebildung zu bestimmen, welcher Auftrag in der Warteschlange eines Arbeitssystems als nächstes bearbeitet werden soll. Unter Verwendung von verschiedenen Methoden soll so die Ist-Reihenfolge der Produktionsaufträge an die Plan-Reihenfolge angepasst werden, um die definierten Leistungsindikatoren wie z.B. eine hohe Termintreue zu erreichen. (Lödding, 2016)

2.2. Logistische Zielgrößen

Zur Messung der Leistung des Unternehmens existiert ein breites Spektrum an Kennzahlen. Diese lassen sich in die Kategorien interne Leistung (z.B. Durchlaufzeit) oder externe Leistung (z.B. Lieferterminabweichung) einteilen.

Exemplarisch sind einige Beispiele aufgeführt, welche im Verlauf der Arbeit verwendet werden. Die hier beschriebenen Kennzahlen sind aus dem in der Praxis verbreiteten und zur Beschreibung von Produktionsprozessen verwendeten Trichtermodell bekannt (Wiendahl, 1997; Nyhuis und Wiendahl, 2012):

- Die Auftragszeit ist die Vorgabezeit, die für die Ausführung eines Arbeitsvorganges an einem Arbeitssystem vorgesehen ist. Der Mittelwert und die Standardabweichung der gemessenen Auftragszeit sowie die Auftragszeitverteilung können genutzt werden, um Aussagen über die Leistungsfähigkeit des Produktionssystems zu treffen.

- Die Durchlaufzeit ist festgelegt als die Spanne zwischen Auftragsfreigabe und Bearbeitungsende des Auftrags. Sie lässt sich auch beschreiben als die Spanne zwischen dem Ende des Auftrags an der aktuellen Arbeitsstation abzüglich des Zeitpunktes des Vorgänger-Auftrags. Dabei beinhaltet diese sowohl Liege-, Rüst-, Transport- wie auch Bearbeitungszeiten.

- In Fällen mit einer begrenzten Anzahl an Aufträgen kann als Kennzahl der frühste Fertigstellungszeitpunkt betrachtet werden. Dabei ist der Zeitpunkt ausschlaggebend, an dem der letzte Auftrag an der letzten Maschine fertiggestellt wird.

- Die Terminabweichung wird berechnet aus dem Ist-Bearbeitungsende und dem Soll-Bearbeitungsende und kann sowohl negative wie positive Werte annehmen. Dabei bedeutet ein positiver Wert, dass der Vorgang gegenüber der Planung verzögert wurde. Die Termintreue beschreibt das Verhältnis zwischen pünktlichen und allen Aufträgen in Prozent.

Mit Hilfe der gegebenen Definitionen können die Arbeitsvorgänge beschrieben, eingeplant und die Leistung des Produktionssystems bewertet werden. (Schuh, 2006; Nyhuis und Wiendahl, 2012; Lödding, 2016)

2.3. Organisationsformen der Fertigung

Um die Fertigungssteuerung, die kurzfristige Zuweisung von Aufträgen zu Maschinen, durchführen zu können, ist die Betrachtung der Organisationsform notwendig. Je nach Erzeugnisstruktur, Häufigkeit der Leistungswiederholung, und der Organisationsform lassen sich unterschiedliche Steuerungsverfahren für die Zuweisung von Aufträgen zu Maschinen verwenden. Jede Ausprägung von Organisationsform besitzt dabei individuelle Entscheidungs- und Freiheitsgrade.

Die Auflagehöhe des Erzeugnisses und die Wiederholbarkeit pro Jahr haben Einfluss auf räumliche Struktur und Strategie. Zusätzlich kann die Form der Transportmengen von losweisem Transport über den One-Piece-Flow Einfluss haben. Je nach Ausprägung ermöglichen manche Organisationsformen zum Beispiel Alternativen bei der Auswahl von potenziellen Maschinen. (Hackstein, 1984)

Im Folgenden werden die in der Arbeit relevanten mehrstufigen Organisationsformen und ihre Komplexität genauer betrachtet. Jeweils passend zu den einzelnen Anforderungen werden die Ausprägungen von Werkstatt-, Reihen- oder Fließfertigung bis zur Matrixfertigung erläutert. (Wiendahl, 2010; Lödding, 2016)

Für die folgenden Abbildungen gilt gleichermaßen: Die Zahlen stehen jeweils für Aufträge mit individueller Prozessreihenfolge, kenntlich gemacht durch die unterschiedlichen Linientypen. In jedem Produktionssystem sind verschiedene Bearbeitungskompetenzen wie Bohren (B), Drehen (D), Reiben (R), Schleifen (S) oder Fräsen (F) vorhanden. In der Matrixfertigung wird das Zeichen (|) zur Trennung von mehreren Kompetenzen innerhalb einer Arbeitsstation (WS) verwendet. (E) und (A) stehen jeweils für den Ein- und Ausgang der Maschinen.

Werkstattfertigung

In der Werkstattfertigung sind die Fertigungsmittel nach dem Verrichtungsprinzip angeordnet. Dabei sind gleichartige Systeme wie zum Beispiel Bohrmaschinen räumlich in einer Werkstatt zusammengefasst. Es entsteht eine organisatorische Einheit von gleichen Arbeitssystemen, welches eine Ballung der Kapazitäten und Kompetenzen zur Folge hat. In Abbildung 2 ist die räumliche Struktur der Werkstattfertigung dargestellt. Diese Konzentration von Maschinen und Operatoren an einem Ort erhöht die Flexibilität, mit der verschiedenste Aufträge bearbeitet werden können. Der Materialfluss kann als ungerichtet bezeichnet werden. Nachteile entstehen aus Transportwegen zu und von der Werkstatt. Als

weitere Nachteil werden neben der langen Durchlaufzeit auch der schlechte Überblick über die Auftragsbestände sowie die aufwendige Kapazitäts- und Reihenfolgeplanung gesehen. Dieses Konzept wird häufig im Sondermaschinenbau verwendet, wenn Aufträge komplett unterschiedliche Vorgangsreihenfolgen haben. (Wiendahl, 2010)

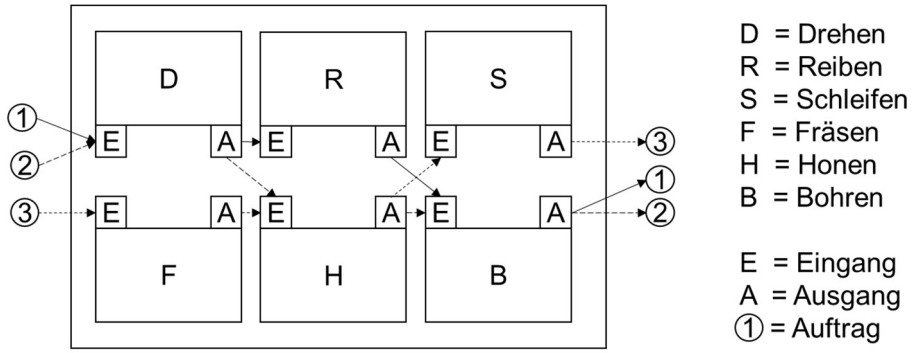


Abbildung 2: Räumliche Struktur der Werkstattfertigung (Wiendahl, 2010)

Reihen- und Fließfertigung

Im Gegensatz zur Werkstattfertigung sind bei der Reihen- und Fließfertigung die Fertigungsmittel an der Arbeitsvorgangsreihenfolge orientiert. Der Materialfluss ist hierbei gerichtet. Dabei unterliegen die Aufträge der Fließfertigung in der Regel einem getakteten Materialfluss, wohingegen die Reihenmontage in der Regel keinem Takt folgt. Bei der Reihenfertigung können einzelne Arbeitsschritte übersprungen werden. Im Gegensatz zur Werkstattfertigung kann in der, auf das Produkt abgestimmten, Fließfertigung aufgrund des hohen Automatisierungsgrades effizienter produziert werden, allerdings nur eine geringe Anzahl an Varianten. Nicht zuletzt auf Grund der geringen Pufferbestände können die daraus resultierende Durchlaufzeit der einzelnen Produkte, im Vergleich zur Werkstattfertigung, als kurz betrachtet werden. In Abbildung 3 ist die räumliche Struktur der Fließfertigung dargestellt. Durch die geringen Puffermöglichkeiten wirken sich Störungen in Form von Staus und Leerläufen aus. (Wiendahl, 2010; Lödding, 2016)

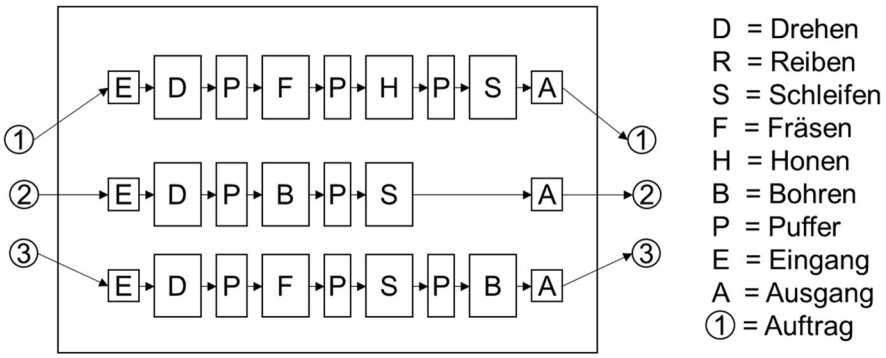


Abbildung 3: Räumliche Struktur einer Reihenfertigung (Wiendahl, 2010)

Flexible Fließ- oder Werkstattfertigung

Wenn im Kontext der Fließfertigung für jeden Vorgang nicht nur eine Maschine, sondern mehrere Maschinen zur Bearbeitung zur Auswahl stehen müssen die Aufträge entsprechend zugeordnet werden. Dabei müssen die Aufträge immer noch die komplette Fertigung durchlaufen, allerdings können die Routen je nach Situation variieren. Der Vorgang zur Maschinenzuweisung wird auch Routing genannt. Um die Aufträge zuordnen zu können wird ein gemeinsamer Puffer vor der Maschinengruppe eingerichtet, an dem die Aufträge auf ihre Maschinenzuordnung warten. Diese Anwendungsfälle, in denen eine definierte Fertigungssequenz und mehrere Maschinen für eine Operation verfügbar sind, werden flexible Fließfertigung genannt (Wang, 2005; Acker, 2011). Wenn nun zusätzlich noch mehrere Fertigungssequenzen mit alternativen Maschinenreihenfolgen möglich sind, handelt es sich um sogenannte flexible Werkstattfertigungen (Doh et al., 2013). An dieser Stelle sollte festgehalten werden, dass die Definitionen nicht absolut trennscharf sind und die Konzepte der flexiblen Werkstattfertigung und Matrixproduktion fließend ineinander übergehen.

Matrixproduktion

Anders als der standardisierte Montageprozess, wie sie in Reihen- und Fließfertigung üblich sind, müssen neuere Fertigungssysteme heute auf eine höhere Variantenvielfalt und Varianz in der Auftragszeit vorbereitet werden. Exemplarisch dafür kann die Automobilfertigung betrachtet werden (Grill-Kiefer, 2020). Die Anzahl an unterschiedlichen Ausführungsformen eines Teils, einer Baugruppe oder eines Produktes wird als Variantenvielfalt bezeichnet (Wiendahl und Lehnert, 2004). Dabei kann es beim Wechsel zwischen den einzelnen

Varianten zu Einrichtungszeiten für Maschinen, den so genannten Rüstzeiten, kommen.

Eine Möglichkeit, auf diese Herausforderung zu reagieren, ist das Konzept der Matrixproduktion. Dabei sind die verschiedenen Maschinen in der Fertigung in der Lage unterschiedliche Vorgänge zu bearbeiten und dynamisch zwischen ihren Fähigkeiten zu wechseln. So können sie Vorgänge von anderen Maschinen übernehmen, sollte das notwendig sein. Die Auswahl von bereits gerüsteten Maschinen ist eine Möglichkeit, diese nicht wertschöpfenden Zeiten zu vermeiden. Der Materialtransport wird von einem autonomen Material-Handhabung-System, wie zum Beispiel selbstfahrenden Flurförderfahrzeugen übernommen. Dabei ist die Struktur grundlegend verschieden zur Fertigungslinie. Die Maschinen sind in einer rechteckigen Matrix angeordnet und haben keinen gerichteten Materialfluss (Greschke et al., 2014; Schönemann et al., 2015). So können zum Beispiel Produkte die auf Grund der Materialversorgung nacheinander gefertigt werden mussten, mit der Matrixproduktion nebeneinander gefertigt werden und einander überholen (Fries et al., 2020). Exemplarisch sind in Abbildung 4 einzelne Arbeitsstationen (AS) mit mehreren Fertigkeiten verfügbar. Die Produkte nehmen je nach Vorgangsfolge unterschiedliche Wege durch die Fertigung. Sollte eine Station belegt sein, kann eine Alternative mit derselben Fertigkeit gewählt werden. Die Fertigkeiten Bohren, Drehen und Fräsen sind nicht notwendigerweise für alle Maschinen gleich häufig vertreten. Dabei können die einzelnen Elemente in der Matrixproduktion als eigenständige Teilnehmer dezentral Entscheidungen treffen. Die Autonomie und die Skalierbarkeit des Layouts favorisieren ein dezentrales Steuerungsverfahren (Schönemann et al., 2015; Greschke, 2016).

In ihrer Studie stellen die Autoren Hofmann et al. (2019) fest, dass die Matrixfertigung niedrigere Durchlaufzeiten und prozentual geringere Verspätung hat, als die Linienfertigung, vor allem wenn ungeplante Maschinenausfälle berücksichtigt werden. Sie weisen darauf hin, dass sowohl die Reihenfolge der Arbeitsvorgänge wie auch die Maschinenauswahl mit einer gewissen Flexibilität möglich sind. Dabei zeigt sich, dass ein Interaktionseffekt zwischen der Routenauswahl und Reihenfolgebildung besteht. Weiterhin weisen die Autoren darauf hin, dass eine höhere Flexibilität nicht zwingend mit einer Verbesserung der Leistungsindikatoren wie z.B. der Durchlaufzeit oder Terminabweichung einhergeht. (Hofmann et al., 2019)

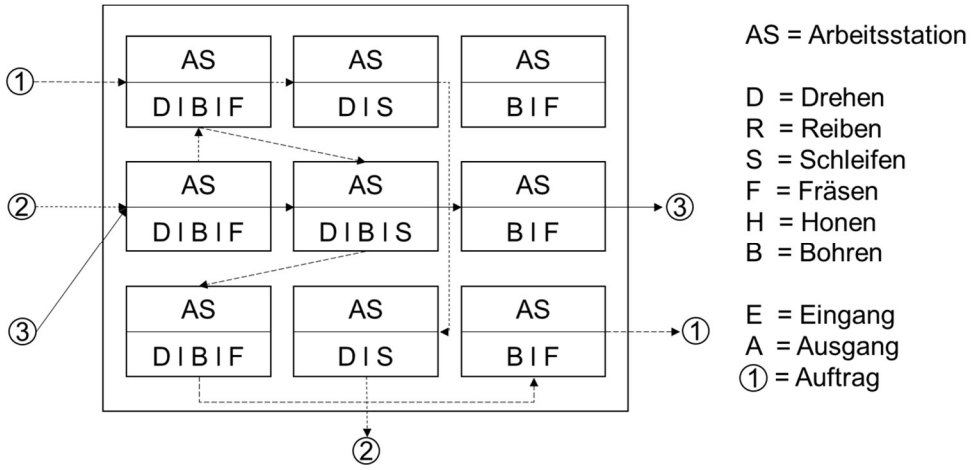


Abbildung 4: Räumliche Struktur einer Matrixfertigung (eigene Darstellung)

Stehen mehrere Maschinen für einen Vorgang zur Verfügung und bilden sich über den Verlauf der Fertigung für dieselbe Vorgangsreihenfolge unterschiedliche Routen, wird dieser Zustand in der Literatur als Routenflexibilität beschrieben. Die Routenflexibilität ermöglicht es, auch bei Störungen und dynamischen Veränderungen, Produkte ökonomisch und effizient durch das Produktionssystem zu führen. Es gilt ein Gleichgewicht aus verbesserter Maschinenauslastung und geringen Materialbeständen in der Fertigung sowie den damit verbundenen Aufwänden für qualifizierte Mitarbeiter, Werkzeuge und Vorrichtungen zu finden. Im Kontext von sich ändernden Kundenanforderungen und starken Schwankungen der Nachfrage kann diese Organisationsform strategisch sinnvoll sein. (Kumar, 2016; Sharma und Jain, 2016)

Die Schwierigkeit besteht darin, eine Zuweisung von Aufträgen zu Maschinen zu finden, die im komplexen System neben dem richtigen Zeitpunkt zur Bearbeitung auch die richtige Maschine auswählt.

2.4. Komplexität in der Reihenfolgeplanung

Die Erstellung von Plänen zur Festlegung von Arbeitsabläufen ist ein Problem, das neben der Produktion von diskreten Produkten auch in vielen weiteren Anwendungsfällen bekannt ist. Dabei müssen Restriktionen wie die Verwendung von Ressourcen eines bestimmten Typs für einzelne Vorgänge und das Ende von bestimmten Vorgängen vor Anderen beachtet werden. Im Falle der Reihenfolgeplanung und Maschinenzuweisung soll dies so geschehen, dass ein definiertes Zielkriterium, wie zum Beispiel die Verspätung, minimiert oder maximiert wird.

Das Problem des Handlungsreisenden ist an dieser Stelle bekannt und eignet sich zur Veranschaulichung der Zusammenhänge. Aus diesem Grund wird es analog zum Zuordnungsproblem von Aufträgen auf Maschinen diskutiert. Bei dem Problem des Handlungsreisenden gilt es, in einer begrenzten Anzahl an Städten die kürzeste Rundreise zu finden. Dabei muss die letzte Stadt der Reise wieder die Startstadt sein. Die Distanzen zwischen den Städten werden als Kosten betrachtet und die Kosten für die Rundreise sollen minimiert werden. Wie in Tabelle 1 gezeigt, gibt es bei 2 Städten eine mögliche Route, um in der Stadt zu enden, in der auch begonnen wurde. Bei drei Städten gibt es zwei verschiedene Routen. Generell gilt, dass es bei einer Anzahl von n Städten $(n-1)!$ Routen gibt, sofern die Auswahl nicht weiter eingegrenzt wird. Bei 10 Städten sind es bereits über 362000 mögliche Routen. Es ist also davon auszugehen, dass bei größeren Problemen auch länger gerechnet werden muss. Derselbe Sachverhalt ist bei der Zuordnung von Aufträgen auf Maschinen zu beobachten.

Tabelle 1: Anzahl der möglichen Routen in Abhängigkeit von der Anzahl der Städte für das Problem des Handlungsreisenden

Städte	Anzahl möglicher Routen
2	1
3	2
10	362880
15	87178291200

Zur Berechnung der Lösung des oben gezeigten Problems wird nicht nur Speicher, sondern auch Rechenzeit benötigt, wobei beides von der Größe des Problems abhängt. Für die Rechenzeit ist ausschlaggebend, wie viele Rechenschritte zur Lösung des Problems benötigt werden. Im Falle des Handlungsreisenden lässt sich die Anzahl der Rechenschritte zum Beispiel in Abhängigkeit der Anzahl von Städten (n) beschreiben.

Kann die Rechenzeit durch ein Polynom beschrieben werden, wird es als Kategorie $\mathcal{P}$ klassifiziert und ist vergleichsweise „einfach“ zu berechnen. Im Kontext der Reihenfolgeplanung kann hier exemplarisch ein Problem mit einer einzelnen Maschine betrachtet werden, bei dem die Minimierung der Durchlaufzeit das Ziel ist. Dabei müssen Vorrangs-Beziehungen gelten und Auftragsfreigabezeitpunkte bekannt sein. Wahlweise ist auch die Minimierung der Durchlaufzeit in einer Fließfertigung mit zwei Maschinen in polynomieller Zeit lösbar.

Sobald eine dritte Maschine hinzukommt oder eine Werkstattfertigung berechnet werden soll, ist dies allerdings nicht mehr der Fall. Sobald die Rechenzeit exponentiell wächst, können die Algorithmen nicht länger wirtschaftlich für die Lösung entsprechender Probleme eingesetzt werden, die Probleme werden dann als $\mathcal{NP}$-schwer klassifiziert. Insbesondere die Betrachtung von Fließ- und Werkstattfertigungen mit verschiedenen Restriktionen, wie zum Beispiel blockierenden Maschinen und begrenzten Pufferplätzen (Brucker et al., 2006) oder Transportrobotern (Brucker und Knust, 2012) sind in diesem Zusammenhang häufig Thema für Untersuchungen. Die Studien von Brucker und Knust zeigen, dass die Minimierung der Durchlaufzeit in einer Fließfertigung mit Transportzeiten, identischen Produktionszeiten und zwei Maschinen generell in polynomieller Zeit lösbar ist, da der Plan für die erste Maschine einfach um die Transportzeit verschoben werden kann. Werden die Produktionszeiten allerdings maschinenabhängig so ist das Problem $\mathcal{NP}$-schwer. Eine ausführliche Liste der Komplexität für unterschiedliche Szenarien ist in (Brucker et al., 2004) zu finden. Dabei ist festzustellen, dass Fließfertigungen mit mehr als drei Maschinen in fast allen Konfigurationen $\mathcal{NP}$-schwer sind. Auch wenn der Zuwachs an Rechenleistung und Verbesserungen der Algorithmen dazu führen, dass größere Instanzen gerechnet werden können (Gomes et al., 2005) ändert dies nichts daran, dass sich diese Probleme bei einer realitätsnahen Größe nicht optimal lösen lassen.

2.5. Dynamische Anforderungen an die Reihenfolgeplanung

Für vollständig definierte Probleme können in der Theorie optimale Maschinenzuweisungen und Reihenfolgepläne erstellt werden. Wie bereits in Abschnitt 2.3 beschrieben ist dies allerdings nur für kleine Szenarien möglich. Auch wenn verschiedene Methoden die Lösungsgeschwindigkeit auf Kosten der Lösungsgüte verbessern, ist dies nur bis zu einem gewissen Grad wirtschaftlich. Wenn zusätzlich noch ungeplante Änderungen auftreten oder Aufträge über die Zeit im System eintreffen, müssen Entscheidungen zeitnah getroffen werden, um den Materialfluss nicht zu behindern. Bei der Auswahl angemessener Maßnahmen für den jeweiligen Fall sind mögliche negative Wechselwirkungen in Betracht zu ziehen (Schuh et al., 2019).

Dabei wird die dynamische Reihenfolgeplanung mit unvorhergesehenen Ereignissen in ressourcen- und auftragsbezogene Probleme unterteilt. Zu ressourcenbezogenen Ereignissen gehören zum Beispiel Maschinenstörungen oder fehlendes Personal und Material. Diese Störungsarten können jeweils als Maschinenstörung interpretiert werden, da sie alle eine Verlängerung der Bearbeitungszeit zur Folge haben. Eilaufträge und Stornierungen können als

auftragsbezogene Probleme klassifiziert werden. Zur Lösung des Problems sind drei Ansätze bekannt: die Planung kann komplett reaktiv, prädiktiv-reaktiv oder robust-proaktiv durchgeführt werden. Bei der komplett reaktiven Planung wird keine Reihenfolge im voraus erstellt. Entscheidungen werden immer vor Ort und verzögerungsarm getroffen. Die Verwendung von Prioritätsregeln ist hierbei gängige Praxis. Diese einfachen Regeln können schnell Aufträge auswählen, vernachlässigen dabei aber die globale Situation in der Fertigung. Weiterhin ist es schwer bei dieser Methode die Systemleistung vorherzusagen. Die prädiktiv-reaktive Planung erstellt zentrale Ablaufpläne, die dann angepasst werden, wenn entsprechende Ereignisse eintreten. Dabei kann die dynamische Anpassung zu negativen Auswirkungen im System führen und die Leistung reduzieren. Aus diesem Grund ist bei der Erstellung von Alternativen die Robustheit, in Form der Abweichung vom Originalplan, zu beachten. Die robust-proaktive Planung fokussiert sich darauf, Ablaufpläne zu erstellen, die entsprechende Ereignisse vorsehen und diese bereits einplanen. Das zuverlässige Vorhersagen und Einplanen dieser Puffer um einen stabilen Ablaufplan zu entwickeln, stellt dabei die Herausforderung dar (Ouelhadj und Petrovic, 2009).

2.6. Dezentrale Reihenfolgebildung in der Fertigung

Im Kontext der deutschen Industrie 4.0-Offensive wird ein Paradigmenwechsel in der Industrie angestrebt. Dabei spielen Technologien wie Cyber-Physische Systeme, Internet-of-Things Anwendungen, Radio Frequency Identification und 5G eine entscheidende Rolle. Mit Hilfe der verwendeten Technologien stehen Informationen deutlich schneller und im höherem Detailgrad zu Verfügung als zuvor. Das ermöglicht es, den einzelnen Elementen in der Produktion (Aufträge, Maschinen, fahrerlosen Transportfahrzeugen und Menschen) miteinander zu kommunizieren und auf der gegebenen Datenbasis autonom Entscheidungen zu treffen (Botthof; Gabriel, 2016; Acatech Study, 2017). Im Rahmen des Sonderforschungsbereiches 637 („Selbststeuerung logistischer Prozesse - Ein Paradigmenwechsel und seine Grenzen“) wurden bereits etliche Studien zur autonomen Selbststeuerung durchgeführt. Diese belegen, dass einzelne Elemente basierend auf lokalen Beobachtungen Entscheidungen treffen können, die bessere Ergebnisse erzielen als bestehende, zentrale Verfahren (Scholz-Reiter und Freitag, 2007; Scholz-Reiter et al., 2009; Scholz-Reiter et al., 2011). Dennoch stellt sich die Frage, bis zu welchem Grad die Elemente im System autonom Entscheidungen treffen können und sollen (Gronau, 2016).

Weiterhin lassen sich Kombinationen aus zentraler und dezentraler Entscheidungsfindung zur stabilen Erstellung von Ablaufplänen diskutieren, um die Vorteile beider Ansätze zu verbinden, wobei die entstehenden Interaktionseffekte schwer nachzuvollziehen sind (Grundstein et al., 2013; Schukraft et al., 2016). In ihren Studien zeigen die Autoren Martins et al. (2020a); Martins et al. (2020b) weitere Ansätze zur autonomen Reihenfolgeplanung. Einige Ansätze sind bio-analogen oder sozialen Interaktionsverhalten nachempfunden, die meisten Methoden haben allerdings einen sehr rationalen Ursprung. Unabhängig von der verwendeten Methode bleiben die Nachvollziehbarkeit von Entscheidungen und die Interaktionseffekte zwischen zentraler und autonomer dezentraler Planung Thema für weitere Studien.

2.7. Zusammenfassung der Herausforderung

Ausgehend von der Gewinnabsicht müssen Unternehmen im Spannungsfeld der logistischen Zielgrößen wirtschaftlich produzieren. Das Dilemma zwischen Lieferzeit und niedrigen Beständen steht hierbei exemplarisch für eine Vielzahl an Herausforderungen. Um effizient produzieren zu können, müssen die vorhandenen Aufträge in der bestmöglichen Weise, abhängig von der Organisationsform der Fertigung, zu einem definierten Zeitpunkt einer einzelnen Maschine zugewiesen werden (Dangelmaier, 2009).

Es ist zu erkennen, dass der Einfluss der PPS auf die Leistung durch Kennzahlen gemessen werden kann. Die Studie von Alemão et al. (2021) zeigt, dass im Feld der Reihenfolgeplanung überwiegend die Leistungsindikatoren Durchlaufzeit und Verspätung betrachtet werden. Dabei werden, laut Aussage der Autoren und aufgrund der Komplexität, häufig Kombinationen aus bis zu 3 der folgenden Faktoren betrachtet: Flexibilität des Fertigungssystems, die Fertigstellungszeitpunkte der Produkte, das Auftreten von dynamischen Ereignissen, Wartungsaktivitäten von Maschinen, Produkttransporte innerhalb der Fertigung, Rüstzeiten bei unterschiedlichen Produktfamilien, Vorrangsbeziehungen von Prozessschritten oder schwankenden Prozesszeiten.

Zusammenfassend kann festgestellt werden, dass die bisherigen zur Planung verwendeten Methoden für kleine Szenarien, nicht aber für reale Anwendungsfälle, optimale Lösungen in polynomieller Zeit errechnen können. Für das Zuweisen von Maschinen und die Auswahl von Vorgängen zur Bearbeitung in realen Szenarien braucht es eine Lösungsmethode, die einen Kompromiss zwischen Rechenzeit und Lösungsgüte findet. Dabei sind verschiedene technologische Restriktionen sowie die Eigenschaften der

Organisationsform zu beachten. Die Menge an Produktfamilien und Rüstvorgängen hat einen entscheidenden Einfluss auf die Leistung des Systems und sollte über den zeitlichen Verlauf genau beobachtet werden, um eventuelle Anpassungen zu machen.

Aufgrund der technologischen Fortschritte stehen mehr Informationen und Kommunikationsmöglichkeiten zur Verfügung und es können, in kurzer Zeit, gute lokale und dezentrale Entscheidungen getroffen werden. Die Interaktion zwischen zentralen und dezentralen Teilnehmern ist bei der Erstellung und Anpassung der Pläne zu berücksichtigen.

Da die realen Systeme in den seltensten Fällen vollständig definiert und statisch sind, ist bei der Erstellung und Anpassung eine dynamische Komponente zu beachten. Das Auftreten von ungeplanten Ereignissen muss entsprechend durch eine Reaktion auf die Änderungen oder eine prädiktive Planung kompensiert werden. An letzter Stelle gilt es zu bedenken, dass komplizierte Fertigungssteuerungsverfahren mehr Fehlerpotenzial eröffnen und dabei zusätzlich erklärungsbedürftig sind, was ihre Akzeptanz in der Industrie verringert (Lödding, 2016).

3. Methoden der Reihenfolgeplanung

Für die Lösung von Problemen in der Produktions- und Transportplanung ist es erforderlich, diese zunächst als formales Modell aufzustellen. Ausgehend von den Prozessen und Anforderungen müssen die Zusammenhänge des Systems als Zielfunktionen und Restriktionen, ebenso wie die möglichen Entscheidungsvariablen beschrieben werden. Die realen Anforderungen sind häufig erst nach einer Reduktion der Komplexität für die Verwendung mit weiteren Verfahren geeignet. So aufbereitet können, basierend auf dem Problem, verschiedene Ansätze zur Lösung genutzt werden; Optimierung, Heuristik oder Simulation (Suhl und Mellouli, 2013).

Im Folgenden werden einzelne Ansätze und Algorithmen der Reihenfolgeplanung erläutert und in Abbildung 5 zur Übersicht schematisch eingeordnet. Die Abbildung 5 zeigt in schwarzer Schrift die im Verlauf der Arbeit betrachteten Methoden, neben den dort aufgeführten ist eine Vielzahl alternativer Lösungsansätze (exemplarisch in Grau gezeigt) vorhanden.

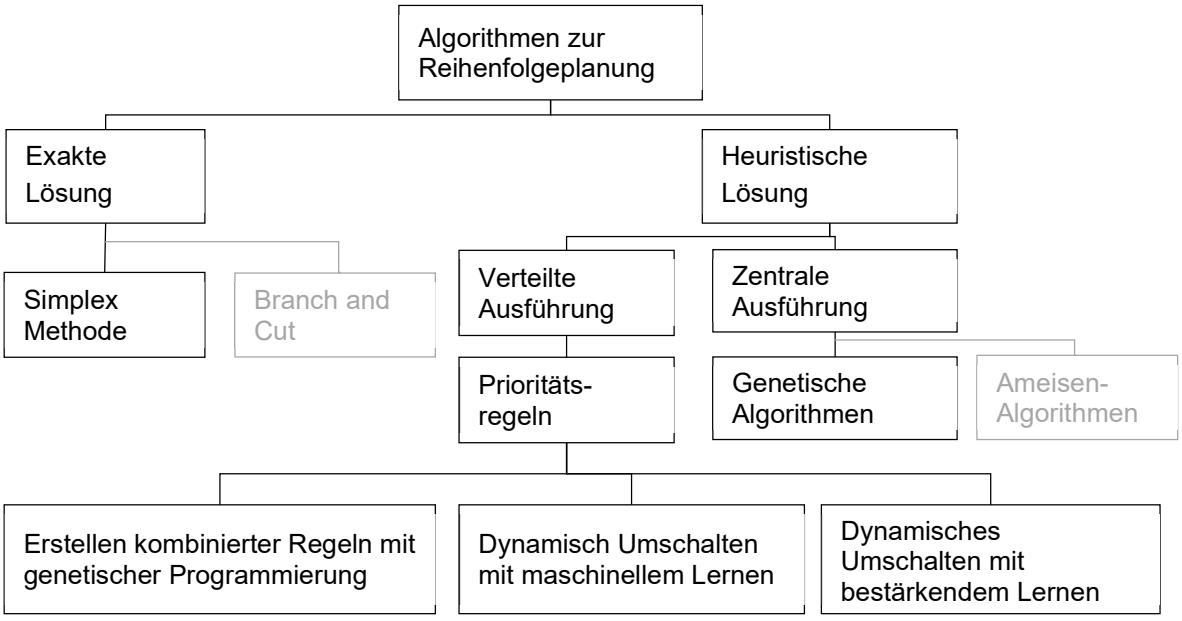


Abbildung 5: Ansätze zur Berechnung von Maschinenbelegung und Routenauswahl (Homberger et al., 2019)

3.1. Exakte Verfahren zur Reihenfolgeplanung

Wenn sich das Problem der Produktionsplanung, für das jeweilige Szenario, mathematisch beschreiben lässt, ist es theoretisch möglich für das spezifische Szenario eine exakte und optimale Lösung zu berechnen.

Wenn sowohl die Zielfunktion wie auch die Restriktionen linear sind, können Lösungsansätze der linearen Optimierung verwendet werden um die (Un-) Gleichungen zu lösen. Dies ist vor allem bei der Allokation von knappen Ressourcen unter konkurrierenden Möglichkeiten der Fall. Ein bekanntes Beispiel ist die Maximierung des Gewinns für die Produktion von zwei Produkten (zum Beispiel Bier oder Gürtelschnallen) in Abhängigkeit von Maschinenkapazität und Ressourcenverfügbarkeit.

Eine gängige Lösungsmethode ist die Verwendung des Simplex-Algorithmus (Dantzig, 1990), die zuerst nach einer gültigen Werte-Kombination der Entscheidungsvariablen im Lösungsraum sucht und dann durch iterative Verbesserungen zur optimalen Lösung gelangt. Schon der erste Schritt stellt bei großen Szenarien mit vielen möglichen Kombinationen von Entscheidungsvariablen eine Herausforderung dar. Zur Lösung dieser Probleme wurden spezielle Verfahren wie das Branch and Bound entwickelt (Land und Doig, 1960). Moderne Solver kombinieren diese Methoden, um schnell zum bestmöglichen Ergebnis zu kommen.

Da eine Teilbarkeit der Ressource (Mensch, Fahrzeug) häufig nicht möglich ist, ist die Verwendung von ganzzahligen Variablen bei solchen Systemen notwendig. Weiterhin können Entscheidungen mit Hilfe von Binär-Variablen (0/1) dargestellt werden, um logische Abhängigkeiten zu beschreiben. Diese Probleme werden gemischt-ganzzahlige Probleme (Mixed Integer Linear Problem) genannt und sind schwerer zu lösen als reine lineare Probleme. Die Schwierigkeit wird durch den Anstieg an möglichen Werte-Kombinationen erzeugt, wenn dem Problem weitere Elemente (zum Beispiel neue Jobs) hinzugefügt werden (Suhl und Mellouli, 2013).

Im Kontext der kombinierten Reihenfolgeplanung, Routenzuweisung und Fahrzeugauswahl wurden problemspezifische, mathematische Modelle entwickelt, die in der Literatur zu finden sind. Exemplarisch sind an dieser Stelle zwei Ansätze aufgeführt, welche in der Arbeit genutzt werden. Sie sind auf das Szenario einer flexiblen Werkstattfertigung mit Material-Handhabungs-System, reentranten Prozessen, parallelen Maschinen, Vorrangs-Beziehungen sowie reihenfolgeabhängigen Rüstzeiten und blockierenden Maschinen angepasst. Die Restriktion der blockierten Maschinen und die Verwendung von Transferstationen stammen aus der Robo Cup Logistik Liga. Die Restriktionen wurden im späteren Verlauf der Arbeit vernachlässigt.

Gröflin und Klinkert (2009) betrachten das Problem einer optimalen Reihenfolgeplanung und Fahrzeugauswahl in einer flexiblen Werkstattfertigung

mit blockierenden Maschinen. Das entwickelte Modell beinhaltet neben den oben genannten Restriktionen unter anderem einen Transfer zwischen zwei Vorgängen. Die Restriktion beschreibt eine Synchronisierung zwischen zwei Vorgängen, indem der erste Vorgang eine Übergabe und der Zweite eine Übernahme mit demselben Zeitwert hat.

Poppenborg et al. (2012) beschreiben die Reihenfolgeplanung und Fahrzeugauswahl in einer flexiblen Werkstattfertigung mit blockierenden Maschinen. Das Zielkriterium, das es zu minimieren gilt, ist die gewichtete Verspätung. Der Fokus liegt dabei auf der Unterscheidung zwischen Prozess- und Transportzeiten und der Betrachtung von Transferstationen, welche Material zur Verfügung stellen.

Die gezeigten Ansätze sind in der Lage, kleine und vollständig definierte Szenarien optimal zu lösen. Ausgehend von der Komplexität und Dynamik in realen Systemen (siehe auch Kapitel 2.4) führt die Verwendung von mathematischer Optimierung aber nicht immer in vertretbarer Zeit zu der optimalen Lösung. So kann es sein, dass Aufträge bereits vollständig abgearbeitet, in Bearbeitung oder eingeplant sind, wenn neue Aufträge im System eintreffen oder sich spontan Änderungen ergeben. Bei der Erstellung der jeweils neuen optimalen Lösung muss dieses Wissen entsprechend mitbetrachtet und in die Berechnung des Plans eingearbeitet werden. Der rollierende Aufruf und das Wiederholen der Berechnung in Kombination mit einer langen Rechendauer erfüllen somit nicht die Anforderungen an das zu entwickelnde System.

3.2. Heuristiken zur Reihenfolgeplanung

Es müssen also alternative Lösungen gefunden werden, die in einem kurzen Zeitraum gute und in kurzer Rechenzeit eine Annäherung an die optimalen Lösungen erstellen können: Diese Gruppe von Methoden wird Heuristiken genannt. Bei diesen Methoden kann allerdings nicht garantiert werden, dass eine im mathematischen Sinne optimale Lösung gefunden wird, auch wenn dies nach längerer Rechenzeit möglich ist. Heuristiken sind in der Regel problembasiert; Das heißt, dass die Suchmethode bekannte Eigenschaften des zu lösenden Problems nutzt, um schneller gute Lösungen zu generieren (Suhl und Mellouli, 2013). Die heuristischen Ansätze können mit Hilfe der mathematischen Modellierung evaluiert werden, da für sie dieselben Zielfunktionen, Restriktionen und Entscheidungsvariablen gelten.

3.2.1. Genetische Algorithmen

Genetische Algorithmen (GA) sind eine Methode der evolutionären Algorithmen (EA), die auf dem Prinzip natürlicher Auslese und Genetik basieren (Holland, 1984). GAs verwenden die systematische Aneinanderreihung von Entscheidungsvariablen zur Lösung des Optimierungsproblems. Zum Beispiel wird bei einem Problem wie dem Handlungsreisenden die Route als ein Chromosom repräsentiert. Die einzelnen Schritte zur Optimierung sind hierbei die Evaluation, Selektion, Rekombination, Mutation und das Ersetzen von Chromosomen. Basierend auf ihrem Gütemaß (Fitnesswert) werden bestimmte Chromosomen ausgewählt und in die nächste Population übernommen. Dort werden sie systematisch miteinander kombiniert, damit ihre Nachfolger über eine bessere Fitness verfügen. Weiterhin findet eine zufällige Mutation einzelner Individuen statt. Die schlechtesten Individuen werden zurückgelassen. Die Idee ist, durch das Rekombinieren und Mutieren von guten Lösungen zu noch besseren Lösungen zu gelangen (Koza, 1992).

Neben der Auswahl geeigneter Rekombinationsverfahren ist ein entscheidender Faktor, bei der Verwendung von GAs, die Population. Im Gegensatz zu traditionellen Suchmethoden basieren GAs auf einer Population von Lösungskandidaten. Die Populationsgröße, die normalerweise ein benutzerdefinierter Parameter ist, ist einer der wichtigen Faktoren der die Skalierbarkeit und Leistung von GAs beeinflusst. Zum Beispiel können kleine Populationsgrößen zu einer vorzeitigen Konvergenz führen und minderwertige Lösungen liefern. Andererseits führen große Populationsgrößen zu unnötigem Aufwand an wertvoller Rechenzeit (Sastry et al., 2005; Michalewicz, 2013).

Exemplarisch sind hier einzelne Ansätze aus der Literatur aufgezeigt, die eine Verwendung von GAs im Kontext der Reihenfolgeplanung zeigen:

Mattfeld und Bierwirth (2004) betrachten in ihrer Studie eine Werkstattfertigung mit dynamischen Freigabezeitpunkten. Dabei ist die Minimierung der Verspätung mit Hilfe eines GAs das Ziel. Die Chromosomen werden als Reihenfolge der Jobs auf einer Maschine codiert. Statische Reihenfolgeregeln werden um bis zu 7,5 % in allen gemessenen Zielgrößen geschlagen. Der Aufwand der Evaluation der Chromosomen wird betrachtet und durch intelligentes Eingrenzen des Lösungsraumes reduziert, ist aber dennoch nicht von der Hand zu weisen.

Mati und Xie (2008) betrachten in ihrem Beitrag eine Werkstattfertigung mit Ressourcenflexibilität. Dabei benötigt ein Prozess in einigen Fällen mehr als eine Ressource. Die Reduktion der Gesamtdurchlaufzeit in Anbetracht der

Ressourcenzuweisung und der Reihenfolgeauswahl ist das Ziel. Der Beitrag zeigt, dass der Ansatz mit bestehenden Heuristiken vergleichbar ist.

Pezzella et al. (2008) verwenden einen GA zur Reihenfolgeplanung in einer flexiblen Werkstattfertigung zur Reduktion der Gesamtdurchlaufzeit, die Zeit bis alle Aufträge vollständig bearbeitet sind. Sie identifizieren die Initialisierung mit einer guten Basislösung als einen der entscheidenden Faktoren und integrieren mehrere Strategien in ihren GA, um diese zu verbessern. Dadurch sind sie in der Lage, in deutlich kürzerer Laufzeit als vergleichbare Ansätze, gute Lösungen zu erstellen.

Ausgehend von den oben aufgeführten Ansätzen lässt sich feststellen, dass GAs in der Lage sind für ein breites Spektrum an Problemen sehr gute Lösungen zu erstellen, wenn die Zeit reicht. Die Kodierung der Chromosomen spielt dabei eine entscheidende Rolle. Es gilt bei diesen Ansätzen, dass die Rechendauer der Lösung proportional zur Menge an Individuen ist und die Evaluation große Mengen Rechenleistung erfordert.

3.2.2. Einfache Prioritätsregeln

In beiden Fällen, der Verwendung von exakten und heuristischen Methoden wird von einer zentralen Planungsinstanz ausgegangen. Dies ist jedoch auf Grund von lokalen Informationen und Dynamik nicht immer möglich. Somit sind zentrale Ansätze nur bedingt zur dynamischen Anpassung der Reihenfolge von Aufträgen geeignet.

Aus diesem Grund werden für die dezentrale und dynamische Reihenfolgebildung häufig Prioritätsregeln verwendet. Bei diesen wird, basierend auf lokal verfügbaren Informationen, allen wartenden Operationen ein skalarer Wert zugewiesen. Im Regelfall entscheidet der kleinste Wert, welcher Vorgang als Nächstes abgearbeitet wird. Im Gegensatz zu den zentralen Ansätzen wird diese Heuristik individuell auf die jeweilige Warteschlange angewendet. Auch wenn die Regeln keine optimalen Ergebnisse garantieren, sind die Regeln auf Grund ihrer Einfachheit gut nachvollziehbar und können sehr schnell Ergebnisse liefern (Gere Jr, 1966). Weiterhin sind sie für die Fähigkeit bekannt, gute Ergebnisse in dynamischen Systemen zu liefern. In der Literatur werden die Begriffe Prioritätsregel und Reihenfolgeregel häufig synonym verwendet. Um Klarheit zu schaffen, wird in dieser Arbeit von Prioritätsregeln gesprochen. Da die Verwendung im Kontext der Reihenfolgebildung, Routenauswahl und Fahrzeugzuweisung verwendet wird, handelt es sich um die generelle Methode zur Auswahl von Vorgängen.

Einfach Prioritätsregeln für die Reihenfolgebildung

Die Verwendung von Prioritätsregeln zur Reihenfolgebildung als Heuristik ist lange bekannt. Bereits 1976 schrieben Panwalkar und Iskander, dass umfangreiche Studien zur Verwendung von Prioritätsregeln in den Anwendungsfällen der Reihenfolgebildung und Zuweisung bekannt sind. In der Studie wurden mehr als 100 Regeln zusammengefasst, welche über verschiedene Szenarien hinweg robuste Lösungen erzielen. Es ist jedoch keine Regel bekannt, die in allen Situationen herausragende Ergebnisse liefert (Panwalkar und Iskander, 1977). Im Folgenden sind einige Beispiele für Prioritätsregeln für die Reihenfolgebildung erläutert. Für alle Regeln und den Rest der Arbeit gilt die folgende Notation:

m	Index der Maschine, auf der die Operation auszuwählen ist
t	Zeitpunkt, an dem der Prioritätswert berechnet wird
i	Index des Auftrags, für den der Prioritätswert berechnet wird
j	Index der Operation des Auftrags i
$\mathrm{a}_{\mathrm{i}}^{\mathrm{m}}$	Ankunftszeit des Auftrags i auf der Maschine m
$p_{i,j,m}$	Prozesszeit der Operation j des Auftrags i auf der Maschine m
d_i	Geplanter Fertigstellungszeitpunkt des Auftrags i
$s_{i,l}$	Rüstzeit, wenn Auftrag i vor Auftrag l bearbeitet wird
w_i	Gewichtungsfaktor des Auftrags i
$\mathrm{Z}_{\mathrm{i}}^{\mathrm{t}}$	Prioritätswert des Auftrags i zum Zeitpunkt t
$\mathrm{N}_{\mathrm{m}}^{\mathrm{t}}$	Set an Aufträgen, die zum Zeitpunkt t an Maschine m warten

Nachfolgend sind einige bekannte Regeln aus der Studie von Panwalkar und Iskander gezeigt:

- Die Regel „Shortest Processing Time“ (SPT) wählt den Auftrag aus der Warteschlange, welcher die kürzeste Prozesszeit an der Maschine hat. Von dieser Regel sind Variationen bekannt, die einen Gewichtungsfaktor des Jobs mit in Betracht ziehen.

$$\mathrm{Z}_{\mathrm{i}}^{\mathrm{t}} = \mathrm{p}_{\mathrm{i,j,m}}$$

- Die Regel „Earliest Due Deadline” (EDD) wählt den Auftrag, welcher den frühsten Fälligkeitstermin hat. Von dieser Regel sind Variationen bekannt, die nicht die Deadline des kompletten Auftrags, sondern der jeweiligen Operation (dann „Operation Due Date“ genannt) betrachten.

$$\mathrm{Z}_{\mathrm{i}}^{\mathrm{t}} = \mathrm{d}_{\mathrm{i}}$$

- Die Regel „First In First Out" (FIFO) ist von Supermarktkassen bekannt und wählt den Auftrag, der den frühsten Ankunftszeitpunkt in der Warteschlange der jeweiligen Maschine hat. Von dieser Regel sind verschiedene Versionen bekannt, die entweder die Ankunftszeit im System oder an der Maschine betrachten.

$$Z_i^t = a_i^m$$

- Die Regel „Similar Setups prefered" (SIMSET) wählt Aufträge mit derselben Rüstkonfiguration oder der geringsten Rüstzeit. Sie ist auch unter dem Namen „Shortest Setup Time" oder „Minimum Setup" bekannt.

$$Z_i^t = s_{i,l}$$

Alle genannten Regeln basieren auf lokal verfügbaren Informationen und weisen die Prioritätswerte basierend auf spezifischen Attributen zu. Exemplarisch ist dies in Abbildung 6 gezeigt. Bei drei unterschiedlichen Aufträgen können mit unterschiedlichen Prioritätsregeln komplett unterschiedliche Vorgangreihenfolgen erstellt werden.

Job ID	Ankunfts-zeit	Prozess-zeit	Fälligkeits-termin
3	3	4	7
2	2	2	8
1	1	3	9

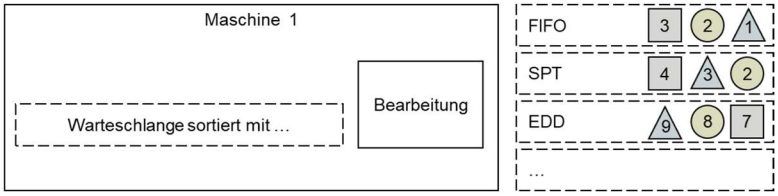


Abbildung 6: Drei unterschiedliche Vorgangsreihenfolgen bei bekannten Aufträgen (eigene Darstellung)

Ähnlich verhält es sich auch für die Auswahl von Maschinen, wenn mehrere Möglichkeiten zur Auswahl stehen.

Einfache Prioritätsregeln für die Routenauswahl

Analog zur Auswahl von Vorgängen können Maschinen für die Bearbeitung des nächsten Vorgangs ausgewählt werden. Ausgehend davon, dass mehrere Maschinen zur Verfügung stehen, welche die Operation bearbeiten können, können auch hier lokale Informationen zur Auswahl genutzt werden (Conway, 1964; Panwalkar und Iskander, 1977).

- Die „Number In Next Queue"-Regel ist ebenfalls aus dem Supermarkt bekannt und wählt grundsätzlich die Warteschlange mit der niedrigsten

Anzahl an wartenden Aufträgen aus. Die Regel ist auch als „Shortest Queue“ bekannt.

- Die „Work In Next Queue”-Regel wählt die Maschine mit der wenigsten Arbeit aus, d. h. die Summe der Bearbeitungszeiten aller wartenden Operationen. Sie ist auch unter dem Namen „Smallest Workload“ bekannt.

- Die „Least Utilized Maschine“-Regel wählt die Maschine, welche am wenigsten ausgelastet ist.

Unter gewissen Umständen können auch operationsspezifische Attribute wie die kürzeste Bearbeitungszeit erneut in Betracht gezogen werden. Dies ist allerdings nur möglich, wenn die Bearbeitung auf zwei Maschinen unterschiedlich lange dauert.

3.2.3. Kombinierte Prioritätsregeln für die Reihenfolgebildung

Ausgehend von den oben aufgezeigten Regeln kann festgehalten werden, dass die Auswahl der jeweils besten Regel schwierig ist. Aus diesem Grund wurden Regeln entwickelt, die aus mehreren Termen bestehen und so auf verschiedene Szenarien anwendbar sein sollen. Sie werden kombinierte Prioritätsregeln genannt. Die Regeln sollen nicht nur in Bezug auf das Zielkriterium stabil sein, sondern auch die Varianz reduzieren.

Holthaus-Regel

Holthaus und Rajendran (2000) beschreiben in ihrer Studie zwei Regeln die die durchschnittliche Durchlaufzeit von Aufträgen reduzieren. Eine der beiden vorgestellten Regeln, 2PT+WINQ+NPT, besteht aus drei Termen und einem Gewichtungsfaktor. Der Prioritätswert berechnet sich aus zwei Mal der Prozesszeit des Jobs i auf der Maschine m, der Summe aller anstehenden Prozesszeiten, die an der nächsten Maschine $m+1$ für den nächsten Vorgang von Job i warten und, schlussendlich, der nächsten Prozesszeit von Job i auf der Maschine $m+1$ (Formel (1)). Bei der Verwendung von dezentralen Entscheidungsansätzen für die Maschinensequenz und mehreren Alternativen ist die Abschätzung der nächsten möglichen Warteschlange allerdings mit großer Unsicherheit behaftet.

$$Z_i = 2 * p_{i,j,m} + WINQ_{m+1} + p_{i,j,\mathrm{m+1}} \qquad (1)$$

Im Vergleich zur SPT-Regel, die als Standard für Benchmarks verwendet wird, zeigt die Holthaus-Regel deutliche Verbesserungen bezüglich der Reduktion der durchschnittlichen Durchlaufzeit über verschiedenste Auslastungslevel und Szenarien. Besonders in Szenarien mit hoher Maschinenauslastung zeigt die Regel gute Ergebnisse. Dies ist unter anderem durch die Betrachtung der nachfolgenden Warteschlangen zu erklären. Die Studie zeigt, dass trotz der geringen Durchlaufzeit, die Verwendung von SPT bei hohen Auslastungen und engen Fälligkeitsterminen zu einem geringeren Anteil an verspäteten Jobs führt.

ATCS

Vepsalainen und Morton (1987) zeigen in ihrer Arbeit eine kombinierte Regel, die nicht nur die gewichtete kürzeste Prozesszeit, sondern auch das Fälligkeitsdatum durch einen Gewichtungsfaktor berücksichtigt. Die vorgeschlagene Regel erhöht die Priorität einer Operation, wenn das Fälligkeitsdatum näher rückt. In der vorgestellten Regel mit der Bezeichnung "Apparent Tardiness Cost“ (ATC) wird das Tripel aus Bearbeitungszeit, Fälligkeitstermin und Gewicht ($p_{i,}, d_i, w_i$), das mit jedem Auftrag verbunden ist, verwendet. Darüber hinaus wird ein Gewichtungsfaktor, der so genannte k_1-Faktor, zur Anpassung der Regel an das Szenario verwendet. Dieser sollte, je nach Szenario, zwischen zwei und drei gewählt werden.

Lee et al. (2002) erweiterten die Regel um die Rüstzeit, welche mit dem zweiten Term in Betracht gezogen wird (siehe Formel (2)). Hierbei wird die Rüstzeit ($s_{i,l}$) des jeweiligen Auftrags durch die mit einem Gewichtungsfaktor multiplizierte mittlere Rüstzeit geteilt. Der Gewichtungsfaktor (im Folgenden k_2-Faktor genannt), der gängiger Weise zwischen 0,01 und 1 liegt, wird hinzugefügt. In der Studie wird darauf hingewiesen, dass die beiden k-Faktoren abhängig von der Probleminstanz sind und individuell eingestellt werden müssen.

$$Z_i^t = \frac{w_i}{p_i} exp\left(-\frac{(d_i - t - p_i)^+}{k_1 \overline{p}}\right) exp\left(-\frac{s_{i,l}}{k_2 \overline{s}}\right) \qquad (2)$$

Die ATCS-Regel hat über verschiedene Szenarien hinweg gute Ergebnisse gezeigt, vor allem dann, wenn die k-Faktoren korrekt eingestellt waren. Mönch und Zimmermann (2007) bestätigen in ihrer Arbeit, dass die Regel auch in hoch komplexen Szenarien wie zum Beispiel dem Mini Fab Scenario oder dem MIMAC Testset, gute Ergebnisse liefert, sofern die k-Faktoren in Abhängigkeit von der Systemauslastung angepasst werden. Beide Szenarien sind an die Halbleiterfertigung angelehnt, haben komplexe Prozessabfolgen und Restriktionen.

3.2.4. Generierte Regeln

Wie oben gezeigt konnte die Verwendung von kombinierten Prioritätsregeln zu einer Verbesserung von Leistungsindikatoren führen. Dennoch bieten die händisch generierten Regeln, die lokal verwendet werden, noch Verbesserungspotenzial für die global gemessenen Leistungsindikatoren des Systems. Aus diesem Grund können die Ansätze der EA für die Erstellung und Programmierung von kombinierten Routen- und Reihenfolgeregeln genutzt werden. Dabei wird nicht die konkrete Lösung, sondern die neu generierte Regel als Chromosom codiert. Die Darstellung der Individuen ist also grundlegend verschieden und das Verfahren wird Genetische Programmierung (GP) genannt. Der wohl wichtigste Unterschied liegt in der variablen Länge der Chromosomen und der Verwendung von arithmetischen Operatoren als Gene. Da diese Methode eine neue Heuristik erstellt bzw. diese anpasst, wird sie auch Hyper-Heuristik genannt.

In der systematischen Betrachtung der Literatur wird gezeigt, dass die so erstellten Regeln in drei verschiedenen Varianten dargestellt werden. Eine Möglichkeit ist es, sie als lineare Kombination von Attributen mit Gewichten, wie bereits von der Holthausregel bekannt, darzustellen. Weiterhin können neuronale Netze verwendet werden. Die letzte und wohl am häufigsten verwendete Methode ist die Darstellung als Baum, der die kombinierte Prioritätsregel beschreibt (Branke et al., 2015; Branke et al., 2016; Nguyen et al., 2017). Exemplarisch werden hier einige Beispiele aufgezeigt, um die Vor- und Nachteile herauszuarbeiten.

Ho und Tay (2005) zeigten in ihrer Studie, dass die Verwendung von GP zur Generierung von kombinierten Prioritätsregeln in vielen Szenarien zur Anwendung kommen und die durchschnittliche Verspätung reduzieren können. Im Rahmen der Studie wurde die Verwendung von verschiedenen Termen, unter anderem der Prozesszeit, dem Fälligkeitstermin sowie der Anzahl und Dauer der restlichen Prozesse für die Erstellung der Regeln betrachtet. Diese wurden mit den gängigen mathematischen Operatoren kombiniert und als Baum dargestellt. In der Studie wurde die Auswahl der nächsten Maschine auf der Route über die kürzeste Wartezeit-Regel durchgeführt. Die Autoren generieren fünf verschieden Kombinationen, die Regel mit den besten Ergebnissen umfasste 14 Terme, wobei einige Zahlen und Gewichte sind. Im Vergleich konnten die neuen Regeln die Referenzregel EDD in 75 % - 85 % der getesteten Szenarien schlagen. In Bezug auf die Durchlaufzeit stellen die Autoren fest, dass FIFO weiterhin gute Ergebnisse bringt und die kombinierten Regeln in diesem Zusammenhang noch Potenzial haben. Schlussendlich konnten die Autoren, in einer weiteren Studie,

nachweisen, dass die generierten Regeln den Anteil an verspäteten Jobs im Vergleich zu den Referenzregeln reduzieren konnten (Tay und Ho, 2008).

Pickardt et al. (2013) generieren Prioritätsregeln mit GP und weisen diese dann mit einem EA an die Maschinen zu. Dabei kodiert das Chromosom die verwendete Regel passend zu jeder Maschine (Gene). Dabei ist die mit GP erstellte Prioritätsregel lediglich eine von vielen zur Auswahl. Zur Evaluation der Chromosomen wird nicht, wie sonst üblich, der Mittelwert über mehrere Replikationen eines Simulationslaufes berechnet, sondern immer dieselbe Reihenfolge an Zufallszahlen verwendet. Die Autoren stellen fest, dass dieses Verfahren, in Kombination mit einem neuen Seed für jede Generation, gute Ergebnisse liefert und die Rechenzeit reduziert.

Nguyen et al. (2013) verwenden die GP ebenfalls für die Erstellung von kombinierten Prioritätsregeln. Sie kombinieren nicht nur operationsbezogene Terme und mathematische Operatoren, sondern auch if-else-Abfragen, logische Operatoren und definierte Schwellwerte. In der ausführlichen Studie werden 540 Regeln generiert. Die Autoren führen in ihrer Auswertung die verwendeten Terme nach Häufigkeit auf. Es zeigt sich, dass bestimmte Terme, die ein Verhältnis in der Warteschlange angeben, häufiger vorkommen. Weiterhin zeigen sie auf, dass Sie bestehende kombinierte Prioritätsregeln schlagen können.

Für die Verwendung von GPs zur Erstellung komplexer kombinierter Prioritätsregeln im Kontext von flexibler Werkstattfertigung haben Zhang et al. mehrere Ansätze veröffentlicht (Zhang et al., 2018, 2019). Im Zusammenhang mit der möglichen Routenauswahl haben Sie kombinierte Regeln entwickelt, die nicht nur die Reihenfolgebildung, sondern auch die Maschinenauswahl für die nächste Operation betrachten. Sobald eine Maschine frei wird, wird die Reihenfolgeregel ausgelöst, um den Vorgang mit der höchsten Priorität als Nächstes zu bearbeiten. Dabei wird auch die Routing-Regel ausgelöst und eine mögliche Maschine für den nächsten Vorgang, unabhängig von der Bearbeitungsreihenfolge, zugewiesen. Das heißt, dass an jedem Entscheidungspunkt die entsprechende Route und Position in der Warteschlange zugewiesen und so die Routing- und Reihenfolgebildung in einer interaktiven Weise durchgeführt werden. Für die Verwendung von GP für die Erstellung von kombinierten Reihenfolge- und Routenregeln werden in der Studie beide Regeln als individuelle Bäume dargestellt. Die Terme zur Verwendung in den Chromosomen beinhalten aus demselben Grund auch maschinen-, auftrags- und system-bezogene Terme. Die Ergebnisse zeigen, dass GP in der Lage ist, nach einem etwa einstündigen Trainingsprozess, stabile Regeln zu generieren, welche bis zu 50 Terme enthalten können. Das zu Grunde liegende Trainingsszenario

beinhaltet stochastische Schwankungen auf der Prozesszeit, der Anzahl an Operationen und der Anzahl an verfügbaren Maschinen pro Operation. In Zhang et al. (2019) werden für verschiedene Szenarien jeweils individuelle Regeln generiert um entsprechend der Zielkriterien und der Auslastung gute Ergebnisse zu erzielen. In letzterem Beitrag waren die Regeln jedoch kleiner und enthielten nur bis zu 8 Terme. Die Analyse der verwendeten Terme zeigt, dass bei der Routenauswahl häufig die Menge an Arbeit in der Warteschlange relevant war. Bei der Auswahl der Terme zur Reihenfolgebildung war in der Vielzahl der Fälle die Prozesszeit enthalten.

Das Problem bei der Verwendung von GPs bei der Erstellung von Prioritätsregeln ist, dass diese für jedes Szenario neu erstellt werden müssen und kein Vorwissen im GPs gespeichert werden kann. Somit muss das Wissen erst über den Verlauf generiert oder vom Entwickler mitgegeben werden (Nguyen et al., 2017). Weiterhin kann die Erstellung von Prioritätsregeln nur durchgeführt werden, wenn die Parameter der genetischen Programmierung vorher auf das Szenario kalibriert wurden und eine gute Konfiguration der Trainingsparameter bekannt ist (Michalewicz, 2013). Weiterhin benötigen die Ansätze große Mengen Rechenleistung für die Erstellung einer einzelnen Prioritätsregel und deren Evaluation. Daher ist die Methode für statische Szenarien sehr gut, für die dynamischen Anforderungen allerdings nur bedingt geeignet.

3.3. Hyperheuristik zur dynamischen Reihenfolgebildung

Ausgehend von dem Wissen, dass es sich bei realen Produktionsszenarien um dynamische Umgebungen handelt und verschiedene Regeln für unterschiedliche Systemzustände geeignet sind, ist es naheliegend, die Regeln dynamisch an die Situation anzupassen. Hierbei handelt es sich ebenfalls um so genannte Hyper-Heuristiken, da eine Heuristik zur Reihenfolgebildung von anderen Heuristiken angepasst werden. In diesem Fall wird konkret auf den Anwendungsfall der dynamischen Anpassung von Reihenfolgeregeln eingegangen. Im Rahmen der Arbeit gilt es herauszufinden, wann welche Regel zu einer guten Leistung im System führt und diese im richtigen Moment anzuwenden. Zum Aufbau einer solchen Wissensbasis ist es erforderlich, die Situationen vorher mindestens einmal gesehen zu haben oder eine Schätzung über das Systemverhalten machen zu können. Dabei muss die Auswahl der Regel auf der Beobachtung des Systemzustandes basieren und so schnell passieren, dass es die Bearbeitung von Operationen nicht beeinträchtigt.

Exemplarisch ist das Vorgehen in Abbildung 7 gezeigt. Die einzelnen Elemente werden in den folgenden Abschnitten detaillierter beschrieben, dennoch soll das generelle Konzept an dieser Stelle kurz erläutert werden. Basierend auf einem virtuellen Abbild einer Produktion wird ein Simulationsmodell als digitaler Zwilling des Produktionssystems erstellt. Dieses Modell wird verwendet, um bestehende Prioritätsregeln zu evaluieren und die Leistung über eine breite Auswahl an Szenarien zu dokumentieren. Dieser Datensatz, der als Wissensbasis fungiert, wird anschließend genutzt, um ein Regressionsmodell zu erstellen und eine Verhaltensvorhersage für bekannte Regeln bei unbekannten Situationen durchzuführen. Das Regressionsmodell kann nach dem Training offline evaluiert werden und sollten alle Tests bestanden sein, zur Verwendung in einem Entscheidungsunterstützungssystem online bereitgestellt werden. In der online Anwendung werden dann die jeweils aktuellen Systemzustände aus dem dynamischen System durch das Entscheidungsunterstützungssystem bewertet und die passende Regel zur jeweiligen Situation ausgewählt.

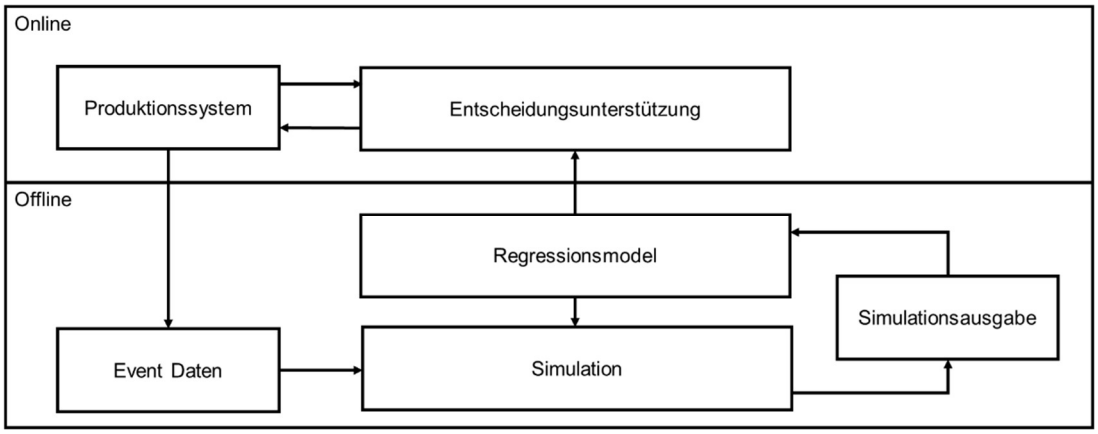


Abbildung 7: Genereller Aufbau zur dynamischen Auswahl und Anpassung von Reihenfolgeregeln (eigene Darstellung)

Ein entsprechendes wissensbasiertes und dynamisches System hat den Vorteil, dass es mindestens so gute Leistung bringt wie die beste bekannte Regel in dem Szenario. Weiterhin bietet die Anpassung das Potenzial bei Veränderungen besser zu sein als die beste statische Regel. Auf der anderen Seite bildet eine entsprechende Wissensbasis lediglich eine Untergruppe aller möglicher Szenarien ab, da nicht alle Situationen trainiert werden können. Bestimmte Regeln können über einen langen Zeitraum angewendet zu stabilen Leistungen führen, sollten aber nicht dynamisch angewendet werden. Schlussendlich kann eine aufgebaute Wissensbasis schlecht generelle Zusammenhänge wiedergeben, wenn die Situation unbekannt ist.

Ein entscheidender Vorteil der Anpassung von Reihenfolgeregeln im Gegensatz zur Blackbox Optimierung und vollständigen Erstellung konkreter Pläne ist die bessere Nachvollziehbarkeit der Anpassung von Vorhersage (Nunes und Jannach, 2017; Rehse et al., 2019).

Bei allen Ansätzen, die eine Schätzung über Leistungs- und Systemverhalten basierend auf Beobachtungen durchführen, stellt sich die Frage nach der optimalen Anzahl an Trainingspunkten. Ebenso ist die Auswahl des Beobachtungszeitraums sowie der beobachteten Systemzustände ausschlaggebend für die Leistung. Schlussendlich stellt sich die Frage nach dem Nachlernen von Wissen für unbekannte und neuen Szenarien und einer Vergleichbarkeit, Generalisierbarkeit und Nachvollziehbarkeit der verwendeten Methode (Priore et al., 2006; Priore et al., 2014; Usuga Cadavid et al., 2020).

Um diese Probleme zu lösen sind verschiedene Methoden bekannt, in den letzten Jahren haben vor allem die Methoden des maschinellen Lernens an Bedeutung gewonnen. Vor allem die Verwendung von neuronalen Netzen, Entscheidungsbäumen (Jun und Lee, 2021) und bestärkendem Lernen (Heger und Voss, 2020, 2021) haben gute Ergebnisse gezeigt.

3.4. Einführung ins maschinelle Lernen

Maschinelles Lernen, beschreibt verschiedene Funktionen und Verfahren, die Strukturen und Zusammenhänge aus Daten erfassen können. Klassischerweise betrachtete Zusammenhänge sind dabei Regressionen, Klassenzugehörigkeiten oder die Einteilung von Objekten in Gruppen. Diese Zusammenhänge lassen sich dabei in der Regel nicht ohne weiteres mathematisch formulieren. (Buxmann und Schmidt, 2019)

In dieser Arbeit werden die Methoden des maschinellen Lernens zur Schätzung von Systemzusammenhängen als Regressionsverfahren verwendet. Dabei werden, basierend auf einer Funktion Eingaben in eine Ausgabe transformiert. In Abbildung 8 ist dies exemplarisch aufgezeichnet. Der mittlere Kasten beschreibt die Funktion, die aus einer Eingabe eine Ausgabe macht. Mit diesem Vorgehen wird das Ziel verfolgt systematisch eine Vorhersage über die Leistung des Systems, zum Beispiel die Durchlaufzeit, zu tätigen. Bei der Erstellung dieser Funktionen durch den Menschen kommt es häufig zu Problemen. Diese Probleme können sowohl die Verarbeitung der Eingabeinformationen wie auch die Erstellung des Programms betreffen. (Schacht und Lanquillon, 2019)

Abbildung 8: Basierend auf einer Eingabe macht die Funktion eine Ausgabe (Schacht und Lanquillon, 2019)

Zur Automatisierung und besseren Wiederholbarkeit der Funktion wird diese in einen Algorithmus übertragen. Dabei steht im Kern des maschinellen Lernens ein Programm, das Daten ausgehend von einer Eingabe in eine Ausgabe transformiert. Im Rahmen des maschinellen Lernens wird diese Funktion aber nicht länger durch den Menschen, sondern durch den Computer erstellt (Abbildung 9). So werden, basierend auf Daten und entsprechenden Lernverfahren, die Funktionen selbstständig erarbeitet. Hierbei ist zwischen überwachtem und unüberwachtem Lernen zu unterscheiden. Die finale Funktion wird bis auf weiteres als Blackbox angenommen.

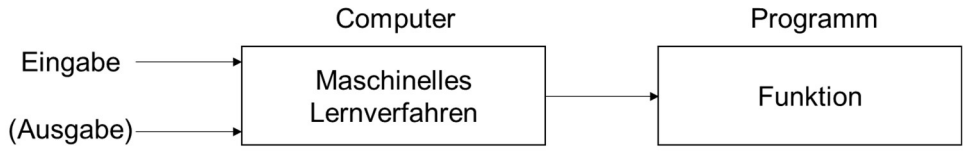


Abbildung 9: Eine Funktion als Ausgabe des maschinellen Lernverfahrens (Schacht und Lanquillon, 2019)

Beim überwachten Lernen (engl. supervised learning) soll anhand von Beispielen, das sind Eingabe-Ausgabe-Paare, eine Funktion bestimmt werden, die gegebene Eingabewerte auf bekannte Zielwerte abbildet. Typische Beispiele sind hier die Klassifikation von Objekten oder die Vorhersage von Zahlenfolgen. Unüberwachtes Lernen (engl. unsupervised learning) heißt auch Lernen aus Beobachtungen. Im Gegensatz zum überwachten Lernen sind hier jedoch keine Ausgabedaten verfügbar, die Anwendungsfälle sind also im Bereich der Clusterbildung zu sehen. Das bestärkende Lernen nimmt eine Position zwischen den beiden Ansätzen ein. (Schacht und Lanquillon, 2019)

Die Prozesskette zur Erstellung der Funktionen (auch Modellen genannt) reicht von der Erfassung von Daten über die Vorverarbeitung derselbigen bis hin zur Bereitstellung für den jeweiligen, individuellen Anwendungsfall. Dabei gilt es im iterativen Verfahren das beste Lernverfahren zur Modellierung der Funktion zu finden, die Parameter des Verfahrens zu optimieren und die bestmögliche Metrik zur Evaluation der Modelle zu finden. Schlussendlich soll das finale Modell ein Objekt in der realen Welt klassifizieren oder einen Zustand vorhersagen. (Rebala et al., 2019; Joshi, 2020)

3.4.1. Entscheidungsbäume

Entscheidungsbäume fallen in die Kategorie des überwachten Lernens, da der Beobachtungsraum abhängig von der Zielvariable zerlegt wird (Matzka, 2021). Entscheidungsbäume zerteilen, meist durch Trenngeraden, den Beobachtungsraum in Rechtecke, innerhalb derer alle Elemente gleich klassifiziert werden. Das iterative Vorgehen wird dabei so lange durchgeführt, bis ein Abbruchkriterium erreicht wird (Trabs et al., 2021). Die Herausforderung besteht darin, die Trennung des Beobachtungsraumes durch Rechtecke möglichst passend zu wählen, sodass so viel Datenpunkte wie möglich in die richtigen Klassen sortiert werden.

Am folgenden Beispiel soll verdeutlich werden, wie ein Entscheidungsbaum zu verstehen ist und warum dieser gut von Menschen interpretiert werden können. Wie in Abbildung 10 zu erkennen ist, sind die Klassen Kreis und Dreieck durch einfache Trennungen des Raumes basierend auf ihrer Position zu klassifizieren. Alle Beobachtungsobjekte, die eine Position $x_2 < 1$ haben, können eindeutig als Kreis identifiziert werden. Für die Trennung oberhalb des Wertes wird unterschieden ob die Beobachtungsobjekte als Position $x_1 > 1$ haben. Wenn dies der Fall ist, sind Sie eindeutig der Kategorie Dreieck zuzuordnen. Alle anderen Objekte lassen sich somit als Kreis definieren. Neue Objekte mit unbekannten Werten können anhand dieser Regeln entsprechend neu einsortiert werden. In einem vergleichbaren Anwendungsfall im Kontext der Reihenfolgeregeln könnten entsprechende Werte als durchschnittliche Verspätung > 200 Minuten oder die durchschnittliche Maschinenauslastung > 90 % betrachtet werden, was sehr gut für Menschen interpretierbar ist. (Rai, 2020; Matzka, 2021)

Weiterhin ist es möglich, die Klassifikation basierend auf bestimmten Merkmalen hierarchisch durchzuführen und die wichtigsten Merkmale zuerst zu überprüfen, um den Suchraum für die Klassifikation einzugrenzen und den Informationsgewinn pro Schritt zu maximieren (Runkler, 2010).

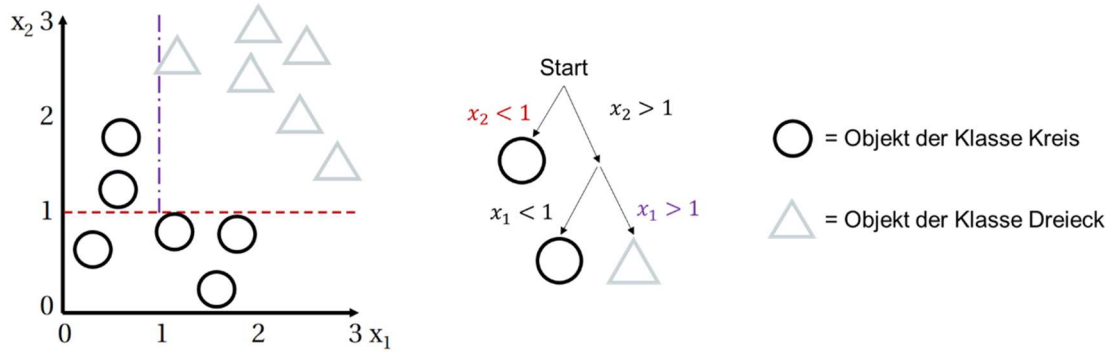


Abbildung 10: Einfacher Entscheidungsbaum zur Klassifikation (Matzka, 2021)

Im Kontext der Klassifikation sind die Entscheidungsbäume also hervorragend geeignet. Für komplexe Anwendungsfälle werden so genannte Ensemble Methoden eingesetzt, wobei mehrere Entscheidungsbäume unabhängig voneinander entwickelt und dann für die weitere Analyse zusammengefasst werden. Hierbei gilt es zu unterscheiden, ob eine Mehrheitsentscheidung (Bagging) oder eine Gewichtung (Boosting) bei der endgültigen Klassifikation verwendet wird. Im Rahmen der Regression werden Mittelwerte aus den Schätzungen der unterschiedlichen Bäume erstellt, um einen kontinuierlichen Wert zu erhalten. Die Erklärbarkeit ist bei diesen Ensmble-Methoden allerdings nicht länger in der oben gezeigten Form gegeben (Lundberg et al., 2020; Matzka, 2021).

Im Kontext der Reihenfolgebildung können Entscheidungsbäume genutzt werden um eine konkrete Reihenfolgeregeln passend zur Situation auszuwählen. Dabei werden die Regeln aus bestehenden Produktionsplänen abgeleitet und anhand von getroffenen Entscheidungen gelernt. Diese so entwickelten Entscheidungsbäume können dann anschließend für das System bereitgestellt werden und Entscheidungen treffen. Sie zeigen aber deutliche Schwächen in nicht näher beschrieben und unbekannten Szenarien (Shahzad und Mebarki, 2016).

Im Kontext eines Ein-Maschinen-Modells sowie der flexiblen Fließfertigung unter Betrachtung von Freigabezeiten ist dies ebenfalls möglich. Auch hier werden die Regeln in einem mehrschrittigen Prozess aus bestehenden Produktionsplänen (berechnet durch mathematische Optimierung) gelernt. Die Autoren zeigen, dass der beschriebene Ansatz bestehende und bekannte Regeln wie EDD und kombinierte Reihenfolgeregeln wie ATC in mehreren Szenarien mit vorab definierten und begrenzten Anzahl an Aufträgen im Bezug auf die Summe der gewichteten Verspätung schlagen kann (Jun et al., 2019; Jun und Lee, 2021). Es ergeben sich bei diesem Vorgehen ähnlich Probleme wie bei der Erstellung von kombinierten Regeln mit genetischer Programmierung (siehe Kapitel 3.2.4).

3.4.2. Künstliche Neuronale Netze

Künstliche Neuronale Netze (NN) sind informationsverarbeitende Systeme, die dem Nervensystem von Menschen und Tieren nachempfunden sind. Die Grundidee der Entwicklung von NN besteht darin, das (menschliche) Gehirn zu simulieren. Sie haben in den letzten Jahrzehnten, aufgrund ihrer Fähigkeit, das Verhalten der Eingabedaten im Trainingsdatensatz zu erlernen und die gelernten Verhaltensweisen auf zuvor unbeobachtete Daten anzuwenden, als Methode an Popularität gewonnen. (Buxmann und Schmidt, 2019)

Bereits 1958 stellt Rosenblatt in seinem Beitrag die Fragen: „Wie werden Informationen wahrgenommen?“, „Wie werden Sie gespeichert und erinnert?“ und „Wie nimmt die gespeicherte Information Einfluss auf unser Verhalten?“ (Rosenblatt, 1958). Die Verwendung des vorgeschlagenen Konzeptes des Perzeptrons, die mathematische Abbildung eines biologischen Neurons mit Gewichtung und Schwellwert als kombinierte Verarbeitungseinheit, wird seither in NN verwendet und ist durch den Aufbau verschiedener Strukturen in der Lage hoch komplexe Funktionen abzubilden (Hornik et al., 1989). Dabei werden mehrere Perzeptronen parallel zueinander in einer Schicht und mehreren Schichten hintereinander zu einem Netzwerk aufgebaut. Der Input orientiert sich an der Anzahl der Variablen die als Eingabe verwendet werden. Die Ausgabe des Netzwerkes orientiert sich an dem gewünschten Ausgabewert. Für die Schichten dazwischen gibt es, je nach Anwendungsfall, verschiedene Strukturen, die genutzt werden können. Die Schichten zwischen Ein- und Ausgabe werden „versteckt“ genannt, da Sie keinen direkten Kontakt mit der Außenwelt haben (engl. Hidden Layer). Die einzelnen Neuronen sind durch Synapsen verknüpft, welche mit einem Gewichtungsfaktor versehen sind (ω_0 und ω_1 in Abbildung 11). Innerhalb der Neuronen sind Aktivierungsfunktionen zu finden, die basierend auf der Summe aller eingehenden Signale, ein neues Signal weitergeben. Je nach Gewichtungsfaktor und verwendeter Aktivierungsfunktion auf der Synapse wird das ausgehende Signal weitergeleitet und moduliert. Während des Trainingsprozesses werden, basierend auf bekannten Daten (Trainingsdaten), die Gewichte an den Synapsen angepasst, so dass der geschätzte Wert des NN mit dem tatsächlichen Wert übereinstimmt und der Fehler minimiert wird (Kruse et al., 2011).

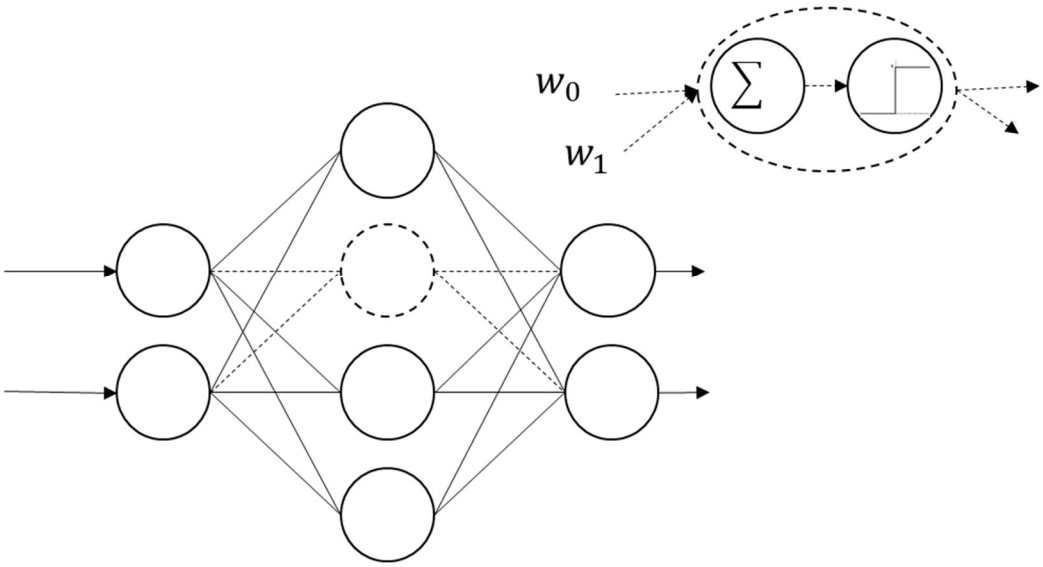


Abbildung 11: Ein neuronales Netz und ein einzelnes Perzeptron (eigene Darstellung)

Wie bereits beschrieben ist die Verwendung von NNs als Methode des maschinellen Lernens als Regressionsmethode und somit zur Verhaltensvorhersage von Leistungsindikatoren eine Möglichkeit. Im Folgenden sind exemplarisch Ansätze für die Verwendung von NNs zur dynamischen Auswahl und Anpassung von Prioritätsregeln gezeigt. Die Simulation (siehe Abschnitt 3.5) wird, in den gezeigten Fällen für die Generierung der Trainingsdaten sowie für die Evaluation der NNs im Onlinesystem verwendet.

El-Bouri und Shah (2006) betrachten die Verwendung von NNs zur Auswahl von Reihenfolgeregeln unter der getrennten Betrachtung der frühestmöglichen Fertigstellungszeitpunkte und der Minimierung der durchschnittlichen Durchlaufzeit. Dabei vergleichen sie drei verschiedene Regeln, die jeweils auf jeder Maschine individuell angewendet werden. Sie zeigen, dass die Verwendung von maschinenindividuellen Regeln bessere Ergebnisse bringt als die Verwendung einer Regel für das ganze System. Die Autoren weisen darauf hin, dass die verwendete Methode, auf Grund der verwendeten Codierung, nur auf Szenarien mit fünf Maschinen funktioniert und eine Übertragbarkeit aktuell nicht möglich ist. Die Autoren identifizieren die Frequenz und den Zeitpunkt der dynamischen Anpassung als weitere Herausforderung für den Ansatz.

Mouelhi-Chibani und Pierreval (2010) beschreiben die Verwendung von NNs zur Auswahl von einfachen Prioritätsregeln zur Reihenfolgebildung. Ausgehend von der aktuellen Systemleistung wird, unabhängig von definierten Zeitintervallen, eine Prioritätsregel ausgewählt und angewendet. Dabei werden 22 Systembeobachtungen aufgezeichnet, unter anderem die Länge der Warteschlangen, die durchschnittlichen Bearbeitungs- und Wartezeiten, sowie

die prozentualen Anteile an Aufträgen deren Bearbeitungszeit in definierten Bereichen liegen. Die Autoren stellen in ihrem Szenario mit zwei Maschinen fest, dass die Verwendung von NNs zur dynamischen Auswahl im Vergleich zu SPT und EDD bei einer konstant hohen Auslastung gute Ergebnisse bringt. Die Autoren beschreiben neben der Größe des Szenarios die Messung der Auswirkung der Reihenfolgeregel über die Zeit als besondere Herausforderung.

Da bereits aus Abschnitt 3.2.2 und 3.2.3 bekannt ist, dass einfache Prioritätsregeln von den kombinierten Prioritätsregeln über eine breite Massen an Szenarien geschlagen wurden, wird fortan nur noch die ATCS-Regel betrachtet. Die im Folgenden gezeigten Methoden sind generell als Hyper-Heuristik (Erläuterung siehe erster Abschnitt 3.3) auf ähnliche Regeln übertragbar.

Mönch et al. (2006) zeigen in ihrem Beitrag, wie die Verwendung eines „feed-forward network“ und eines Entscheidungsbaums zur Anpassung von k-Faktoren bei der ATC-Regel verwendet werden können. Basierend auf fünf Szenario-abhängigen Eingabefaktoren wird ein k-Faktor vorgeschlagen und in verschiedenen Szenarien evaluiert. Dabei stellen die Autoren fest, dass sich das Training aus einem Szenario schlecht auf ein anderes übertragen lässt. Weiterhin stellen die Autoren fest, dass die Verwendung von Entscheidungsbäumen in ihrem Szenario bessere Ergebnisse liefert als die Verwendung von NNs. In den Ergebnissen fassen die Autoren zusammen, dass die Verwendung von Methoden des maschinellen Lernens zur dynamischen Anpassung geeignet ist und besonders in den Szenarien mit reihenfolgeabhängigen Rüstzeiten noch starker Forschungsbedarf besteht.

In den Beiträgen (Heger, 2014; Heger et al., 2016) vergleichen die Autoren die Verwendung von Gaußscher Prozess Regression und NNs. Im Kontext des MiniFab Szenarios mit reihenfolgeabhängigen Rüstzeiten, zeigen Sie, dass eine Anpassung der k-Faktoren positive Auswirkung auf die Leistung des Systems hat. Eine umfangreiche Studie zur Größe des Trainingsdatensatzes wird im letzteren der Beiträge durchgeführt. Weiterhin betrachten die Autoren die Größe des Zeitfensters, in dem Änderungen im System passieren, um eine dynamische Änderung vorzunehmen. Die Autoren zeigen, dass die dynamische Anpassung in der Lage ist, Produktmix-Wechsel zu kompensieren und die Leistung des Systems über eine Anzahl verschiedener Szenarien zu verbessern. Die Autoren stellen fest, dass komplexere Szenarien noch weiteres Potenzial enthalten können und schlagen die Verwendung von unterschiedlichen Regeln an verschiedenen Maschinen vor.

3.4.3. Bestärkendes Lernen

Eine Methode die gut geeignet ist, Entscheidungen in dynamischen Systemen zu treffen, ist das bestärkende Lernen (engl. Reinforcement Learning (RL)). Im Gegensatz zu den Methoden des überwachten Lernens, bei denen eine Wissensbasis vorab aufgebaut wird, interagiert RL direkter mit dem System und lernt die richtigen Verhaltensweisen basierend auf dem beobachteten Verhalten und der erhaltenen Rückmeldung. Auch unterscheidet sich RL vom unüberwachtem maschinellen Lernen, denn es geht nicht darum Muster in unstrukturierten oder unbeschrifteten Daten zu finden. In Zusammenhang mit der Produktionssteuerung sind bereits Ansätze mit RL in der Literatur zu finden. Der entscheidende Unterschied zwischen RL und anderen Methoden des maschinellen Lernens ist der dynamische Ansatz des Lernens in Interaktion mit der Umgebung. Unterschiedliche Ausprägungen von Lernverfahren haben hierbei jeweils Vor- und Nachteile, welche in bestimmten Situationen und unter gewissen Restriktionen, zum Vorteil genutzt werden können. Die Autoren Sutton und Barto (2018) stellen klar, dass alle Agenten im Systeme ein gemeinsames Ziel erreichen möchten. Die nachfolgenden Beschreibungen sind zu großen Teilen aus der Quelle entnommen und durch zusätzliche Informationen ergänzt.

Analog zu den anderen Verfahren des maschinellen Lernens lässt sich auch dieses Verfahren in zwei Phase trennen: eine Trainings- und eine Anwendungsphase. Dabei wird innerhalb der Trainingsphase ein definierter Sachverhalt, basierend auf Beobachtungen und zufälligen Handlungen, als Modell gelernt. In der Anwendungsphase hingegen wird das trainierte Modell verwendet, um die bestmöglichen Entscheidungen basierend auf Beobachtungen zu treffen.

In Abbildung 12 ist das System für RL exemplarisch und generalisiert aufgezeigt. Auf der linken Seite ist der Agent und auf der rechten Seite die Umwelt des Agenten zu sehen. Die Interaktion zwischen den beiden wird durch die Aktion, die Belohnungsfunktion und die Veränderung der beobachteten Systemzustände beschrieben. Während des Trainings wird durch die Interaktion des Agenten mit der Umwelt eine entsprechende Veränderung des Systems bewirkt und beobachtet. Die Messung der Veränderung und die Betrachtung der Belohnungsfunktion können eine Aussage über gute und schlechte Interaktionen tätigen. Dabei hängt die Belohnung maßgeblich vom Zustand des Systems und der ausgeführten Aktion ab. Da die Belohnung eine Möglichkeit ist, kurzfristige Veränderungen zu bewerten, benötigt es noch eine langfristige Bewertung des Verhaltens. Zu diesem Zweck wird die Werte-Funktion [engl. Value-function] gebildet. Sie gibt Aussage über das langfristige Verhalten des Agenten und

bezieht mögliche nächste Schritte mit in die Berechnung ein. Innerhalb des Agenten sind mehrere Elemente zu finden: Die Strategie und eine Beobachtungskomponente. Die Strategie kann als das Verhalten des Agenten betrachtet werden, welche die Handlungen abhängig von den Beobachtungen der Zustände beschreibt. Hierbei werden mögliche Zustand-Handlungs-Paare mit einem Belohnungswert gespeichert, um sie später, in bekannten Situationen, wiederverwenden zu können. Dabei kann die Strategie neben Tabellen auch durch NN (wie in der Abbildung dargestellt) oder andere Regressionsverfahren realisiert werden. Schlussendlich beschreibt die Strategie des Agenten, ob die jeweilige Handlung bei der aktuellen Situation einen „gute“ Handlung ist oder nicht. Die gelernte Strategie ist der Kern des Systems.

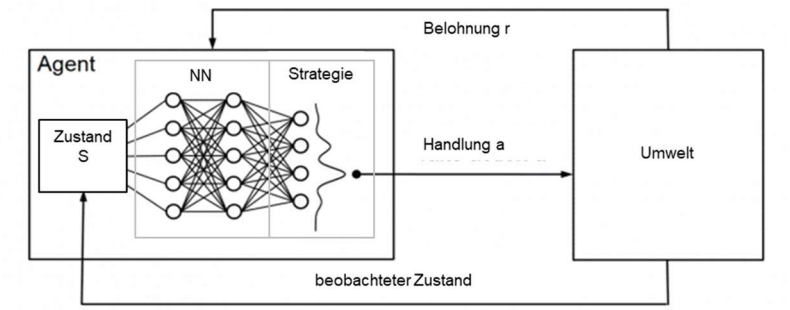


Abbildung 12: Abbildung des RL-Agenten (Sutton und Barto, 2018)

Ausgehend vom Zeitpunkt t kann der erwartete Gewinn G des Agenten als Summe aller Belohnungen R über die Zeit betrachtet werden. Wenn die Episoden allerdings unendlich lang sind, ist auch die Belohnung, unabhängig von der Strategie, unendlich und damit maximal. Aus diesem Grund werden die Belohnungen, je weiter sie vom aktuellen Zeitpunkt entfernt sind, weniger wert. Beschrieben wird dies in der Value-Function durch γ, den Discount-Faktor. Eine Belohnung, die k Schritte in der Zukunft erhalten wird, ist nur noch das γ^{k-1} fache von dem Wert, wie sie es zum aktuellen Zeitpunkt gewesen wäre, wert. Die Berechnung ist in Formel (3) gezeigt.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=1}^{\infty} \gamma^k R_{t+k+1} \qquad (3)$$

Um eine Aussage über die Güte der Handlungen machen zu können, die zu einer hohen Belohnung geführt haben, ist es möglich nach dem Ende zurückzublicken und zu prüfen, welche Zustände gut waren. Das Verfahren wird „Monte Carlo Learning“ genannt. Diese Szenarien zeichnen sich häufig durch klar beschriebene Modelle der Zustände (zum Beispiel Positionen von Steinen auf dem Feld) aus. Anders als bei Spielen wie Tic-Tac-To, Go und Schach, gibt es

Anwendungsfälle, die kein definiertes Ende haben und deren Übergänge zwischen Zuständen sich nicht präzise beschreiben lässt: so zum Beispiel Produktionssysteme (Lorenz, 2020). Aus diesem Grund müssen Verfahren verwendet werden die die Güte der Handlungen basierend auf Zeitschritten evaluieren. Diese Methoden werden „Temporal Difference Learning“ genannt (TD-Verfahren). Dabei wird geschätzt, wie der vermeintliche Gewinn aus dem nächsten Schritt aussehen kann und aus dem Fehler zwischen geschätztem und tatsächlich eingetretenem Wert gelernt (Sutton und Barto, 2018; Lorenz, 2020).

Zur Veranschaulichung sind an dieser Stelle exemplarisch vier Schritte aus dem Prozess gezeigt (Abbildung 13). Ausgehend vom Zustand S_t wird die Handlung A_t durchgeführt. Das System nimmt den Zustand S_{t+1} an und der Agent erhält die Belohnung R_{t+1}. Basierend auf dem neuen Zustand wird A_{t+1} ausgeführt, das System nimmt Zustand S_{t+2} an und der Agent erhält die Belohnung R_{t+2}, bis der Zeitpunkt T erreicht wird, der das Ende der Episode angibt. Ein Zustand S beschreibt an dieser Stelle alle für das Problem relevanten Beobachtungen, den von diesen ausgehend werden die neuen Handlungen ausgewählt.

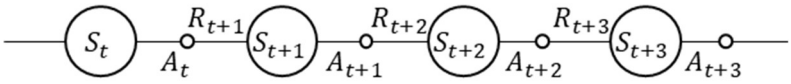


Abbildung 13: Belohnungen über verschiedenen Zustands-Handlungspaar (eigene Darstellung)

Es werden also die Belohnungen von Zustands-Handlungs-Paaren bewertet. Der Q-Wert beschreibt dabei, wie gewinnbringend die Handlung A_t im Zustand S_t ist. Die Auswahl der Handlungen im Training geschieht bis zu einem gewissen Grad zufällig. Dabei wird eine Zufallszahl generiert und mit dem ε-Wert (Epsilon) verglichen. Wenn die Zufallszahl größer ist als der Wert, wird die Handlung zufällig ausgewählt. Wenn der Zufallswert kleiner ist, wird die Handlung basierend auf dem besten Q-Wert gewählt. Abhängig von der gewählten Trainingsvariante bleibt der ε-Wert konstant oder wird immer größer. Im letzteren der Fälle wird so sichergestellt, dass zum Ende weniger neue und unbekannte Handlungen durchgeführt werden und bereits bekannte Handlungen präzisiert werden. Die Wahl des korrekten ε-Wertes und das Verhalten über die Zeit wird auch Exploration-Exploitation-Problem genannt.

Bei den bekannten TD-Verfahren kann zwischen on- und off-Policy Verfahren unterschieden werden (Lorenz, 2020). Bei dem State-Action-Reward-State-Action-Verfahren (SARSA) handelt es sich um ein On-Policy-Verfahren. Dabei wird eine Handlung gewählt, die Veränderung im System beobachtet, eine weitere Handlung gewählt und auch hier die Veränderung beobachtet. Danach

wird der Q-Wert eines Zustand-Handlung-Paares durch den alten Wert und der gewichteten Summe aus dem erzielten Gewinn und dem Unterschied zwischen den nachfolgenden Q-Werten berechnet (siehe Formel (4)). Dabei werden die Handlungen basierend auf den ε-Werten gewählt und diese Unsicherheit entsprechend mitgelernt. (Sutton und Barto, 2018)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)] \quad (4)$$

Im Gegensatz zu SARSA vergleicht das Q-Learning-Verfahren die Handlungen mit dem maximalen Gewinn. Es werden, unabhängig von der bisher gelernten Strategie, immer die Handlungen mit dem größten Gewinn als Maßstab angelegt. Aus diesem Grund wird das Verfahren auch als Off-Policy-Verfahren kategorisiert. Dies wird in der Berechnung (siehe Formel (5)) des neuen Q-Wertes deutlich.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \quad (5)$$

Zum besseren Verständnis ist das Training des Q-Learning Verfahrens einmal als Pseudo-Code beschrieben (Algorithmus 1). (Sutton und Barto, 2018)

Algorithmus 1: Pseudocode für das Agententraining

```
Initialisiere Q_{s,a}
For jede Episode do
    Initialisiere Zustand s
    For jeden Schritt in der Episode do
        Wähle eine Handlung basierend auf dem Q-Wert
        Führe die Handlung a durch und beobachte R und S_{t+1}
        Berechne den neuen Q-Wert
        S_t ← S_{t+1}
    Ende der For-Schleife
Ende der For-Schleife
```

An dieser Stelle sollten die folgenden Aspekte deutlich werden: Die Datenbasis für das Training der Ansätze zur dynamischen Anpassung von Reihenfolgeregeln mit Hilfe von NN und DT basieren auf der Auswertung der Leistung kompletter Simulationsläufe (einer Episode). RL hingegen sieht die Beobachtungen während des Simulationslaufes (je ein Schritt) und dokumentiert die durchgeführten Handlungen mit dem im Algorithmus definierten Zeitabstand. Aus diesem Grund ist es von entscheidender Bedeutung den Unterschied im Trainingsverhalten zwischen RL und den anderen Methoden des maschinellen

Lernens deutlich zu machen. Dasselbe gilt auch im Gegensatz von EA und GP zu RL. Die bioanalogen Ansätze wählen die Lösungen mit der höchsten Belohnung unabhängig von der Strategie für die jeweilige Episode und der Lebenszeit des Individuums. In den meisten Ansätzen ist die Interaktion mit der Umgebung, also für die Episode statisch, und so unabhängig vom aktuellen Zustand des Systems sowie den getroffenen Entscheidungen. In dem Fall bekommt das komplette Verhalten über die Zeit einen positiven Wert, unabhängig von einzelnen Handlungen, welche großen Einfluss hatten. Weiterhin ist an dieser Stelle darauf hinzuweisen, dass beim oben gewählten Verfahren die Handlungen des Agenten basierend auf dem maximalen Gewinn und nicht mehr mit zufälligem Anteil gewählt werden.

Die Verwendung von NN zur Vorhersage der besten Handlung in Zusammenhang mit RL hat sehr gute Ergebnisse erzielt (Silver et al., 2017). Weiterhin zeigt die Verwendung von komplexen TD-Verfahren, wie dem Actor-Critic-Verfahren, über verschiedene Anwendungsszenarien gute Ergebnisse und Handlungsstrategien. Vinyals et al. (2017) zeigen in ihrer Studie, dass die Agenten komplexe Sachverhalte erfassen und, mit genügend Training, bessere Leistungen als ein Mensch erzielen können. Die Kombination von menschlichen Wissen in Form von aufgezeichnetem Verhalten und TD-Verfahren ist in der Lage die Leistung des Agenten noch weiter zu verbessern (Vinyals et al., 2019). Die Verwendung von TD-Methoden und populationsbasiertem Training konnte in einem Spielekontext zu sinnvollem Zusammenarbeiten mehrerer Agenten führen (Jaderberg et al., 2019). Auch wenn es sich bei StrarCraft II und Quake III um Computer-Spiele handelt, können sich die Konzepte auf andere Anwendungsfälle übertragen lassen. In Zusammenhang mit der Produktion sind bereits Ansätze mit RL in der Literatur zu finden. Die Verwendung von RL für die Auswahl von einzelnen Vorgängen, basierend auf Beobachtungen, hat gute Ergebnisse gebracht.

So zeigen Gabel und Riedmiller (2008) sowie Gabel (2009) in ihren Studien die Verwendung von mehreren Agenten, die aus einer Auswahl an wartenden Vorgängen den Nächsten zur Bearbeitung auswählen. Dabei ist zu beachten, dass jeder RL-Agent für sich selbst, basierend auf seinem lokalen Wissen, entscheidet. Die Studie stellt klar, dass keine Interaktion zwischen den Agenten stattfindet. Die Autoren zeigen auf, dass RL-Agenten nach ihrem Training in der Lage sind, statische Prioritätsregeln zu schlagen. Die endgültigen Agenten sind in der Lage selbstständig Entscheidungen zu treffen und auf unvorhergesehene Ereignisse zu reagieren.

Waschneck et al. (2018a; 2018b) zeigen in ihrer Studie die Verwendung und das Zusammenspiel von mehreren DQN-Agenten in einer Werkstattfertigung. Die Herausforderung, verschiedene Strategien für verschiedene Maschinen in einem einzelnen NN zu speichern, lösen Sie durch das Erstellen maschinenindividueller Netze. Die Methode wurde bereits in vergleichbaren Szenarien verwendet (Vinyals et al., 2019). Der Handlungsraum des Agenten ist die Auswahl von Vorgängen, die vor der Maschine warten. Die Autoren zeigen, dass ein Agent innerhalb von zwei Tagen trainiert werden kann. Trotz vergleichbarerer Leistung mit FIFO kann der trainierte Agent bestehende Expertenpläne noch nicht schlagen.

Lee et al. (2019) verwenden das SARSA-Verfahren zur Auswahl von Vorgängen vor Maschinen. Das verwendete Szenario ist eine komplexe Fertigung aus dem Kontext der Halbleiterfertigung. Auch wenn RL in der Lage ist, bessere Entscheidungen als der Mensch zu treffen, beschreiben die Autoren die Grenzen ihres Ansatzes klar und deutlich: Sollten sich bestimmte Zustände im System ändern, wie zum Beispiel die Nachfrage, ist dem Agenten das Szenario unbekannt, die beobachteten Zustände können die Änderung nicht abbilden und es kann zu schlechten Handlungen kommen. Weiterhin ist die korrekte Wahl der Belohnungsfunktion immer noch abhängig vom Menschen.

Die Beiträge von Stricker und Kuhnle betrachten die Verwendung von RL-Agenten in komplexen, an die Halbleiterfertigung angelehnten, Produktionsnetzwerken. Dabei werden unterschiedliche Agenten für die Reihenfolgebildung und die Routenfindung entwickelt. Die Verwendung des Q-Learning-Verfahren mit umfangreichen Zustandsbeobachtungen und individuellem Handlungsraum wird beschrieben. Dabei ist der Handlungsraum immer die Auswahl einer konkreten Handlung des Agenten wie z.B. der Transport von A nach B, Warten oder ohne Material bewegen. Die Belohnungsfunktion ist händisch erstellt und bewertet die Fertigstellung von Aufträgen. Der erste Beitrag zeigt, dass eine generelle Strategie innerhalb von drei Tagen gelernt werden kann, die die Durchlaufzeit gegenüber der Reihenfolgeregel FIFO um etwa 5 % reduzieren kann. Ausgehend vom Handlungsraum ist davon auszugehen, dass der RL-Agent die Routenauswahl übernimmt und an dieser Stelle nur indirekt Einfluss auf die Reihenfolge hat. In den folgenden Beiträgen werden die Ergebnisse weiter ausgebaut und zeigen, wie RL-Agenten in der Lage sind im Kontext von komplexen Szenarien ganzheitliche Strategien zu erlernen, die mit der Leistung bekannter Heuristiken vergleichbar sind. Der letzte der Beiträge zeigt, dass RL-Agenten in der Lage sind, auf Änderungen im System zu reagieren und ihre Strategie anzupassen, sollte dies notwendig werden. Weiterhin werden

verschiedene Belohnungsfunktionen gegenübergestellt und in unterschiedlichen Szenarien getestet. In allen Beiträgen wird die statische Reihenfolge FIFO als Referenz verwendet und geschlagen. (Stricker et al., 2018; Kuhnle et al., 2019; 2021; Kuhnle, 2020)

In der Studie von Hofmann et al. (2020) wird ein Q-Learning-Verfahren für die Matrixproduktion mit zehn Maschinen präsentiert. Dabei werden sowohl die Reihenfolgebildung wie auch die Routenauswahl betrachtet. In der Studie wird eine Kommunikation zwischen den Agenten ermöglicht, was die Leistung allerdings nicht signifikant verbessert. Die Agenten haben die Möglichkeit zu wählen, welche Maschine und welchen Prozess sie bearbeiten möchten. Die Studie zeigt, dass die Verwendung von RL-Agenten im Vergleich zu einem regelbasierten Ansatz wie FIFO eine 5 % Verbesserung der Durchlaufzeit erreichen kann. Die Studie zeigt weiterhin, dass die Verwendung von RL-Agenten in einem dynamischen System die maximalen Durchlaufzeiten von Vorgängen reduzieren kann. Die Studie bestätigt die Ergebnisse von Stricker und Kuhnle.

Ähnlich ist auch die Studie von Park et al. (2020) angelegt. Im Rahmen einer Halbleiterfertigung wird die Verwendung eines Q-Learning-Verfahrens getestet. Für jede Maschine sind individuelle Agenten verantwortlich, welche aber ein gemeinsames NN teilen. Dabei sind in der Studie zur Zeit der Planung alle Aufträge bekannt und alle Maschinen unbelegt. Wie im Kontext der Halbleiterfertigung üblich können nur entsprechend gerüstete Maschinen Vorgänge bearbeiten. Die Studie testet ausführlich das Training des RL-Agenten in unterschiedlichen Szenarien und zeigt auf, dass die Verwendung von Q-Learning einem GA Ansatz und einer statischen Regel überlegen ist. Dabei ist anzumerken, dass es sich um ein Szenario handelt, in dem die Aufträge vorab bekannt sind.

Es lässt sich an dieser Stelle zusammenfassen, dass die bisher beschriebenen Ansätze konkrete Zuweisungen von Aufträgen zu Maschinen mit Hilfe von bestärkendem Lernen durchführen. Zur Auswahl von Vorgängen werden dabei entweder sehr kleine Szenarien betrachten oder sehr lange Trainingszeiten benötigen, um Strategien zu entwickeln. Das Problem liegt hier in der Bewertung der Auswahl einzelner Vorgänge in komplexen Systemen und die dazugehörigen Zustandsvektoren. Weiterhin ist bei einer so entwickelten Strategie nicht ersichtlich, warum bestimmte Vorgänge ausgewählt werden.

Im Gegensatz zu der Auswahl von Vorgängen an der Maschine sind Ansätze bekannt, die die Reihenfolgeregeln dynamisch anpassen. So zeigen Aydin und

Öztemel (2000) in ihrer Studie die Verwendung eines RL-Agenten zur Auswahl von Prioritätsregeln. Im Szenario mit neun Maschinen werden fünf Aufträge mit mehreren Vorgängen bearbeitet. Die Verwendung von Q-Learning führt dazu, dass basierend auf der Länge der Warteschlagen und der möglichen Verzögerung der Aufträge die Prioritätsregel zur Reihenfolgebildung zwischen drei verschiedenen Regeln ausgewählt wird. Über verschiedene Tests hinweg zeigt sich, dass der RL-Agent gute Ergebnisse im Vergleich zur statischen Anwendung erreicht.

Wang und Usher (2005) zeigen in ihrer Studie die Verwendung einer Q-Learning-Methode zur Verbesserung der Systemleistung. Zu diesem Zweck betrachten Sie eine Maschine und die Auswahl einer Prioritätsregel zur Reihenfolgebildung auf dieser Maschine. Dabei werden die möglichen Zustände im System auf zehn Wertegruppen reduziert und die Q-Werte für die verschiedenen Regeln tabellarisch dokumentiert. Die Studie zeigt, dass, je nach Zielkriterium, erfolgreich zwischen den Regeln gewechselt werden kann. Allerdings wurden reihenfolgeabhängige Rüstzeiten, Produktmixwechsel und dynamische Änderungen im System nicht beachtet.

Shiue et al. (2018) verwenden RL um Maschinen in einem Produktionssystem unterschiedliche Prioritätsregeln zuzuweisen. Die verwendete Q-Learning-Methode ist in der Lage, in einem kleinen Szenario mit fünf Maschinen verschiedene Regeln zuzuweisen. Im Vergleich mit statischen Reihenfolgeregeln kann RL die Systemleistung, in Form der durchschnittlichen Durchlaufzeit, verbessern. Die Autoren merken an, dass die Verwendung in komplexeren Szenarien noch Forschungspotenzial bietet. In dem Beitrag werden stochastische Inputs beschrieben, eine Auswertung der Schwankungen und Auswirkungen auf das Verhalten des Agenten wird allerdings nicht analysiert. Reihenfolgeabhängige Rüstzeiten werden in diesem Beitrag nicht betrachtet.

Anschließend prüfen Shiue et al. (2020) die Auswirkungen der dynamischen Anpassung von verschiedenen Reihenfolgeregeln auf unterschiedlichen Maschinen in einem komplexeren System. Die Autoren testen ihren Ansatz an zwei Anwendungsbeispielen, einem flexiblen Produktionssystem und einer Halbleiterfertigung. Dabei wird eine Wissensbasis aufgebaut und ein Q-Learning Ansatz mit RL verwendet. Die Autoren erstellen eine Simulation, um eine Trainingsumgebung für den Agenten zu realisieren. In der Simulationsstudie werden wechselnde Produktmixe für beide Szenarien betrachtet. Der Vergleich mit statischen Regeln zeigt eine deutliche Verbesserung durch den trainierten Agenten. Die Autoren stellen fest, dass die Verwendung von NN im Kontext von

dynamischer Reihenfolgebildung in Kombination mit Regelselektion noch interessante Möglichkeiten bietet.

Die Literatur zeigt, dass die Verwendung von RL zur dynamischen Anpassung von Reihenfolgeregeln als Hyper-Heuristik in kleinen Szenarien möglich ist. Dabei sind die Ansätze jedoch häufig Szenario spezifisch. Somit bleibt das Übertragen von Wissen auf unbekannte Szenarien weiterhin eine große Herausforderung. Ebenso ist die Interaktion von Reihenfolge und Routenzuweisung noch nicht hinreichend nachvollziehbar. Weiterhin ist die Kombination von mehreren Agenten, die Bewertung und die Nachvollziehbarkeit ihrer Handlungen immer noch eine Herausforderung.

Aufgrund der erfolgreichen Anwendung von populationsbasierten Trainingsmethoden in Systemen mit mehreren Agenten sind besonders die Ansätze von Jaderberg et al. (2017) im Kontext der Produktion interessant. In der veröffentlichten Studie lernte eine Menge an Agenten parallel zueinander aber unabhängig voneinander eine Strategie. Dabei wird derselbe Agent mit unterschiedlichen Szenarien konfrontiert, in denen er seine Strategie entwickeln kann. Stellt sich heraus, dass andere Agenten besser sind, wird der Agent zurückgelassen und ersetzt. Dabei wurden die Ansätze von populationsbasierter Entwicklung aufgegriffen und bis zu 30 Agenten gleichzeitig trainiert. Die individuellen Agenten agieren nach dem Training dezentral für ein gemeinsames Ziel.

In allen oben aufgezeigten Beiträgen ist eine Simulation die Grundlage für das Training des Agenten. Die Simulation wird benötigt, da ein Test in einem realen System aus verschieden Gründen nicht möglich ist. So ist zum Beispiel die Betrachtung von verzögerten Belohnungen auf Grund der ständigen Änderungen in einem realen System schwierig zu messen.

3.5. Simulation von Produktionssystemen

Im Rahmen der betrieblichen Praxis können komplexe Probleme auftreten, die nicht analytisch lösbar oder als lineares Problem abgebildet werden können. Gründe dafür können neben stochastischen Einflüssen auch komplexe Sachverhalte und Abhängigkeiten zwischen Systemkomponenten sein. An dieser Stelle eignet sich die Simulation zu Evaluation.

Die einzelnen Elemente des Systems werden hinsichtlich der Funktionalität und Leistung analysiert und als (virtuelles) Abbild erstellt. In der Simulation werden anschließend an den Nachbildungen der Elemente im System, dem abstrakten

Abbild der Realität, Experimente durchgeführt (Gutenschwager 2017). Die Methode ist besonders wertvoll, wenn Untersuchungen am realen System zu teuer, unmöglich oder mit verheerenden Folgen verbunden sind. Als Abbild des realen Systems, sollten sich die gewonnenen Erkenntnisse aus dem Simulationslauf, zurück übertragen lassen. (Suhl und Mellouli, 2013)

Mit der Verbreitung leistungsfähiger Computer hat die, sonst aus dem Flugzeugbau und Fahrzeugbau bekannte Technik, Einzug in den Bereich Operatins Research gefunden (Domschke et al., 2015). Die neuen Technologien ermöglichen es, nicht zuletzt durch weitergehende Digitalisierung der Maschinen in komplexen Produktionsnetzwerken, Verbesserung von Strukturen, Prozesse und Ressourcen zu unterstützen. Ein digitaler Zwilling wird als virtuelles Abbild des realen Produktionssystems angelegt.

Der Vorteil eines virtuellen Abbildes eines komplexen Systems ist die Möglichkeit, die Zeit zwischen Handlungen und so auch die Interaktionen der Teilnehmer im System verkürzt oder verlängert abbilden zu können. Das ermöglicht einen schnellen Erkenntnisgewinn über das zeitliche Verhalten bei unterschiedlichen Situationen des realen Systems. In vielen Fällen können die Ergebnisse und Verhalten von mehrere Wochen Produktionszeit in Sekunden berechnet ausgewertet werden. Ein weiterer Vorteil ist die Möglichkeit das Verhalten unter stochastischen Schwankungen zu betrachten. Die Verwendung zufallsabhängiger Werter kann bei wiederholter Durchführung von Simulationsläufe zwei unterschiedliche Strategien bzgl. ihrer Leistung mit relativer Sicherheit beschreiben und so eine Aussage über die Qualität der verwendeten Strategie machen. Durch die Möglichkeit die Simulationsmodelle mit unterschiedlichen Parameterkonfigurationen oder komplett unterschiedlichen Systemkonfigurationen zu testen, können so schrittweise Wissen über das System generiert werden und neue Ideen getestet werden. (Gutenschwager 2017)

Wenn im Rahmen der Simulation klar trennbare Elemente und Ereignisse verwendet werden, handelt es sich diskrete Simulation. Entscheidend ist hierbei, dass sich die Modellzustände nur zu gewissen, diskreten Zeitpunkten ändern. Im Produktionskontext ist es aus logistischer Sicht uninteressant inwieweit sich das Werkstück während der Bearbeitung verändert. Die relevanten Zeitpunkte sind der Beginn und das Ende der Bearbeitung. Der häufigste Anwendungsfall für diese Methode ist der Vergleich von Systemkonfiguration oder Handlungsstrategien. Exemplarisch kann im Kontext eines Produktionssystems die Verwendung unterschiedlicher Reihenfolgeregeln unter stochastischen Einflüssen getestet werden. Die Vergleichsmetrik in diesem Fall ist die

durchschnittliche Verspätung oder die durchschnittliche Durchlaufzeit der Aufträge über den Simulationslauf. (Eley, 2012)

Bei der Verwendung von Simulation zur Evaluation nicht terminierender Systeme ist neben der Wahl der angemessenen Leistungsindikatoren auch die Art der Messung des jeweiligen Indikators relevant. Da im Rahmen der Simulation von Fertigungssystemen der Leistungsindikator über die Zeit gemessen wird, ist zu beachten, dass das System zu Beginn meist keinen eindeutig definierten Zustand hat und sich erst über die Zeit stabilisiert und einen eingeschwungenen Zustand erreicht. Der Abschnitt wird auch als transiente Phase bezeichnet und gibt kein repräsentatives Abbild des Systems. Erst danach kann der Leistungsindikator als repräsentativer Wert, unabhängig vom Startzustand, erhoben werden. Eine gängige zur Beurteilung bekannte Methode ist von Welch, welcher eine graphische Beurteilung vorschlägt (Gutenschwager 2017). In Abbildung 14 ist exemplarisch ein Systemverhalten über die Zeit gezeigt. Der gemessene Durchsatz (gepunktete Linie) steigt bis zum Zeitpunkt 9 an und schwankt danach um einen Durchschnittswert. Dies ist deutlich ab dem Zeitpunkt 9 am konvergierenden gleitenden Durchschnittswert mit einem Zeitfenster von 3 zu erkennen.

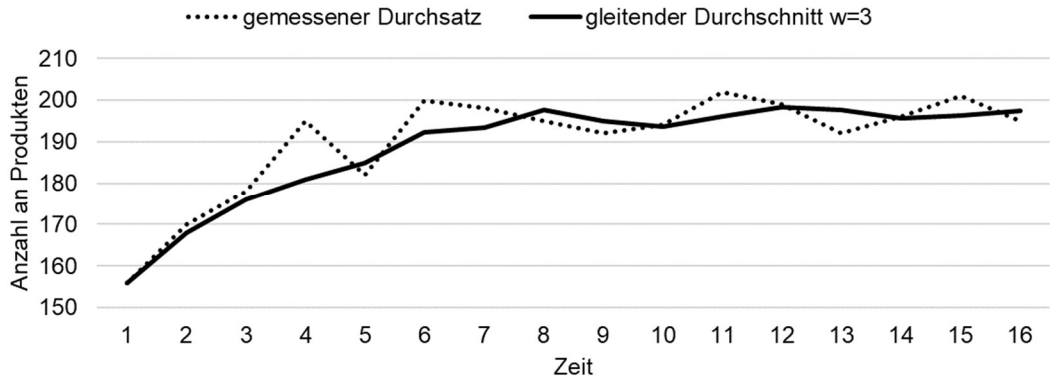


Abbildung 14: Bestimmung der Einschwingphase durch die Betrachtung der Leistung über die Zeit (Gutenschwager 2017)

Die Kombination aus Simulation und Optimierung wird durch diese Methode ebenfalls ermöglicht. Ähnlich wie bei der mathematischen Modellierung, wo das Gleichungssystem zur Evaluation verwendet wird, steht an dieser Stelle die Simulation als Blackbox zur Evaluation des Optimierungsansatzes zur Verfügung. Basierend auf dem Input an Entscheidungsvariablen berechnet sie einen Output und kann diesen als Leistungsindikator an den Optimierer zurückspielen. Dieses Verfahren wird durchgeführt, bis die Abbruchbedingung erfüllt ist. Exemplarisch ist dies in Abbildung 15 dargestellt. Wie bereits bei den GA beschrieben, kann es bei der Verwendung von stochastischen

Simulationsmodellen zu einer langen Laufzeit und häufigen Wiederholungen kommen. (Suhl und Mellouli, 2013)

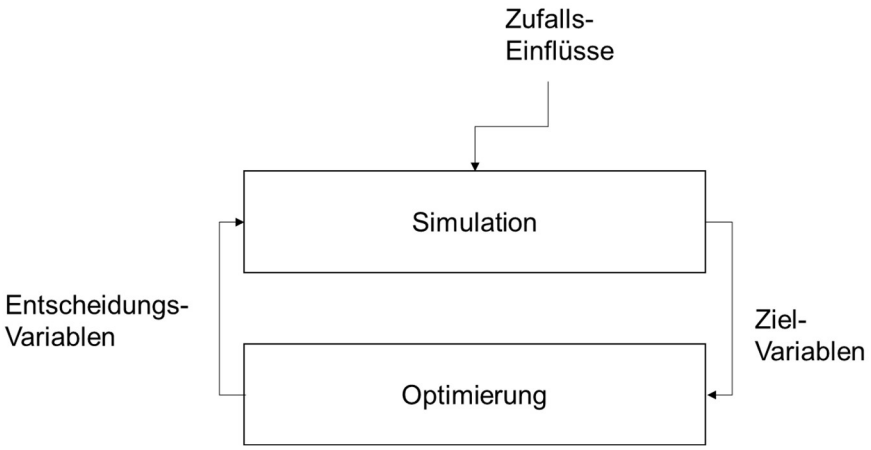


Abbildung 15: Die Simulation kann zur Evaluierung der Optimierung verwendet werden (Suhl und Mellouli, 2013)

3.6. Zusammenfassung Stand der Technik

Die Verwendung von mathematischer Modellierung und optimierenden Verfahren zur Berechnung von zentralen und optimalen Reihenfolgeplänen sowie Routenzuweisung ist für kleine Szenarien möglich. Im Rahmen von größeren und komplexen Fertigungsstrukturen mit Reihenfolge und Routenauswahl ist die Verwendung von exakten Optimierungsalgorithmen auf Grund der langen Laufzeit allerdings nicht wirtschaftlich. Aus diesem Grund wurden Heuristiken entwickelt, die in kurzer Zeit Lösungen berechnen, die sich dem Optimum annähern, dies aber nicht notwendigerweise erreichen. Bei Änderungen im System, wie zum Beispiel einem ungeplanten Maschinenausfall, muss der komplette Plan neu berechnet werden.

Im Gegensatz zur zentralen Zuweisung von Aufträgen zu Maschinen könne diese Vorgänge auch vor Ort an der jeweiligen Maschine, basierend auf dem aktuellen Zustand und der vorhandenen Wartschlange getroffen werden. Wenn die Entscheidungen über die Auswahl des nächsten zu bearbeitenden Vorgangs individuell an den jeweiligen Maschinen getroffen wird, handelt es sich um ein dezentrales Verfahren. Eine dezentrale Entscheidungsfindung kann die Komplexität reduzieren. Dabei müssen aber dynamische Aspekte wie Maschinenausfall und Eilaufträge beachtet werden. Die wohl bekannteste Methode sind die Prioritätsregeln, welche basierend auf einem Attribut der Vorgänge die Reihenfolge aller Wartenden sortieren. Prioritätsregeln haben auf Grund ihrer einfachen Anwendbarkeit und der guten Nachvollziehbarkeit eine

lange Historie in der Industrie. Wenn mehr als ein Attribut in Betracht gezogen wird handelt es sich um kombinierte Prioritätsregeln. Diese können händisch oder algorithmisch erstellt werden. Bekannte Regeln sind zum Beispiel die Holthaus- oder ATCS-Regel. Komplexe und maschinell generierte Regeln umfassen bis zu 70 Terme und sind von Menschen nicht mehr zu verstehen. Die Literatur zeigt weiterhin, dass keine einzelne Regel in allen Situationen allen anderen Regeln überlegen ist.

Aus diesem Grund wurden Methoden entwickelt, die Reihenfolgeregeln dynamisch an vorherrschende Systemzustände anpassen. Ausgehend von einer Wissensbasis können je nach Situation passende Regeln und Gewichte für die einen Terme ausgewählt werden. Der Aufbau einer solchen Wissensbasis geschieht meist durch Simulation und bildet eine Untermenge der möglichen Szenarien ab. Daher sind wissensbasierte Ansätze zur dynamischen Anpassung nur begrenzt in der Lage auf Änderungen zu reagieren. Um in unbekannten Situationen dennoch agieren zu können wird das Systemverhalten mit Regression, entwickelt mit maschinellem Lernen, geschätzt. Die Verwendung von NN, Gaußscher Prozess Regression und anderen Methoden hat gute Ergebnisse geliefert. Dennoch stehen die Ansätze vor der Herausforderung, präzises Systemverhalten vorherzusagen und ihr Wissen auf andere Szenarien zu übertragen.

Im Rahmen von komplexen und dynamischen Szenarien hat RL bereits gute Ergebnisse über ein breites Spektrum an Szenarien erzielt. RL-Agenten sind in der Lage, durch die Interaktion mit ihrem Umfeld komplexe Zusammenhänge zwischen Handlung und Systemzustand zu lernen. Im Kontext der ereignisdiskreten Simulation der Produktionsplanung konnten verschiedene Anwendungsbeispiele zeigen, dass die konkrete Auswahl von Aufträgen zur Bearbeitung auf Maschinen möglich ist und bestehende Heuristiken schlagen kann. Es ist also davon auszugehen, dass die Kombination aus RL und dynamischer Regelauswahl in der Lage sein sollte, dezentral Informationen zu verarbeiten und Entscheidungen zu treffen und schlussendlich die Leistung zu verbessern.

Die bisher beschriebenen Methoden stellen eine gute Grundlage zur dynamischen Anpassung dar, betrachten dabei allerdings nur sehr begrenzte Szenarien und vereinzelte dynamische Aspekte. Eine Methode die über mehrere komplexe Produktionssysteme hinweg, auch in unbekannten Situationen, dynamisch die Reihenfolgeregeln anpasst und so robuste und von Menschen nachvollziehbare Handlungen durchführt, ist zum aktuellen Zeitpunkt nicht bekannt.

4. Handlungsbedarf

Ausgehend von den in Kapitel 2 festgestellten Herausforderungen sind die in Kapitel 3 aufgezeigten Methoden nur bedingt für die Problemlösung geeignet. In diesem Kapitel werden folglich der bestehende Handlungsbedarf sowie die einzelnen Schritte zur Entwicklung einer entsprechenden Lösung beschrieben.

Der Handlungsbedarf ergibt sich aus dem Mangel an passenden Verfahren zum Umgang mit unbekannten Situationen und starken Schwankungen unter Beachtung von Organisationsform, spezifischen Charakteristika des System sowie der Interaktion aus Reihenfolge- und Routenplanung in komplexen Produktionsumgebungen. Die bisherigen Ansätze sind nur bedingt in der Lage die Situation zu erfassen und darauf entsprechend zu reagieren. Weiterhin ist es eine Herausforderung dynamisch, präzise, dezentral und für Anwender verständlich auf unbekannte Situationen zu reagieren. Es ist daher notwendig, eine Methode zu entwickeln die diese Anforderungen erfüllt.

Modellierung des Fertigungssystems

Die Modellierung eines komplexen Produktionssystems bietet die Grundlage für die Evaluation der nachfolgenden Methoden auf Basis logistischer Zielgrößen. Konkret werden in den folgenden Kapiteln ein mathematisches Modell, mehrere Simulationen mit ereignisdiskreter Modellierung erstellt sowie die verwendeten Leistungsindikatoren beschrieben und analysiert.

Durch die mathematische Modellierung wird das Problem als lineares Gleichungssystem definiert und mit bekannten Algorithmen wie dem Simplex für ein kleines Szenario mit statischen Eingaben optimal gelöst. Auch wenn davon auszugehen ist, dass dies nur für kleine Szenarien möglich ist, kann eine optimale Lösung dazu genutzt werden, um das Potenzial der Kombination aus Reihenfolge- und Routenregelung aufzuzeigen. Die Simulation als ereignisdiskretes Modell wird entwickelt, um dynamischen Events sowie die komplexen Interaktionseffekte abzubilden, die das mathematische Modell nicht beschreiben kann. Dabei ist sowohl die Reihenfolgebildung wie auch die Routen- und Fahrzeugauswahl zu betrachten, um die verschiedenen Organisationsformen abbilden zu können. Zudem ist die Betrachtung von reihenfolgeabhängigen Rüstzeiten ein integraler Bestandteil des Modells. Mit Hilfe eines eindeutig definierten Modells kann die Leistungsfähigkeit der neuen Methode zur dynamischen Anpassung von Reihenfolge- und Routenregeln mit anderen Methoden und deren Leistung verglichen werden.

Die Auswahl und Betrachtung eines passenden logistischen Leistungsindikators stellen eine Vergleichbarkeit zwischen den Methoden sicher. Die Formulierung des passenden Leistungsindikators ermöglicht es neben Kriterien wie zum Beispiel der Verspätung oder dem Fertigstellungszeitpunkt auch die Veränderung der Leistung zu erfassen. Je nach gewähltem Indikator können entsprechende Aussagen und Handlungsempfehlungen getätigt werden.

Entwicklung einer Methode (Hyper-Heuristik), die dezentral und dynamisch auf Änderungen in komplexen Szenarien reagieren kann

Da die realen Systeme in den seltensten Fällen vollständig definiert und statisch sind, ist bei der Zuweisung wann welcher Auftrag auf welcher Maschine bearbeitet werden soll ein dynamischer Aspekt zu beachten. Das Auftreten von ungeplanten Ereignissen muss bei der Erstellung der Reihenfolgepläne durch eine Reaktion auf die Änderungen oder eine prädiktive Planung kompensiert werden. Auf Grund der technologischen Fortschritte stehen mehr Informationen und Kommunikationsmöglichkeiten zur Verfügung. Die Interaktion zwischen zentralen und dezentralen Teilnehmern ist bei der Erstellung und Anpassung der Pläne zu berücksichtigen.

Die Literatur zeigt, dass die zur Planung verwendeten zentralen Methoden für kleine Szenarien, nicht aber für reale Anwendungsfälle, Lösungen in polynomieller Zeit errechnen können. Da das RL in komplexen Szenarien bereits gute Ergebnisse gebracht hat, ist davon auszugehen, dass die Methode hier übertragen werden kann. Aus diesem Grund soll eine Methode zur dynamischen Anpassung von Prioritätsregeln mit Hilfe von RL entwickelt werden.

Zu diesem Zweck soll die vorher entwickelte Simulation genutzt werden um als Entwicklungs- und Trainingsumgebung für die neue Methode und den RL-Agenten zur Verfügung zu stehen. Basierend auf der Literatur wird die Verwendung eines RL-Agenten zur dynamischen Anpassung als Hyper-Heuristik mit einfachen (kategorischen Werten) und kombinierten Prioritätsregeln (kontinuierlichen Werte) für die Reihenfolgebildung getestet. Es ist davon auszugehen, dass die Verwendung von TD-Verfahren in diesem Szenario Anwendung findet. Zur erfolgreichen Anwendung müssen die Parameter für das Training des Agenten passend zum jeweiligen Szenario gewählt werden. Im Rahmen der Evaluation müssen der Beobachtungs- und Aktionsraum entsprechend gewählt und ausgewertet werden.

Evaluation der Methode

Um eine Aussage über die Leistungsfähigkeit der neuen Methode zur dynamischen Anpassung von Reihenfolgeregeln bei unterschiedlichen Systemzuständen tätigen zu können muss dies entsprechend evaluiert werden. Weiterhin müssen Tests durchgeführt werden, um eine Aussage über die Generalisierbarkeit des Wissens der neuen Methode machen zu können.

Da eine Erprobung der neuen Methode mit der geforderten Breite nicht im Rahmen eines realen Produktionssystems stattfinden kann, wird das vorab definierte Simulationsmodell unter Betrachtung verschiedener Zustände zur Evaluation genutzt. Im Rahmen der Möglichkeiten werden Parameterstudien durchgeführt und die entwickelte Methode mit anderen Ansätzen verglichen. Die Kombination der zu betrachtenden Faktoren bei den Parameterstudien repräsentieren die komplexen Szenarien. Diese umfassen neben Rüst- und Transportzeiten auch stochastisch verteilten Zwischenankunftszeiten auch schwankende Prozesszeiten.

Exemplarische Erprobung im Rahmen eines realen Produktionssystems

Im Rahmen eines Feldtests wird die Prozesskette von der Datenaufnahme bis zur Implementierung eines trainierten RL-Agenten vereinfacht demonstriert. Ausgehend von einem realen Produktionssystem wird durch Erfassung und Auswertung von Prozessdaten ein digitaler Zwilling als Simulation eines Produktionssystems mit ereignisdiskreter Modellierung erstellt. Diese Simulation wird verwendet, um einen RL-Agenten zu trainieren, der erfolgreich evaluiert und im realen System als Entscheidungsunterstützung implementiert wird. Der im Produktionssystem bereitgestellte Agent wird im Rahmen des Feldtests im Zusammenspiel mit realen Personen verwendet und seine Leistung dokumentiert. Der Vergleich der Leistung des RL-Agenten und Planern gibt Aufschluss über die Anwendbarkeit des Konzeptes. Weiterhin zeigt der Anwendungsfall die möglichen Schnittstellen zwischen virtuellem Training und realem Anwendungsfall.

Ziel

Ziel der Arbeit ist es, eine Methode zu entwickeln, die in der Lage ist, den aktuellen Anforderungen zur Reihfolge- und Routenplanung in einem komplexen Fertigungssystem gerecht zu werden. Unabhängig von der Komplexität des Systems soll die neue Methode in der Lage sein, dynamisch und dezentral

Entscheidungen zu treffen und dabei Aspekte wie spontane Ereignisse genauso zu berücksichtigen wie unvollständige Informationen. Das präzise und schnelle Anpassen des Verhaltens der Methode an den aktuellen Kontext im Produktionssystem soll, zu jeder bekannten und unbekannten Situation, zu einer vergleichbaren oder messbaren Verbesserung der logistischen Leistungsindikatoren gegenüber bekannten Methoden führen.

5. Konzept & Evaluation

Im Folgenden wird die Analyse der Komplexität eines definierten Fertigungsszenarien sowie die Entwicklung der Methode zur dynamischen Anpassung von Reihenfolgeregeln im selbigen vorgestellt.

In Kapitel 5.1 wird das Szenario der Robo Cup Logistik Liga als Anwendungsfall für ein modernes und komplexes Fertigungssystem gezeigt. In dem beschriebenen Fertigungssystem werden fahrerlose Transportsysteme (FTS) zum Materialtransport zwischen den teilweise parallelen Maschinen genutzt. Ein mathematisches Modell zur Berechnung des optimalen Maschinen- und Fahrzeugbelegungsplans sowie eine einfache Prioritätsregel zur Reihenfolgebildung werden bezüglich ihrer logistischen Leistung verglichen.

Nachfolgend wird in Kapitel 5.2 ein Modell als ereignisdiskrete Simulation zur Modellierung von Unsicherheiten durch stochastische Verteilungen präsentiert. Die Interaktion zwischen Reihenfolge-, Routen- und Fahrzeugauswahl-Regeln in unterschiedlichen Systemzuständen wird modelliert und analysiert. Eine Auswertung der Leistungsindikatoren, ein Regressionsmodell zur Vorhersage von Systemverhalten unter unbekannten Zuständen und eine statische Betrachtung der besten Kombination und die Auswirkung auf die logistische Leistung des Systems werden aufgezeigt.

Die Anwendung von RL zur dynamischen Auswahl von Reihenfolgeregeln wird in Kapitel 5.3 gezeigt. Das in 5.2. gezeigte Szenario wird erweitert und angepasst, um einen größeren Lösungsraum und höhere Komplexität zuzulassen. Durch weitere Maschinen und die Berücksichtigung von reihenfolgeabhängigen Rüstzeiten soll ein besserer Bezug zu realen Fertigungssystemen herstellbar sein. Dabei lehnen sich die Erweiterungen an Szenarien aus der Halbleiterfertigung an. Die Definition des Beobachtungs- und Aktionsraums des Agenten wird beschrieben und es werden erste Versuche zur Anpassung einfacher Prioritätsregeln evaluiert. In diesem Szenario sind die Routen- und Fahrzeugauswahlregeln statisch.

In Kapitel 5.4 wird die Verwendung von RL im Kontext von kombinierten Prioritätsregeln verwendet. Als konkretes Anwendungsbeispiel wir die ATCS-Regel mit ihren k-Faktoren vom Agenten dynamisch an die Situation angepasst. Zum besseren Verständnis werden vorab zwei Parameterstudien durchgeführt, die das generelle Verhalten der Regel aufzeigen und als Datengrundlage für eine Referenzmethode verwendet werden. In diesem Kapitel wird weiterhin evaluiert, bis zu welchem Grad der Agent in der Lage ist auf unbekannte Situationen im Produktionsablauf zu reagieren.

In Kapitel 5.5 werden die Methoden DT, NN und RL zur Schätzung der Systemleistung in Kombination mit der ATCS-Regel entwickelt und in unterschiedlichen Szenarien evaluiert. Detailliert werden das Training der Regressionsverfahren sowie die Vor- und Nachteile für die jeweiligen Szenarien erläutert.

Im letzten Kapitel 5.6 wird der generelle Prozessablauf zur Modellerstellung, zum Training des Agenten und die Anwendung in der Produktion unter Laborbedingungen im Kontext der Leuphana Lernfabrik aufgezeigt.

5.1. Die flexible Werkstattfertigung mit fahrerlosen Transportfahrzeugen

Das folgende Szenario stammt aus der Robo Cup Logistik Liga (https://ll.robocup.org/) und wird dort als Wettkampfszenario verwendet. Die Liga konzentriert sich auf die Anwendungen von FTS in der innerbetrieblichen Logistik. Dem RoboCup-Gedanken folgend ist das Ziel dieser Liga, wissenschaftliche Arbeiten zu ermöglichen, um eine flexible Lösung des Material- und Informationsflusses innerhalb der industriellen Produktion durch koordinierte Teams autonomer mobiler Roboter zu erreichen. Die Aufgabe der Roboter ist es, Rohstoffe aus einer Versorgungsstation zu holen, sie in einer definierten Abfolge zwischen Maschinen zu transportieren, die Produktion an diesen Maschinen abzuwickeln und die fertigen Produkte schließlich wieder an eine Versorgungsstation auszuliefern (Voß et al., 2016).

5.1.1. Beschreibung des RoboCup Logisitk Liga Szenarios

Konkret handelt es sich um eine Werkstattfertigung mit sechs Maschinen, zwei Versorgungsstationen und bis zu drei Robotern, die durch optimale Belegung der Maschinen, innerhalb möglichst kurzer Zeit, bis zu sechs Aufträge fertigen müssen. Im Rahmen der Fertigung sind vier verschiedene Produkte mit unterschiedlichen Prozesszeiten möglich. Für diese Studie wurden die Werte vereinfacht und sind in Tabelle 2 für sechs Aufträge exemplarisch dargestellt. Dabei ist zu beachten, dass die Aufträge ab dem Zeitpunkt 0 zur Verfügung stehen. Die Tabelle muss wie folgt gelesen werde: Auftrag J1 ist vom Typ 1 und benötigt als erste Operation eine Bearbeitung auf Maschinengruppe 3, welche 10 Minuten dauert. Im zweiten Schritt benötigt Auftrag J1 eine Bearbeitung auf Maschinengruppe 1 die dann 30 Minuten dauert.

Tabelle 2: Beschreibung der Aufträge mit Auftragstyp, Prozessabfolge und Fälligkeitstermin

Auftrag	Typ	Benötigte Maschinengruppe (Prozesszeit in Minuten)			
J1	1	MG3(10)	MG1(30)	MG2(60)	MG4(70)
J2	2	MG2(80)	MG3(50)	MG1(100)	MG4(40)
J3	3	MG3(50)	MG4(40)	MG1(90)	MG2(10)
J4	4	MG2(50)	MG1(50)	MG3(50)	MG4(40)
J5	4	MG2(50)	MG1(50)	MG3(50)	MG4(40)
J6	1	MG3(10)	MG1(30)	MG2(60)	MG4(70)

Ausgehend von den acht unterschiedlichen Positionen für Versorgungsstationen und Maschinen sind die Transportzeiten zwischen den einzelnen Stationen für das jeweilige Szenario bekannt, d. h. der Transport geschieht für jeweils ein Produkt mit einem Fahrzeug des FTS, welche eine definierte Fahrzeit haben. Es fällt auf, dass die Fahrzeiten in (6) symmetrisch sind, d. h. von A nach B fahren dauert genau so lang wie von B nach A.

$$\begin{Bmatrix} 0 & 27 & 21 & 17 & 27 & 26 & 26 & 27 \\ 27 & 0 & 30 & 14 & 6 & 8 & 30 & 3 \\ 21 & 30 & 0 & 19 & 27 & 22 & 25 & 29 \\ 17 & 14 & 19 & 0 & 15 & 9 & 21 & 11 \\ 27 & 6 & 27 & 15 & 0 & 9 & 30 & 8 \\ 26 & 8 & 22 & 9 & 9 & 0 & 28 & 10 \\ 26 & 30 & 25 & 21 & 30 & 28 & 0 & 29 \\ 27 & 3 & 29 & 11 & 8 & 10 & 29 & 0 \end{Bmatrix} \quad (6)$$

Normale Werkstattfertigungen berücksichtigen auf jeder Seite der Maschine üblicherweise unendlichen Pufferplatz. Im Rahmen der Logistik Liga wird allerdings ein Sonderfall behandelt, welcher die Komplexität reduziert. Da dieser Zwischenspeicherplatz nicht vorhanden ist, kann das Problem als blockierende Umgebung betrachtet werden. In einem entsprechenden Szenario können Maschinen keinen anderen Vorgang bearbeiten, bis der zuletzt bearbeitete Auftrag aus der Maschine gelöscht/entladen wurde. Der Auftrag muss auf der Maschine verbleiben, bis die nächste Maschine verfügbar ist und der Transport somit möglich. Diese Umstände verzögern den Start des nächsten Vorgangs des anstehenden Auftrags.

Diese Situation ist in industriellen Umgebungen gut bekannt und wird häufig bei der Planung von Bahnhöfen oder Operationen in einem Krankenhaus angetroffen. Die Berücksichtigung der Transportvorgänge eines FTS und das

Fehlen eines Puffers an der Maschine, ergeben eine entscheidende Abhängigkeit zwischen der Belegung der Maschine und dem Fahrplan des FTS.

Der hier gewählte logistische Leistungsindikator ist der frühste Fertigstellungszeitpunkt aller Aufträge (C_{max}). Diese absolute Kennzahl ist für einen Vergleich bei wenigen Produkten besser geeignet als ein Durchschnittswert.

Zur klaren Abgrenzung sein an dieser Stelle verdeutlicht: Dezimalzahlen werden im Folgenden durch einen Punkt (.) getrennt – 9.81, Wertepaare und Indices werden in Folgenden durch Komma (,) getrennt - $O_{i,j}$. Dasselbe gilt auch für Vektoren und Matrizen - [a, b, c]. Die Kombination beider ist möglich und sieht exemplarisch folgendermaßen aus: [5, 4.5]

5.1.2. Mathematische Modellierung

Für die Bestimmung des optimalen Plans wurde ein mathematisches Modell entwickelt. Die grundlegende Formulierung des mathematischen Modelles ist aus Poppenborg et al. (2012) übernommen. Die linearen Restriktionen beschreiben die Verwendung von Versorgungsstationen und Transportrobotern. Entscheidend ist es eine Unterscheidung zwischen Bearbeitungs- und Fahrzeiten zu machen. Das Modell verwendet dazu eine umfangreiche Beschreibung der aufeinander folgenden Operationen, um sicher zu stellen, dass nach jeder Maschinenoperation eine Transportoperation folgt. Zu diesem Zweck sind eindeutige Vorrangsbeziehungen definiert, so dass bei einem Paar ($O_{i,j}$, $O_{i,j+1}$) die Operation $O_{i,j+1}$ erst starten kann, wenn Operation $O_{i,j}$ fertig gestellt wurde. Die Notation in Abbildung 16 beschreibt jede Operation abhängig vom Auftrag i und der Position des Schrittes j in der Prozessfolge als $O_{i,j}$. In der Abbildung ist zu erkennen, dass $O_{1,1}$ auf Maschine 1 und $O_{1,2}$ auf Maschine 2 bearbeitet wird. Der Roboter hat eine definierte Fahrzeit von M1 nach M2, beschrieben durch t_{M_1,M_2}. Es folgt, für die Bereitstellung von $O_{2,1}$, eine Leerfahrt von M2 zur Versorgungsstation E und von dort eine Fahrt mit Material von E zu M1. Es ist dabei entscheidend, dass mit unterschiedlichen Maschinenreihenfolgen auch unterschiedlich lange Fahrzeiten entstehen. Wäre der Maschinenbelegungsplan anders, würden sich die Fahrzeiten entsprechend ändern. Die verschiedenen Entscheidungsvariablen des Modells beschreiben, welche Operation vor einer anderen kommt, auf welcher Maschine der jeweilige Auftrag bearbeitet und von welchem Roboter dieser transportiert wird. Weiterhin den Zeitpunkt, zu dem jede Operation eines jeden Auftrags startet. Die Parameter sind die Transport- und

Prozesszeiten sowie die benötigte Operationsreihenfolge für die Aufträge und die vorhandenen Maschinen.

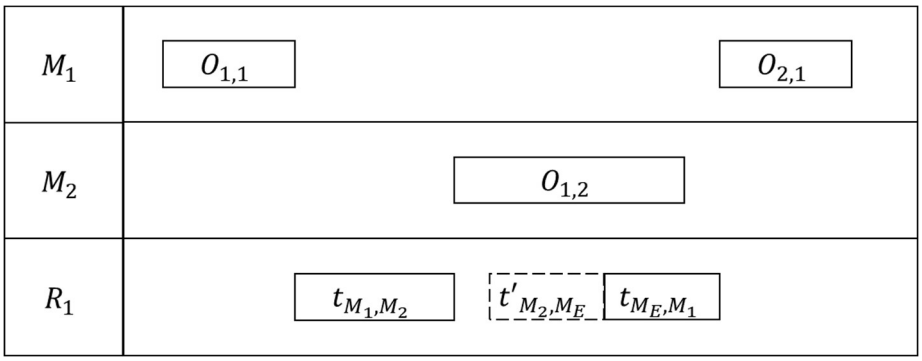


Abbildung 16: Leerfahrten und blockierte Maschinen können die Fertigstellung verzögern (Poppenborg et al., 2012)

Zusätzlich wurde für eine bessere Synchronisierung der Arbeitsschritte und zur Betrachtung der Blockier-Restriktion die Notation von Gröflin und Klinkert (2009) übernommen. Entscheidend ist hierbei die Teilung der einzelnen Prozessschritte in einen Übernahme-, Bearbeitungs- und Übergabe-Prozess. In der Abbildung 17 ist das Gantt-Chart für drei Aufträge mit mehreren Prozessen zu sehen, dabei sind die Aufträge farblich gekennzeichnet. Jeder Prozess j besteht aus den drei Blöcken, der Übernahme (t_j), dem Prozess (p_j) und der Übergabe (h_j). So ist für den Auftrag bestehend aus den Prozessen 1 und 2 auf Maschine 1 und 3 offensichtlich, dass p_2 erst nach der Bearbeitung von p_1 und dem Takeover durchgeführt werden kann. Auffällig ist das p_7 erst spät im zeitlichen Verlauf durchgeführt werden kann, was zu einer langen Blockade von Maschine 2 führt.

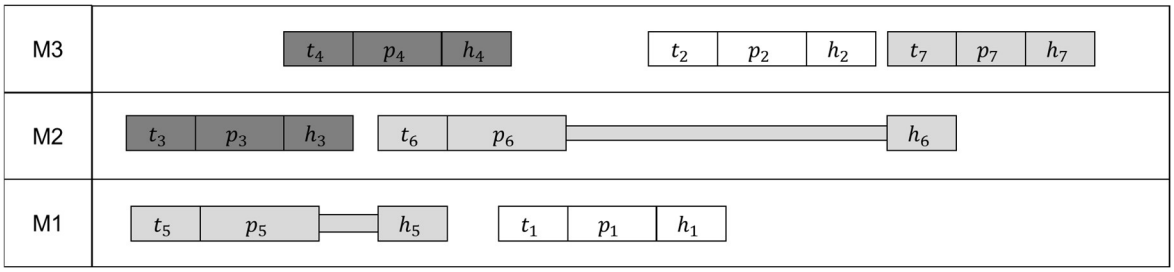


Abbildung 17: In ungünstigen Fällen verzögern Blockier-Zeiten die weitere Bearbeitung (Gröflin und Klinkert, 2009)

Die Kombination der beiden Modelle von Gröflin und Klinkert sowie Poppenborg et al. ermöglicht es, ein komplexes Fertigungssystem mit den für die Logistik Liga benötigten Aspekten zu modellieren. So wird durch die Restriktionen sichergestellt, dass zu jeder Zeit nur eine Operation pro Maschine und Roboter

durchgeführt wird. Weiterhin können Operationen nur auf den definierten Maschinengruppen durchgeführt werden.

Die vollständige Modellierung des Roboterverhaltens mit entsprechenden Transportzeiten, die Restriktion zum Blockieren der Maschinen bis zur Entladung sowie die Integration von Versorgungsstationen und die Betrachtung von Übernahme- und Übergabe-Zeiten ist im Anhang zu finden.

5.1.3. Evaluation des mathematischen Modells

Im Folgenden werden die Laufzeiten und Ergebnisse der Berechnung der optimalen Lösung beschreiben. Die Formulierung des Problems als mathematisches Modell und die resultierende Laufzeit des Lösungsalgorithmus ermöglichen eine erste Abschätzung über die Komplexität des Szenarios. Ein Vergleich zwischen den optimalen Ergebnissen und der Leistung einer dezentralen Heuristik kann von nun an durchgeführt werden.

Bei der Betrachtung des Problems ist darauf hinzuweisen, dass alle Berechnungen in diesem Szenario mit statischen Werten durchgeführt werden. Im Kontext realer Produktionssysteme ist es üblich, dass Prozesszeiten mit einer gewissen Unsicherheit versehen sind. Dies wird im nächsten Kapitel betrachtet.

Für die Berechnung der optimalen Lösung wurde das Modell mit AMPL (Fourer et al., 1993) formuliert und mit dem Gurobi Solver (Gurobi Optimization, 2016) gelöst. Ausgehend von der bekannten Literatur ist davon auszugehen, dass mit zunehmender Anzahl von Aufträgen die Rechenzeit ansteigt. Für die Untersuchung wurde die Anzahl an Aufträgen schrittweise erhöht und die Auswirkungen auf den Leistungsindikator sowie die Rechenzeit bis zur optimalen Lösung dokumentiert. Aufgrund der bereits beschriebenen Komplexität wurde für den Solver ein Zeitlimit von 2 Stunden gesetzt, nach dem die Berechnung auch ohne optimales Ergebnis beendet wird. Es ist davon auszugehen, dass in der Zeit eine gültige, allerdings nicht optimale, Lösung gefunden werden kann. Zum Leistungs-Vergleich wurde die einfache Prioritätsregel FIFO zur Reihenfolgenbildung verwendet. Bei parallelen Maschinen wurde die Auswahl der nächsten Maschine durch die kürzeste Warteschlange ausgewählt.

Wie erwartet und in Abbildung 18 zu erkennen, steigt die Rechenzeit mit steigender Anzahl an Aufträgen an. Die Laufzeit für die Berechnung von fünf Aufträgen im System betrug 70 Minuten. Unter Berücksichtigung der Tatsache, dass der Solver für sechs und mehr Aufträge nach 48 Stunden (173000 Sekunden) keine optimale aber eine gültige Lösung gefunden hat, wurde das

Zeitlimit für den Solver für alle weiteren Berechnungen auf 120 Minuten festgelegt. Das Modell war in der Lage innerhalb der definierten Zeit für bis zu 10 Aufträge eine gültige, wenn auch nicht optimale, Lösung zu finden.

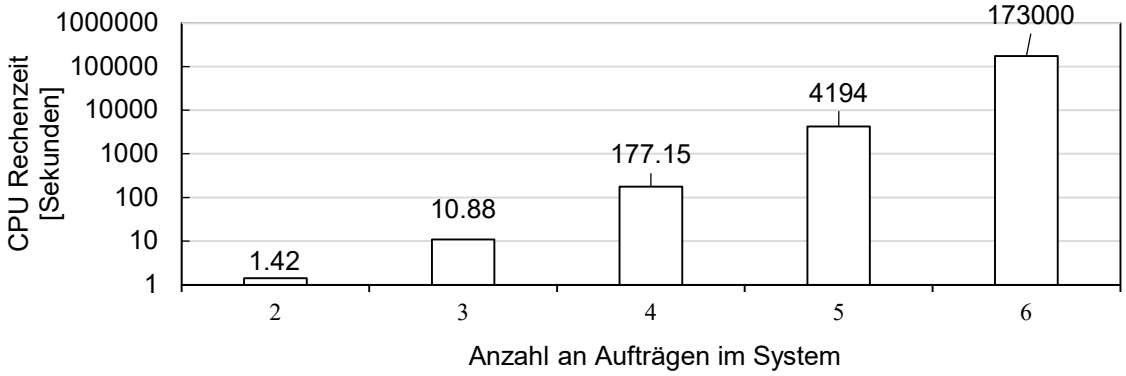


Abbildung 18: Für bis zu 5 Aufträge konnten in vertretbarere Rechenzeit eine optimale Lösung gefunden werden (Heger und Voss, 2017)

Innerhalb der definierten Zeit konnte für bis zu fünf Aufträge die beste Lösung für den frühsten Fertigstellungszeitpunkt (C_{max}) in dem beschriebenen Szenario berechnet werden. Für Instanzen mit mehr als fünf Aufträgen wird die beste gültige Lösung nach 120 Minuten Rechenzeit und deren Lücke (engl. Gap) in Abbildung 19 gezeigt. Das Gap beschreibt die Differenz von der letzten gültigen Lösung zur unteren Schranke. Der Vergleich der besten bekannten Lösung und der Referenzregel FIFO ist ebenfalls in Abbildung 19 gezeigt. Es zeigt sich, dass schon bei 7 Aufträgen im System zwischen der optimalen Lösung und der Referenzregel bis zu 30 % Verbesserungspotenzial liegen können.

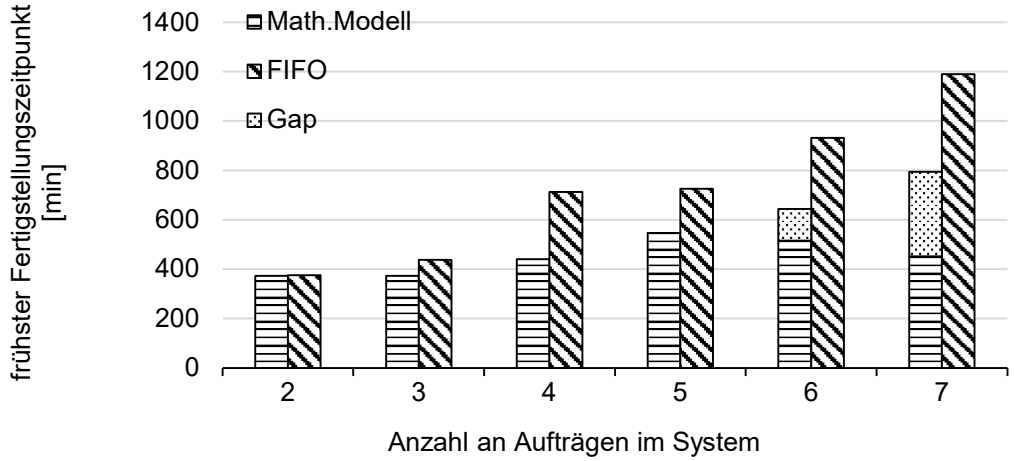


Abbildung 19: Vergleich der optimalen Lösung und FIFO (Heger und Voss, 2018)

5.1.4. Zusammenfassung

Es zeigt sich, dass die optimale Lösung für das beschriebene Szenario mit wenigen Aufträgen möglich ist. Weiterhin zeigt sich, dass zwischen der optimalen Lösung und einer Lösung mit einer Prioritätsregel deutliche Leistungsunterschiede zu erkennen sind. Da die optimale Zuordnung nur bei bis sechs Aufträgen möglich ist, muss eine alternative Methode gefunden werden, um in Szenarien mit kontinuierlichem Zufluss an Aufträgen eine Lösung zu generieren.

5.2. Zusammenspiel von Maschinen und Fahrzeugen in einer flexiblen Werkstattfertigung

Wie im Stand der Technik beschrieben sind über 100 Regeln bekannt, um die Reihenfolgebildung an Maschinen durchzuführen. Weiterhin gibt es für die Auswahl der nächsten Maschine sowie die Auswahl des Fahrzeuges unterschiedliche Prioritätsregeln. In diesem Kapitel werden verschiedene Kombinationen dieser drei Ausprägungen aufgezeigt und getestet, um eine Aussage über die Interaktion der Regeln machen zu können. Es stellt sich weiterhin die Frage, ob diese Kombinationen von Reihenfolge- und Routenregel je nach Situation in der Fertigung unterschiedlich sind. Es wird weiterhin das Szenario aus der Robo Cup Logsitic Liga verwendet, auch wenn die Anzahl der Aufträge nicht mehr begrenzt ist. Weiterhin wurde in diesem Szenario davon ausgegangen, dass über die Zeit Aufträge im System ankommen. Schlussendlich wurde die Restriktion der blockierten Maschinen aufgehoben, um das Potenzial von Reihenfolgeregeln aufzuzeigen.

Da im letzten Kapitel deutlich wurde, dass die mathematische Modellierung in diesem Zusammenhang nicht für große und dynamische Szenarien geeignet ist, wurde ein ereignisdiskretes Simulationsmodell entwickelt. Diese wird als digitaler Zwilling eines Produktionssystems die Prozesse virtuell abbilden. Mit Hilfe des Modells wurden verschiedene Strategien unter unterschiedlichen Systemzuständen getestet. Basierend auf den Simulationsdaten wurde evaluiert, welche Kombination bei welchem Zustand das beste Ergebnis bringt. Dazu wurde ein NN zur Vorhersage von logistischen Leistungsindikatoren basierend auf der Systemauslastung entwickelt.

5.2.1. Beschreibung der flexiblen Werkstattfertigung

Wie bereits beschrieben handelt es sich bei dem Szenario um eine Werkstattfertigung mit sechs Maschinen in vier Maschinengruppen. In zwei der vier Gruppen sind zwei identische Maschinen angeordnet. Die Betrachtung der Anzahl an benötigten FTS und ihren Einfluss auf die Leistung wird im Verlauf der Studie evaluiert. Weiterhin kann die Zwischenankunftszeit der Aufträge basierend auf der gewünschten durchschnittlichen Auslastung gewählt werden. Dabei ist die Zwischenankunftszeit mit einer Poisson Verteilung modelliert. Wie bisher werden vier verschiedene Produkttypen produziert, welche gleichverteilt im System eintreffen. Die Produkttypen haben jeweils vier Prozessschritte und unterscheiden sich in ihren Prozesszeiten. Der Fälligkeitstermin wird über die TWK-Methode berechnet, welche einen vorab definierte Dauer auf den Ankunftszeitpunkt addiert und so den geplanten Fertigstellungszeitpunkt festlegt (Baker, 1984).

Da in dem System durch die stochastisch verteilten Ankunftszeiten eine gewisse Unsicherheit besteht, werden über den Simulationslauf 2500 Aufträge verarbeitet. Nach einer Aufwärmphase von 500 Aufträgen werden 2000 Aufträge für die Leistungsindikatoren (KPIs) berücksichtigt. Weiterhin werden für jede Parameter-Kombination 20 Replikationen durchgeführt. Alle Faktoren sind in Tabelle 3 zusammengefasst.

Tabelle 3:Parameter für die Simulationsstudie

System	Maschinen: 6
	Maschinengruppen: 4
	Organisationsform: Werkstattfertigung
Job	Produktfamilien: 4
	Verteilung der Produktfamilien: Uniform
	Operationen pro Auftrag: 4
	Verteilung der Zwischenankunftszeit: Poisson
	Prozessbearbeitungszeit: statisch
	Fälligkeitstermin: TWK Methode
Simulation	Einschwingphase: 500 Aufträge
	Simulationsdauer: 2500 Aufträge
	Replikationen: 20
KPIs	Durchschnittliche Verspätung
	Durchschnittliche Durchlaufzeit

Im Rahmen des Szenarios sind drei unterschiedliche Entscheidungen zu treffen. Freie Maschinen müssen Aufträge aus der Warteschlange wählen (Reihenfolgebildung), nach der Bearbeitung muss die nächste Maschine zur Bearbeitung ausgewählt werden (Routing) und schlussendlich muss das FTS

gewählt werden (Dispatching), welches den Auftrag zur nächsten Bearbeitung transportiert. Im Folgenden sind für die drei Ausprägungen jeweils drei Regeln zur Evaluation der Kombination genauer erläutert.

Das Konzept der Reihenfolgebildung mit Prioritätsregeln ist bereits aus dem Stand der Technik bekannt. An dieser Stelle werden nur kurz die verwendeten Regeln und ihre Funktion beschrieben. Für die Reihenfolgebildung wurden zum Test gewählt:

- First In First Out (FIFO) – die Operation, welche als erstes in der Warteschlange war, wird als erstes bearbeitet.
- Shortest Processing Time (SPT) – die Operation, mit der kürzesten Bearbeitungszeit wird zuerst bearbeitet.
- Earliest Due Deadline (EDD) - die Operation, deren Auftrag das frühste Fälligkeitsdatum hat wird zuerst verarbeitet.

Diese Regeln für die Reihenfolgebildung werden dann mit Routingregeln kombiniert. Die bekanntesten Routing-Regeln werden exemplarisch noch einmal aufgeführt:

- Least Waiting Time (LWT) - Berücksichtigung der kleinsten Arbeitslast an Operationen in der Warteschlange.
- kleinste Warteschlange (SQ) – Auswahl der Maschine mit der geringsten Anzahl von Operationen in der Warteschlange.
- Least Utilized Machine (LUM) – Auswahl der Maschine mit der geringsten Auslastung.

Die Kombination der Regeln wir dann um eine dritte Kategorie von Regeln, Regeln zur Fahrzeugauswahl (Dispatching), welche nicht im Kapitel 3.2.2 gelistet waren, erweitert. In Kim et al., 1999 werden verschiedene Fahrzeugauswahl-Regeln für FTS vorgestellt und untersuchten. Generell ist dabei zwischen fahrzeug-initiierte und arbeitsplatz-initiierte Regeln zu unterscheiden. Fahrzeug-initiierte Regeln werden angewendet, wenn ein ungenutztes Fahrzeug zwischen mehreren offenen Aufträgen auswählen muss. Arbeitsplatz-initiierte Regeln werden angewandt, wenn mehrere freie Fahrzeuge für eine einzige offene Transportanforderung zur Verfügung stehen. Einfache Regeln, welche seit Jahrzehnten verwendet werden, sind zum Beispiel:

- Shortest Travel Time (STT) - Auswahl des Fahrzeugs mit dem kürzesten Weg zum Ziel.
- Longest Idle Vehicle (LIV) - Auswahl des Fahrzeugs, das von allen Fahrzeugen am längsten im Leerlauf war.
- Least Utilized Vehicles (LUV) - Abfertigung des am wenigsten ausgelasteten Fahrzeugs im System.

5.2.2. Evaluation der Regel-Kombination

Die Zwischenankunftszeit der Aufträge wurde für die Evaluation so gewählt, dass die durchschnittliche Maschinenauslastung bei 80 % liegt. Weiterhin werden fünf FTS betrachtet, was zu einer Auslastung von etwa 40 % führt. Aus den jeweils drei Ausprägungen für Reihenfolge-, Routen- und Fahrzeugauswahl-Regel ergeben 27 verschiedene Regelkombinationen, die unter den definierten Systemzuständen geprüft und bezüglich ihrer logistischen Leistung evaluiert werden.

Wie in Tabelle 4 zu sehen ist, erreicht die Routing-Regel „LWT“ in allen Kombinationen die niedrigsten Werte für die Durchlaufzeit. Die Regel „SQ“ für die Maschinenauswahl liefert ebenfalls gute Ergebnisse und landet an der zweiten Position. Weiterhin lässt sich aus der Tabelle erkennen, dass die Fahrzeugauswahl mit der „STT“-Regel sehr gute Ergebnisse liefert. Zum besseren Verständnis wurden die Ergebnisse der „LWT“-Regel nochmal genauer betrachtet und sind in Abbildung 20 gezeigt. In diesem Fall zeigt sich, dass bei einem zweifachen Standardfehler die Unterschiede in der Reihenfolgeregel in Kombination mit der Verwendung von „STT“ nicht signifikant unterschiedlich sind. Die Verwendung der „LUV“-Regel scheint in diesem Szenario die zweitbeste Wahl zu sein.

Tabelle 4: durchschnittliche Durchlaufzeit bei unterschiedlichen Regelkombinationen

Dispatching	Sequencing	Routing		
		SQ	LUM	LWT
LIV	SPT	686	979	631
	EDD	694	1099	635
	FIFO	687	910	**626**
LUV	SPT	675	963	620
	EDD	682	1061	627
	FIFO	676	893	**619**
STT	SPT	668	950	610
	EDD	674	1054	610
	FIFO	669	883	**607**

Es lässt sich feststellen, dass die Kombination von „LWT“ als Routingregel und „STT“ als Fahrzeugsauwahlregel unabhängig von der Reihenfolgeregel gute Ergebnisse bringt. Dies hat damit zu tun, dass bei einer durchschnittlichen Auslastung von 80 % nur sehr wenige Aufträge in den Warteschlangen vor der Maschine sind und damit alle Regeln ähnlich gut sein müssen.

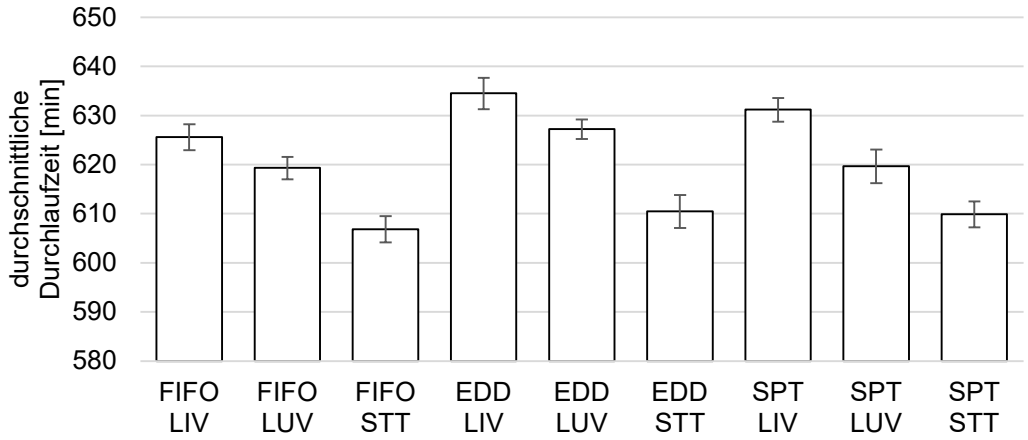


Abbildung 20: Detaillierter Vergleich der Regel-Kombination unter Verwendung von LWT (Heger und Voß, 2019)

Um weitere Aussagen über die Leistung der Kombination aus Reihenfolge-, Routen- und Fahrzeugauswahlregeln in unterschiedlichen Situationen treffen zu können wurde die Parameterstudie erweitert. Zu diesem Zweck wurden Zwischenankunftszeiten definiert, die zu einer Maschinenauslastung zwischen 65 und 99 % führen. Weiterhin wurde die Anzahl der Fahrzeuge von zwei bis fünf variiert, um eine Auslastung von 40 % bis 100 % zu erreichen. Daraus ergaben sich bei fünf unterschiedlichen Zwischenankunftszeiten und vier Stufen von FTS insgesamt 540 unterschiedliche Parameterkombinationen die jeweils zehn Mal repliziert wurden. Weiterhin wurde an dieser Stelle die durchschnittliche Verspätung aller Aufträge im System gemessen.

So ergibt sich für die jeweiligen Kombinationen aus Reihenfolge-, Routen-, und Fahrzeugauswahlregel bei unterschiedlichen Auslastungen von Fahrzeugen und Maschinen der folgende Sachverhalt: Je nach Systemzustand sind unterschiedliche Regelkombinationen Vorteilhaft, um die durchschnittliche Verspätung zu reduzieren. In Tabelle 5 werden exemplarisch zwei Anwendungsbeispielen aus der Simulationsstudie gezeigt. Hierbei handelt es sich um 8 der 540 Ergebnisse, dabei werden 2 der 27 Regelkombination sowie vier der möglichen Kombinationen von Maschinen und Fahrzeugauslastung gezeigt. Es zeigt sich, dass die Regelkombination (STT, FIFO, SQ) auf der rechten Seite bei 90 % Maschinenauslastung und 90 % Fahrzeugauslastung um etwa 11 % schlechter als (LUV, SPT, SQ) ist. Andererseits ist die linke Regelkombination für andere Auslastungsszenarien bis zu 9 % schlechter. Die fett gedruckten Zahlen zeigen im Vergleich, für die jeweilige Kombination aus Maschinen- und Fahrzeugauslastung, den besten Wert an. Bei beiden

Regelkombinationen zeigt sich deutlich, dass mit steigender Auslastung auch die durchschnittliche Verspätung steigt, jedoch bei der einen Kombination mehr als bei der anderen.

Tabelle 5: Durchschnittliche Verspätung in Abhängigkeit von Systemzustand und Regelkombination

Regelkombination (STT, FIFO, SQ)	90 % FTS Auslastung	65 % FTS Auslastung	Regelkombination (LUV, SPT, SQ)	90 % FTS Auslastung	65 % FTS Auslastung
90 % Maschinen Auslastung	**2968**	3015	90 % Maschinen Auslastung	3280	**2833**
70 % Maschinen Auslastung	350	215	70 % Maschinen Auslastung	**322**	**195**

5.2.3. Regressionsverfahren zur Schätzung des Systemverhaltens

Da die Simulationsstudie mit definierten Parametern nur bestimmte Systemzustände abbilden kann, an dieser Stelle aber eine generelle Aussage über die Verwendung der Regelkombinationen im System angestrebt wird, ist es notwendig das Verhalten der Regeln für die unbekannten Systemzustände mit Hilfe eines Regressionsverfahrens zu schätzen.

Zu diesem Zweck kann ein Regressionsverfahren zur Vorhersage der durchschnittlichen Verspätung trainiert werden. Dadurch ist es möglich, eine Aussage über die Leistung der jeweiligen Regelkombination zu treffen, auch wenn keine simulierten Daten zu dieser Situation vorliegen. Basierend auf den Daten der durchgeführten Studie wurden im Folgenden zwei Regressionsmodelle entwickelt, eine lineare Regression und ein NN. Bei den aufgezeichneten Daten werden die ausgewählte Reihenfolge-, Routen-, und Fahrzeugauswahlregel sowie die durchschnittliche Maschinen- und Fahrzeugauslastung als Eingabe und die durchschnittliche Verspätung als Ausgabe betrachtet. Der Datensatz wird gemischt und danach in Test- und Trainingsdaten unterteilt. Eine kurze Vorstudie zeigte, dass ein NN mit drei Schichten mit jeweils 32 Neuronen und maximalen 10000 Iterationen ausreichende Ergebnisse bringt. Die Standardparameter der Toolbox wurden übernommen, in diesem Fall war der verwendete Solver „adam“, die Aktivierungsfunktion RELU. Es wurde keine automatisierte Hyperparameteranpassung vorgenommen. Als Vergleichswert der Leistung der Regressionsverfahren wurden „Mean Square Error“ und „Mean Absolut Error“ verwendet. Beide Verfahren wurden mit „Scikit-learn“ in Python entwickelt und evaluiert. Es zeigte sich, dass das NN eine deutlich höhere Präzision (geringere Werte für den durchschnittlichen quadratischen sowie den durchschnittlichen

absoluten Fehler) hatte als die lineare Regression. Dies erscheint plausibel, da das Verhalten der durchschnittlichen Verspätung besonders mit steigender Maschinenauslastung nicht linear ist.

Um eine Aussage über die jeweils beste Regel treffen zu können wird zu jeder möglichen Kombination aus Maschinen- und Fahrzeugauslastung für alle 27 Regelkombinationen die durchschnittliche Verspätung geschätzt. Für die jeweilige Kombination aus Maschinen- und Fahrzeugauslastung werden dann die Leistungen der Regel-Kombinationen verglichen und der niedrigste Wert dokumentiert. In Abbildung 21 sind für unterschiedliche Systemzustände exemplarisch 5 Regelkombinationen gezeigt. Es fällt auf, dass wie bereits in der Tabelle 5 gezeigt, die Kombination (LUV, SPT, SQ) bei 90 % Maschinen- und 65 % Fahrzeugauslastung ebenso wie (STT, FIFO, SQ) bei 90 % Maschinen- und 90 % Fahrzeugauslastung die niedrigste Verspätung erreichen. Da es 27 verschiedene Regelkombinationen gibt, ist es nicht möglich das Ergebnis anschaulich zu visualisieren, jedoch tabellarisch festzuhalten. Da die Verwendung von unterschiedlichen Regelkombinationen lediglich bei höheren Auslastungen relevant wird, werden in der nächsten Studie lediglich Szenarien mit über 60 % Maschinenauslastung betrachtet. Weiterhin zeigt sich hier, dass die Fahrzeugauswahl basierend auf der kürzesten Anfahrtszeit (STT) häufig gute Ergebnisse bringt.

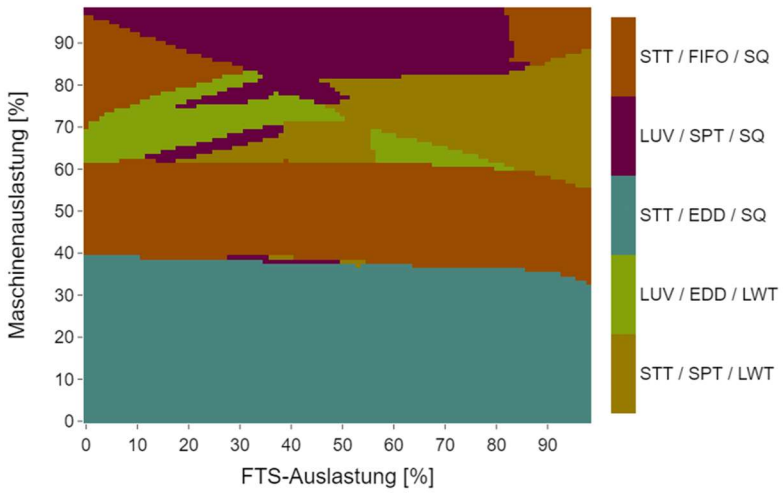


Abbildung 21: 5 Regelkombinationen bei unterschiedlicher Maschinen- und Fahrzeugauslastung (Heger und Voss, 2019)

Um den Zusammenhang zwischen den zwei Auslastungen, unterschiedlichen Regelkombination sowie durchschnittlicher Verspätung darzustellen sind eigentlich drei Achsen notwendig. Wissend, dass mit steigender Maschinenauslastung auch die durchschnittliche Verspätung steigt, soll an dieser Stelle noch einmal verdeutlicht werden, dass unterschiedliche

Regelkombinationen für unterschiedliche Systemzustände benötigt werden. Aus den Abbildungen ist abzuleiten, dass schon eine kleine Änderung des Systemzustandes die beste Kombination an Reihenfolge-, Routen- und Fahrzeugauswahl ändern kann. Es gilt dieses Wissen zu speichern und im Anwendungsfall dynamisch auswählen zu können.

5.2.4. Zusammenfassung

Im Rahmen des vorgestellten Szenarios einer Werkstattfertigung mit mehreren FTS zeigt sich, dass die Lösung des Problems unter Verwendung mathematischer Modellierung und ganzzahliger Optimierung bis zu einer begrenzten Anzahl an Aufträgen zwar möglich, in real Szenarien aber nicht wirtschaftlich ist. Aufgrund der komplexen Interaktion zwischen Maschinen- und Fahrzeugbelegung sowie den möglichen Transportzeiten wird eine lange Rechenzeit benötigt, um den optimalen Plan zu erstellen. Aufgrund von Unsicherheiten im System wie zum Beispiel der Zwischenankunftszeit von Produktionsaufträgen ist die Berechnung der optimalen Lösung an zentraler Stelle für dieses Szenario nicht geeignet. Als eine Alternative bietet sich die Verwendung von Prioritätsregeln an. Diese haben unter unterschiedlichsten Anforderungen in vielen Szenarien bereits gute Leistung erbracht und sind in der Lage auch unter Unsicherheit stabile Leistung zu bringen. Es ist offensichtlich, dass es zwischen der optimalen Lösung und der Verwendung von Prioritätsregeln einen großen Unterschied in der erreichten Leistung gibt.

Im Rahmen der Simulationsstudien zeigt sich, dass eine spezifische Kombination von Reihenfolge-, Routen- und Fahrzeugauswahlregeln für eine definierte Auslastung benötigt wird. Es zeigt sich im Rahmen der durchgeführten Simulationsstudien aber auch, dass keine Regelkombination für alle Systemzustände die beste Leistung liefert. Es gilt also herauszufinden, bei welchem Systemzustand welche Regelkombination die beste Leistung liefert.

Um eine Abschätzung über die Leistung der Regelkombinationen machen zu können wurde ein NN als Regressionsverfahren trainiert. Dies wurde dann genutzt, um für bisher unbekannte Systemzustände eine Schätzung über die Leistung der Regelkombination zu tätigen und so im begrenzten Rahmen die beste Regel für alle möglichen Systemzustände zu finden. Die so aufgebaute Wissensbasis kann verwendet werden, um je nach Systemzustand eine gute Regelkombination auszuwählen. Eine dynamische Anpassung ist somit möglich, zu diesem Zeitpunkt aber noch nicht durchgeführt.

5.3. Entwicklung der Methode zur dynamischen Anpassung von einfachen Prioritätsregeln

Im Rahmen der dynamischen Anpassung stellt sich bei gegebener Wissensbasis die Frage, wie das gewonnen Wissen angewendet werden kann. Zu diesem Zweck werden die Leistungsindikatoren über den Simulationslauf genauer betrachtet. Weiterhin wird eine Methode entwickelt, die das gewonnene Wissen effizient und effektiv anwenden kann.

5.3.1. Erweiterung der flexiblen Werkstattfertigung

Die im folgenden gezeigte Simulationsstudie baut auf dem Beitrag von Holthaus und Rajendran (2000) auf und verwendet entsprechende Werte für Bearbeitungs- und Rüstzeiten. Die Anzahl an verfügbaren Maschinen wird von vorher sechs auf jetzt zehn erhöht, wobei jeweils zwei nicht identische Maschinen in einer Gruppe zusammengefasst sind. Jeder Auftrag hat zehn Prozessschritte, wobei jede Maschinengruppe immer 2-mal benötigt wird. Die Bearbeitungszeiten für jeden Prozess werden aus einer Gleichverteilung im Bereich von 1 bis 99 gezogen. Der Transport zwischen den Stationen wird weiterhin durch FTS durchgeführt, die Fahrzeit wird zufällig aus einer Normalverteilung mit einem Mittelwert von 10 gezogen. Diese Werte für die Fahrtzeit, die in Anlehnung an Kim et al. (1999) gewählt werden, führen zu einem Prozess-/Transportzeit-Verhältnis von fünf. Die Auswirkungen größerer Verhältnisse zwischen den beiden Parametern wurden nicht betrachtet.

Zum komplexen Szenario der flexiblen Werkstattfertigung mit mehreren FTS kommen nun noch unterschiedliche Verteilungen der Produktfamilien sowie reihenfolgeabhängige Rüstzeiten hinzu. Die Kombination aus unterschiedlichen Anteilen an Produktfamilien wird als Produktmix bezeichnet. Der Produktmix [70, 30, 0, 0] besteht also aus 70 % der Familie 1 und zu 30 % aus der Familie 2. Die Rüstzeiten zwischen den Familien sind dabei feste, asymmetrische und sequenzabhängige Werte. Die Werte sind in (7) aufgezeigt und werden wie folgt gelesen: Die Rüstzeit von Familie 1 auf Familie 3 beträgt 10 Minuten. Die Rüstzeit von Familie 4 auf Familie 1 beträgt 5 Minuten. Wenn zwei Aufträge derselben Familie nacheinander gefertigt werden, fallen keine Rüstzeiten an. Wie bereits in den vorhergehenden Studien wird der gewünschte Fertigstellungszeitpunkt durch die TWK-Methode und den DueDate Faktor festgelegt.

$$\begin{pmatrix} 0 & 5 & 10 & 25 \\ 5 & 0 & 10 & 25 \\ 5 & 5 & 0 & 25 \\ 5 & 5 & 10 & 0 \end{pmatrix} \quad (7)$$

Auf Grund des komplexeren Szenarios und dem stärkeren Einfluss der Unsicherheit wird die Länge der Simulation angepasst, insgesamt werden 12500 Aufträge im System gefertigt. Die ersten 2500 Aufträge werden als Einschwingphase in der Rechnung der durchschnittlichen Verspätung nicht berücksichtigt (Welch, 1983). Die Zusammenfassung der Parameter ist in Tabelle 6 gezeigt.

Tabelle 6: Zusammenfassung der Simulationsparameter

System	Maschinen: 10
	Maschinengruppen: 5
	Organisationsform: Werkstattfertigung
Job	Produktfamilien: 4
	Verteilung der Produktfamilien: nach Produktmix
	Operationen pro Auftrag: 10
	Verteilung der Zwischenankunftszeit: Poisson
	Prozessbearbeitungszeit: 1 – 99
	Verteilung der Prozesszeit: gleichverteilt
	Fälligkeitstermin: TWK Methode
Simulation	Einschwingphase: 2500 Aufträge
	Simulationsdauer: 12500 Aufträge
	Replikationen: 30
KPIs	Durchschnittliche Verspätung

Um eine Reihenfolgebildung basierend auf den Rüstzeiten durchführen zu können wird zusätzlich zu den bekannten Regeln die Prioritätsregel „SIMSET“ eingeführt. Diese weist Aufträgen mit derselben Rüstfamilie eine hohe Priorität zu und reduziert somit die Rüstzeiten im System (engl. Similar setup preferred).

5.3.2. Methode zur dynamischen Anpassung einfacher Prioritätsregeln

Wie bereits im Stand der Technik erläutert, hat RL gute Ergebnisse bei der dynamischen Anpassung bei unterschiedliche Systemzustände gezeigt. Aus diesem Grund wird eine Methode zur dynamischen Anpassung der Reihenfolgeregeln mit RL in diesem Szenario entwickelt. Der Agent wird darauf trainiert, die Prioritätsregel passenden zum Systemzustand zu wählen und so die Leistung zu optimieren.

Im Folgenden (Abbildung 22) ist exemplarisch gezeigt, wie über den Verlauf der Simulation (einer Episode), über mehrere Schritte (jeweils eine Woche

Simulationszeit) Prioritätsregeln für alle Maschinen gewählt werden. Nach einer kurzen Einschwingphase (für die Begründung siehe 3.5) werden die Systemzustände betrachtet und pro Schritt eine Reihenfolgeregel ausgewählt. Über den Verlauf einer Episode (ein einzelner Simulationslauf) werden 70 Schritte durchgeführt. Auf Grund der stochastischen Schwankungen im System werden die Simulationsläufe wiederholt, in diesem Fall bis zu 4000 Simulationsläufe. Für jeden Schritt werden Systemzustand, gewählte Handlung (Reihenfolgeregel) sowie die Veränderung Leistungsindikators dokumentiert. In diesem Fall ergeben sich so 280000 Datenpunkten anhand derer der Agent das NN und so seine Strategie trainiert. Das hier beschriebene Vorgehen basiert auf der Umsetzung des TD-Verfahrens, welches in Abschnitt 3.3 beschrieben wurde.

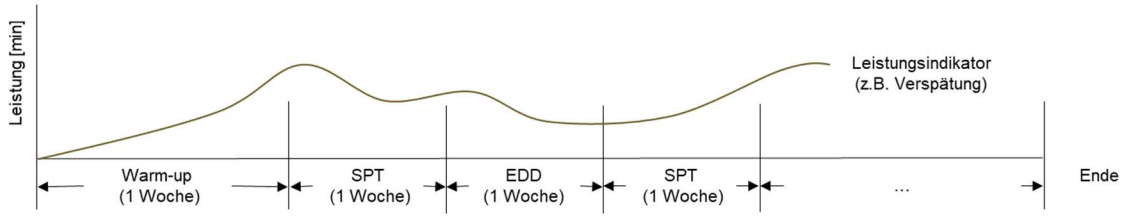


Abbildung 22: Konzept der dynamischen Anpassung (eigene Darstellung)

In diesem Fall kann der Beobachtungsraum durch 27 Werte beschreiben werden. Dazu gehören die zehn durchschnittlichen Maschinenauslastungen sowie die Arbeit in jeder Wartschlange beschrieben durch die Menge an Arbeitsaufträgen und der jeweiligen Dauer der Aufträge. Weiterhin wird der aktuelle Produktmix dokumentiert, genauso wie die verwendete Regel und die durchschnittliche Verspätung im System. Die Werte sind in Tabelle 7 mit entsprechenden Beispielwerten gezeigt. Der mögliche Handlungsraum sind die vier verschiedenen Reihenfolgeregeln aus denen gewählt werden kann (FIFO, SPT, EDD, SIMSET).

Tabelle 7: Exemplarische Beobachtungen des Agenten im System

Beobachtung [Einheit]	Beispielwert
Durchschnittliche Maschinenauslastung [%]	[90, 88, 92, 89, 97, 95, 91, 90, 91, 89]
Arbeit in der Warteschlange [min]	[50, 12, 18, 51, 89, 58, 39, 55, 21, 17]
Produkt mix [%]	[50, 50, 0, 0]
Aktuell verwendete Regel	[1]
Kurzfristige Verspätung [min]	210
Durchschnittliche Verspätung [min]	160

Die Belohnung des Agenten pro Schritt ergibt sich durch die prozentuale Veränderung der durchschnittlichen Verspätung zwischen den beobachteten

Zeitpunkten (t und $t+1$) geteilt durch den Maximalwert der beiden Beobachtungen. Somit wird sichergestellt, dass die Belohnungsfunktion unabhängig vom Produktmix und der Zwischenankunftszeit verwendbar ist. Da zu Beginn der Simulation die Aufträge alle rechtzeitig fertiggestellt werden und erst im späteren Verlauf Aufträge ihren Fertigstellungstermin nicht einhalten können, wird die Belohnung negativ sein. Da der Agent seine Belohnung maximieren möchte wird mit der in Gleichung (8) gezeigten Formel die durchschnittliche Verspätung minimiert.

$$Belohnung = \frac{\bar{T}_t - \bar{T}_{t+1}}{max(\bar{T}_t, \bar{T}_{t+1})} \quad (8)$$

Während des Trainings des Agenten werden über den Verlauf der Simulation entsprechende Zustand-Handlungs-Paare mit der resultierenden Belohnung dokumentiert. Zu diesem Zweck muss die Menge an Handlungen pro Simulation und die Anzahl der Simulationsläufe evaluiert werden. Weiterhin müssen die Hyperparameter für den Agenten zusätzlich zu den Hyperparametern des NN an das Szenario angepasst werden. Für den Agenten müssen neben dem ε-Wert (zur Auswahl zufälliger Aktionen) auch ein entsprechender ɣ-Wert (zur Bewertung des Langzeitverhaltens) gewählt werden. Für das NN wird die Anzahl an versteckten Schichten so wie die Anzahl der Neuronen entsprechend gewählt. Für das folgende Experiment wurden die vorgeschlagenen Standardparameter übernommen und geringfügig an die Laufzeit und Häufigkeit der Simulation angepasst.

5.3.3. Evaluation des Trainings der Methode

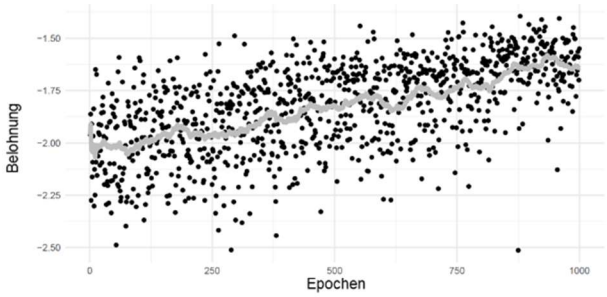
Zur Evaluation des Agenten wird im Folgenden die Belohnung während des Trainings betrachtet, um eine Aussage über das Verhalten zu treffen. Die Auswahl des Zeitintervalls zwischen den Aktionen des Agenten ebenso wie die Dauer eines einzelnen Laufes und die Anzahl an Wiederholungen wurde experimentell ermittelt und wird an dieser Stelle nicht näher betrachtet.

In dem beschriebenen Szenario zeigte sich, dass die Verwendung von 1000 Epochen mit jeweils 70 Aktionen zu guten Ergebnissen für einen einzelnen Produktmix bei einer definierten Systemauslastung geführt hat. In Abbildung 23 (a) ist das Training des Agenten mit einem einzelnen Produktmix abgebildet, wobei jeder Punkt für einen Trainingslauf steht.

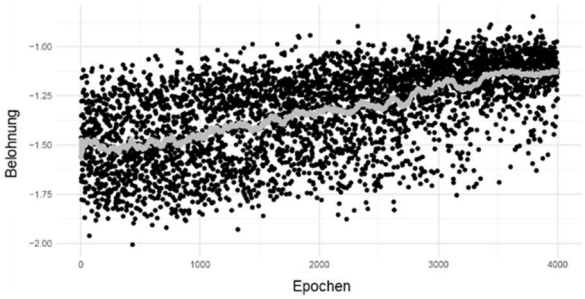
Da bei der Verwendung von mehr Produktmixen im System ein komplexerer Sachverhalt anzunehmen ist, sollte die Trainingsdauer entsprechend der Anzahl

an Produktmixen angepasst werden. In Abbildung 23 (b) ist der Verlauf des Trainings mit vier unterschiedlichen Produktmixen abgebildet. Jeder Punkt beschreibt eine Epoche, die Linie beschreibt den gleitenden Mittelwert über die letzten 100 Epochen. Bei der Betrachtung der Belohnung ist darauf zu achten, dass es sich hierbei nicht um einen absoluten, sondern um einen relativen Wert handelt, der in keiner direkten Verbindung mit der Leistung des Systems steht.

Es ist klar zu erkennen, dass in beiden Fällen die Belohnung über den Verlauf des Trainings einen positiven Trend hat. Die Belohnung ist hierbei jedoch durch die Leistung der jeweiligen Regeln begrenzt. Es ist dem Agenten nicht möglich besonders gute und schlechte Handlungen durchzuführen. Bei genauer Betrachtung fällt auf, dass in Abbildung 23 (b) bis zu 1000 Epochen nur ein geringer Anstieg der Belohnung zu erkennen ist. Daraus lässt sich schließen, dass der Agent über den Verlauf des Trainings erst mit der Zeit Handlungen lernt, die die Verspätung reduzieren. Die Schwankung der Belohnung zum Ende des Trainings lassen sich durch den gewählten ε-Wert erklären: Selbst, wenn der Agent die beste Entscheidung bereits kennt, werden 10 % seiner Handlungen zufällig gewählt, um alternative Möglichkeiten zu erforschen. Dieses Verhalten der zufälligen Handlungsauswahl wird lediglich im Training verwendet, nicht in der späteren Anwendung.



(a) Trainingsverhalten bei einem Produktmix.



(b) Trainingsverhalten bei vier Produktmixen.

Abbildung 23: Abhängig vom Szenario muss die Trainingsdauer des Agenten angepasst werden (Heger und Voss, 2020)

5.3.4. Evaluation der Methode im Szenario

Wie bereits in den vorherigen Abschnitten gezeigt, variiert die beste Regel basierend auf dem Systemzustand. In diesem Szenario ist die Rüstzeit ein entscheidender Einflussfaktor, die vom Produktmix abhängt. Um den Mehrwert der neuen Methode zu zeigen und die verschiedenen, trainierten Agenten zur dynamischen Anpassung bewerten zu können, wird die Strategie daher in verschiedenen Szenarien (verschiedenen Produktmixen) getestet. Exemplarisch sind in der folgenden Abbildung 24 die Referenzregeln und ihre Leistung in unterschiedlichen Situationen (mit 85 und 90 % Einlastung) gezeigt. Es fällt auf, dass bei einer hohen Einlastung und Produktmixen mit hohem Rüstzeitanteil die Regel SIMSET oder SPT zu guten Ergebnissen führt.

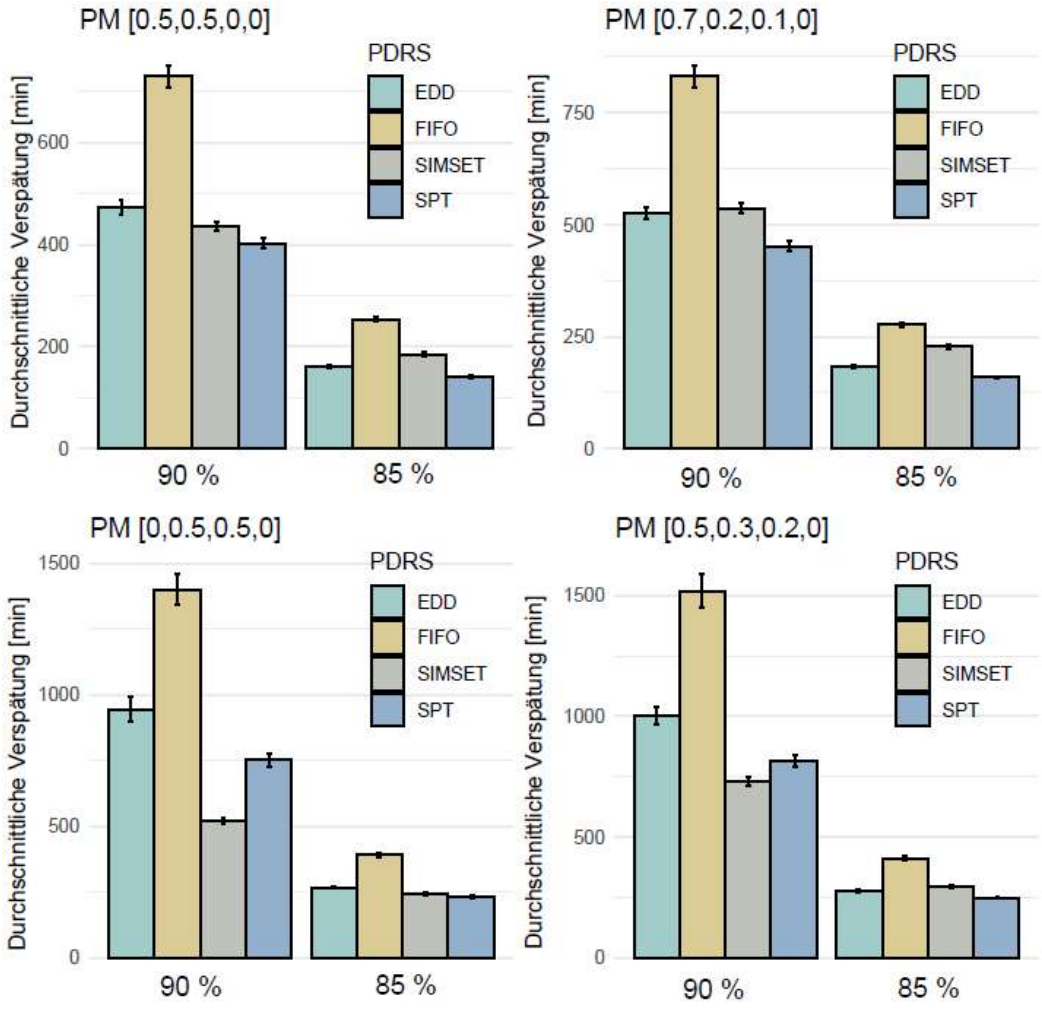


Abbildung 24: Die beste Regel ist abhängig vom Systemzustand und dem Produktmix (Heger und Voss, 2020)

Die erste Überprüfung des Lernerfolges und das Erkennen der Zusammenhänge erfolgt durch eine visuelle Kontrolle des Verhaltens über die Zeit. Danach zeigt ein Vergleich zwischen den unterschiedlichen Agenten die Leistung gegen die Verwendung von statischen Regeln. Dabei werden der Agent, der lediglich auf einem Produktmix trainiert wurde, und der zweite Agent, der auf vier verschiedene Mixe trainiert wurde, in zwei Szenarien evaluiert.

Weiterhin wurde in diesem Zusammenhang die Frequenz der Handlungen evaluiert. In diesem Szenario prüft der Agent den Systemzustand wöchentlich und passt die Reihenfolgeregel entsprechend an. Die wöchentliche Betrachtung ermöglicht es, die Auswirkungen der Handlung auf ~200 Aufträgen im System zu beurteilen. Ausgehend vom Konzept aufgezeigt in Abbildung 22 sollte der Agent also über den Verlauf der Simulation unterschiedliche Regeln verwenden, wenn es die Situation bedarf. Zur visuellen Kontrolle des Verhaltens ist in Abbildung 25 die Leistung des Systems als durchschnittliche Verspätung über die letzten 200 Aufträge (gepunktet) und die Gesamtlaufzeit (durchgezogen) sowie die gewählte Regel (gestrichelt) über die Zeit abgebildet. Es ist gut zu erkennen, dass bei steigenden Werten für die Verspätung der Agent die SIMSET-Regel wählt. Wenn die Verspätung dann wieder fällt, wird SPT gewählt. Bei der Beobachtung des Systems über die Zeit fällt auf, dass der Agent sich lediglich zwischen SIMSET und SPT entscheidet. Das lässt darauf schließen, dass der Agent in der Lage ist den Zusammenhang zwischen Rüstzeit und Verspätung zu erkennen. Im nächsten Schritt wird geprüft, ob dieses Verhalten die Leistung des Systems verbessern kann.

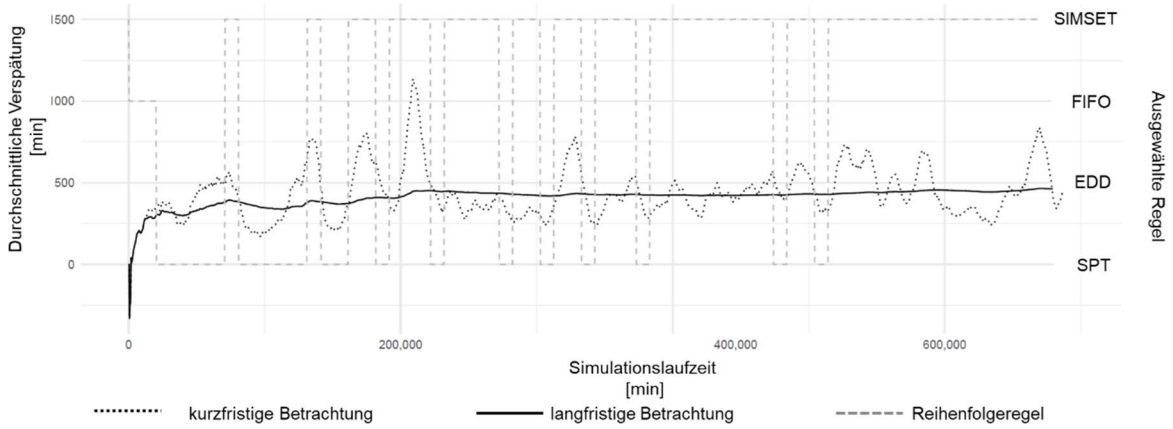
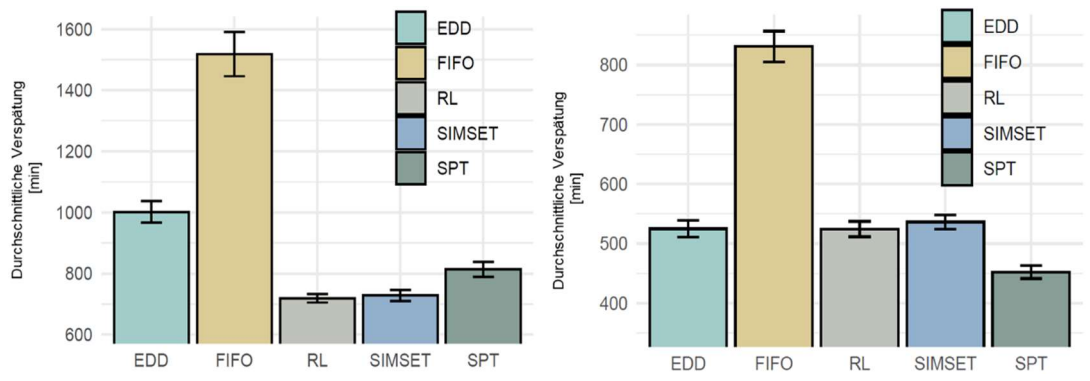


Abbildung 25: Der Agent wählt eine passende Regel zur aktuellen Situation (Heger und Voss, 2020)

Bezüglich der Auswahl von SPT und SIMSET lässt sich aus Abbildung 24 und Abbildung 25 die folgende Erklärung ableiten: Würde der Agent FIFO oder EDD verwenden, wäre die Systemleistung deutlich schlechter. Damit ist die Auswahl

der beiden Regeln mit keinem Gewinn verknüpft und somit keine vorteilhafte Handlung im System.

In der Abbildung 26 ist die Leistung des ersten Trainings des Agenten für unterschiedliche Evaluationsszenarien (unterschiedliche Produktmix bei gleicher Einlastung) gezeigt. Zum Vergleich sind weiterhin einfache Prioritätsregeln und deren resultierende Leistung gezeigt. Es zeigt sich in Abbildung 26 (a), dass der Agent im bekannten Szenario die Leistung der statischen Regeln erreichen kann. In Abbildung 26 (b), dem unbekannten Szenario, zeigt sich, dass der Agent Handlungen wählt, die eine mittlere Leistung erbringen. Dieses Verhalten ist plausibel, da die Systemzustände für den Agenten unbekannt sind. Er muss also das Wissen aus einem anderen Szenario übertragen und Entscheidungen treffen, die mit hoher Wahrscheinlichkeit einen Mehrwert erzeugen.

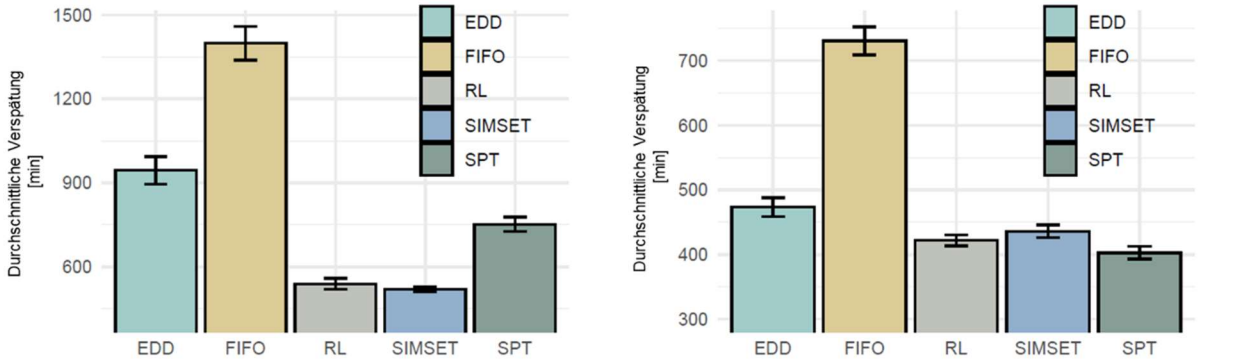


(a) Leistung des Agenten im bekannten Szenario (b) Leistung des Agenten im unbekannten Szenario

Abbildung 26: Die Leistung des Agenten aus dem ersten Training für das bekannte Szenario ist besser als die Referenzregel (Heger und Voss, 2020)

Die Evaluation des zweiten Agenten zeigt in Abbildung 27 ein ähnliches Verhalten, ist jedoch in der Lage robustere Ergebnisse zu erreichen. In diesem Fall handelt es sich um den Agenten, der auf vier verschiedene Produktmixe trainiert wurde. Zur Evaluation wurden wie bereits im ersten Vergleich, das Verhalten des Agenten gegen statische Regeln verglichen. Exemplarisch sind in Abbildung 27 zwei unterschiedliche Produktmixe bei gleicher Einlastung gezeigt. Es ist auf der linken Seite (a) zu erkennen, dass der Agent die Leistung von SIMSET erreicht, ebenso wie er im rechten Bild (b) ähnliche Leistung wie SPT erreicht. Es war dem Agenten mit dem längeren Training also möglich, eine gute Handlungsstrategie zur Auswahl von Prioritätsregeln bei unterschiedlichen Systemzuständen zu lernen. Es zeigt allerdings auch, dass die Ergebnisse nicht

so gut sind wie die Ergebnisse des ersten Agenten, der auf einem spezifischen Produktmix trainiert wurde und nur in diesem angewendet wird.



(a) Leistung für bekannten Mix 1 und Systemzustand

(b) Leistung für bekannten Mix 2 und Systemzustand

Abbildung 27: Der Agent erreicht in beiden Szenarien eine vergleichbare Leistung (Heger und Voss, 2020)

5.3.5. Zusammenfassung

Im Rahmen der komplexen Werkstattfertigung mit reihenfolgeabhängigen Rüstzeiten ist die jeweils beste Reihenfolgeregel abhängig von Systemzustand und Produktmix. Die Verwendung von RL als Methode zum Aufbau der Wissensbasis wurde entwickelt und evaluiert. Es lässt sich zusammenfassen, dass ein Agent durch das Training mit RL in der Lage ist, die Zusammenhänge im System zu erkennen und die Regeln dynamisch an die Situation anzupassen. Weiterhin zeigen die Ergebnisse, dass die gelernten Strategien zwischen Szenarien übertragen werden können, auch wenn die gewählten Handlungen dann nicht optimal sind. Die Ergebnisse zeigen, dass ein längeres Training in einem komplexeren Szenario nicht zwingend zu besseren Ergebnissen führt. Aus diesem Ergebnissen ergeben sich die folgenden Fragen; kann durch die Verwendung einer kombinierten Regel ein besseres Ergebnis erreicht werden und wie lässt sich das gelernte Wissen auf andere Szenarien übertragen ohne Leistungsverluste zu erhalten?

5.4. Methode zu Anpassung von kombinierten Prioritätsregeln

Ausgehend vom Stand der Technik wird die Verwendung von kombinierten Prioritätsregeln zur Verbesserung der Leistung empfohlen. Aus diesem Grund wird im folgenden Abschnitt die Verwendung der ATCS Regel in Kombination mit RL betrachtet. Die Regel hat bereits in verschiedenen Szenarien ihre Tauglichkeit bewiesen und ist bis zu einem gewissen Grad für den Menschen nachvollziehbar.

Weiterhin ist von der Regel bekannt, dass Sie durch die k-Faktoren individuell auf das Szenario zugeschnitten, gute Ergebnisse bringen kann.

5.4.1. Szenario einer flexiblen Fließfertigung

Das Szenario ist ebenfalls an Holthaus und Rajendran (2000) angelehnt, in diesem Fall handelt es sich allerdings um eine flexible Fließfertigung. Die Fertigung besteht aus fünf Maschinengruppen mit jeweils zwei Maschinen in einer Gruppe. Die Prozesszeit ist gleichverteilt zwischen 1 und 99 und nicht maschinenabhängig, somit werden die Maschinen als nicht identisch betrachtet. Im vorliegenden Szenario sind lediglich 3 Produktfamilien betrachtet, da die gewünschten Effekte bereits damit zu demonstrieren sind. Die Simulationsparameter wurden aus der Studie in Kapitel 5.3 übernommen. Wie auch im letzten Szenario verwenden alle Maschinen dieselbe Regel, in diesem Szenario also die ATCS-Regel mit identischen k-Faktoren.

Es ist an dieser Stelle darauf hinzuweisen, dass in diesem wie auch in dem vorherigen Szenario die Größe von Puffer vor den Maschinen einen erheblichen Einfluss auf das Verhalten des Systems hat. Andererseits ist zu beachten, dass dieser Ansatz nur funktioniert, weil zumindest einige Produkte vor den Maschinen in der Warteschlange stehen. Würden keine Produkte vor den Maschinen warten gäbe es kein Potenzial, einen besseren bzw. geeigneten Auftrag zur Verarbeitung auszuwählen. Aus diesem Grund wurde eine Vorstudie zum Thema der Pufferlänge mit der Simulation über mehrere Simulationsläufe durchgeführt. Die Vorstudien zeigt, dass bei einem geplanten Auslastungsgrad von 85 %, zwei und mehr Produkte für etwa 30 % der Simulationslaufzeit zur Bearbeitung anstehen. In seltenen Fällen stehen bis zu zehn Produkte zur Bearbeitung an. Bei einer geplanten Auslastung von etwa 90 % steigt die Anzahl der wartenden Produkte in der Warteschlange leicht an. Wie bereits in 5.1 beschreiben, könnte bei begrenzten Puffern der Fall des Blockierens eintreten, daher ist dies in diesem Beitrag nicht berücksichtigt.

5.4.2. Anpassung der ATCS-Regel mit bestärkendem Lernen

Im Gegensatz zur letzten Anwendung (Kapitel 5.3), bei dem konkreten Regeln ausgewählt wurden, können in diesem Szenario kontinuierliche Werte verwendet werden. So hat der Agent im Gegensatz zum Ansatz aus 5.3.2 die Möglichkeit die k-Faktoren schrittweise mit +/- Operatoren zu ändern. Die Auswahl der k-Faktoren geschieht nicht als diskreter Wert ($k_1 = 5$) sondern als schrittweise

Anpassung ($k_1 += 1$). Exemplarisch ist dies in Abbildung 28 zu sehen. Basierend auf der Kombination kann der Agent k_1 mit der Schrittweite von 1 zwischen 1 und 10 und k_2 mit der Schrittweite von 0.1 zwischen 0.01 und 1.01 variieren. Wie bereits beschrieben wird die Änderung der Leistung basierend auf der ausgewählten Handlung und dem jeweiligen Zustand dokumentiert und die Bewertung dieser Kombination gespeichert.

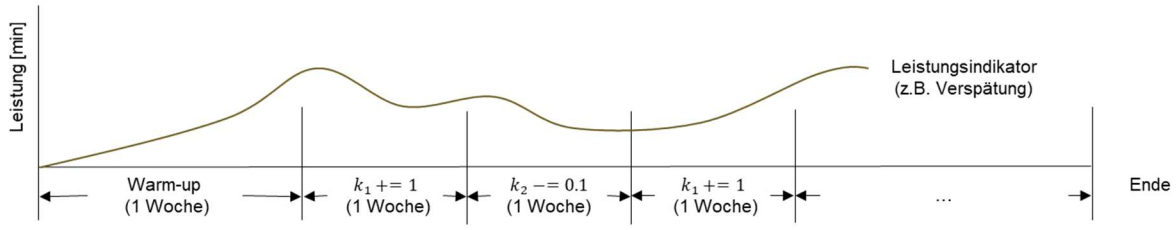


Abbildung 28: Der Agent passt die k-Faktoren schrittweise an (eigene Darstellung)

Da der Beobachtungsraum aus 5.3 ausreichend war, um ein Systemverhalten erfassen und gewinnbringende Handlungen bewerten zu können, wird dieser mit der Anpassung auf die ATCS-Regel übernommen. In Tabelle 8 sind exemplarisch die Beobachtungen des Agenten gezeigt, Änderungen gibt es lediglich bei der der Ausprägung der verwendeten Reihenfolgeregel.

Tabelle 8: Beobachtungen im System mit der ATCS-Regel

Beobachtung [Einheit]	Beispielwert
Durchschnittliche Maschinenauslastung [%]	[90, 88, 92, 89, 97, 95, 91, 90, 91, 89]
Arbeit in der Warteschlange [min]	[50, 12, 18, 51, 89, 58, 39, 55, 21, 17]
Produkt mix [%]	[50, 50, 0, 0]
Aktuell verwendete k-Faktoren	[4, 0.31]
Kurzfristige Verspätung [min]	210
Durchschnittliche Verspätung [min]	160

Da auch die Berechnung der Belohnung unabhängig von der verwendeten Regel ist und Ergebnisse gebracht hat, wird diese ebenfalls beibehalten. Es sei an dieser Stelle darauf hingewiesen, dass die ATCS-Regel bereits eine sehr gute Ausgangslage, unabhängig von den gewählten k-Faktoren bietet. Dadurch ist der Lösungsraum eingeschränkt und die Zusammenhänge zwischen k-Faktoren, Systemzustand und logistischer Leistung schwierig zu erkennen.

5.4.3. Evaluation der durchschnittlichen Verspätung

Zum generellen Verständnis wird im Folgenden das Verhalten von k-Faktoren unter verschiedenen Systemzuständen beschrieben. Weiterhin wird der Bedarf für produkt-mix spezifische k-Faktoren durch eine Parameterstudie aufgezeigt. Ferner wird, im dritten Schritt, eine detaillierte Studie durchgeführt, um für zwei spezifische Produktmixe die besten statischen k-Faktoren zu finden.

Im Folgenden ist für den Produktmix [70, 30, 0], mit einem geringen Rüstanteil, das Verhalten von unterschiedlichen k_2-Faktoren über alle k_1-Faktoren gezeigt. In Abbildung 29 ist das Verhalten bei einer Einlastung von 90 % gezeigt. Auf der x-Achse sind die k_2-Faktoren mit Schrittweite 0.1 von 0.01 bis 1.01 zu sehen. Auf der y-Achse ist die durchschnittliche Verspätung in Minuten aufgetragen. Die Punkte beschreiben den Mittelwert über 30 Replikationen mit dem Vertrauensintervall. Es ist offensichtlich, dass ein kleiner k_2-Faktor, der eine Reihenfolgeregelung mit Rüstvermeidung beschreibt, zu einer besseren Leistung führt.

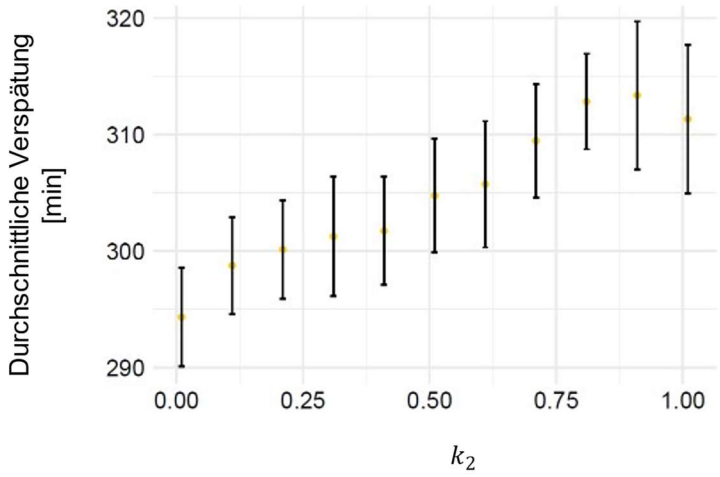


Abbildung 29: Kleine k_2-Werte führen bei einer hohen Auslastung zu einer besseren Leistung (Heger und Voss, 2021)

Im Gegensatz dazu ist in Abbildung 30 die Leistung des Systems bei niedriger Einlastung (85 %) gezeigt. Auf der x-Achse sind die k_2-Faktoren und auf der y-Achse ist die durchschnittliche Verspätung in Minuten aufgetragen. Die Punkte beschreiben den Mittelwert über 30 Replikationen mit dem Vertrauensintervall. Es ist klar zu erkennen, dass im Gegensatz zu Abbildung 29 kleine k_2-Werte zu einer schlechteren Systemleistung führen und große k_2-Faktoren vorteilhaft für das System sind.

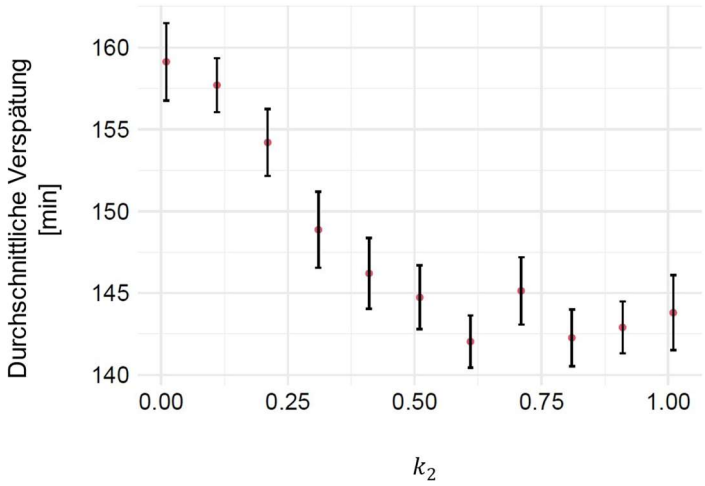


Abbildung 30: Die Verwendung von kleinen k_2-Werten bei niedriger Auslastung für zu einer besseren Leistung (Heger und Voss, 2021)

Um dieses Verhalten über mehrere Produktmixe hinweg nachzuweisen wurden 66 unterschiedliche Produktmixe überprüft und die beste Kombination aus k-Faktoren für die zwei unterschiedlichen Einlastungen dokumentiert. In Abbildung 31 beschreiben die beiden großen Spalten die Einlastung von 90 und 85 %. Jede der beiden Spalten ist in fünf Zeilen aufgeteilt, die die Häufigkeit des jeweils besten k_1-Faktors in Kombination mit dem besten k_2-Faktor beschreiben. Die rechte obere Ecke kann also gelesen werden als: Bei einer 85 % Einlastung war die Kombination von [1, 1.01] fünfmal die beste Kombination. Weiterhin war die Kombination aus [1, 0.91] viermal die beste Kombination bei der gegebenen Einlastung. Die zweite Zeile wird gelesen als: bei einer Einlastung von 85 % hat die Kombination [3, 0.91] dreimal die beste Leistung erbracht, usw. Die Summe der Einlastungsspalten bilden die 66 unterschiedlichen Produktmixe ab. Es ist klar zu erkennen, dass bei einer 90 % Einlastung die Häufigkeit der niedrigen k_2-Faktoren, die zur besten Leistung geführt haben, deutlich zunimmt. Weiterhin fällt auf, dass mit kleineren k_2-Faktoren häufig auch größere k_1-Faktoren einhergehen. Ausgehend von der Häufigkeit wären für die 66 Produktmixe bei 90 % Einlastung die Kombination von [9, 0.01] und für 85 % die Kombination [1, 0.91] eine gute Wahl, um die Leistung zu maximieren. Entscheidend ist hierbei, dass sich diese Werte von den individuellen Werten für die Produktmixe deutlich unterscheiden können. So kann für den Produktmix [70, 30, 0] bei einer 90 % Einlastung festgestellt werde, dass $k_2 = 0.01$ die beste Leistung erbringt, wobei bei einer 85 % Einlastung, $k_2 = 0.61$ die beste Leistung erbringt.

Es zeigt sich, dass die produktmixspezifischen k-Faktoren abhängig vom Systemzustand notwendig sind, um gute Leistung zu erhalten. Aus diesem Grund wurden mehrere Produktmixe zusätzlich noch genauer untersucht, um das Verhalten beschreiben zu können. Aus den Abbildungen 30 bis 32 kann gefolgert werden, dass die Reduktion des k_2-Faktors für viele Produktmixe zwischen 85 – und 90 % Auslastung liegt.

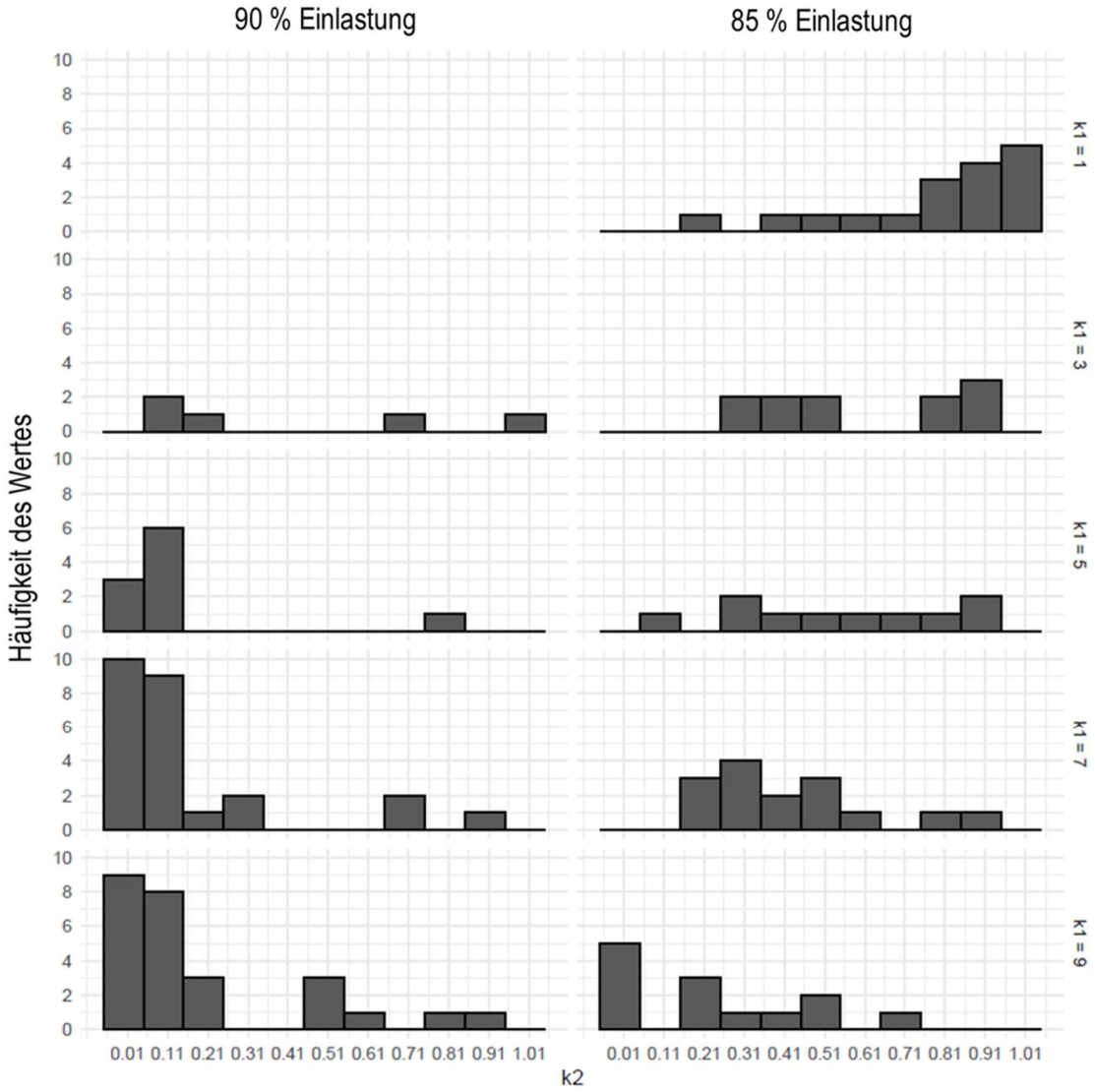


Abbildung 31: Die Häufigkeit der besten k-Faktoren für 66 Produktmixe (Heger und Voss, 2021)

So ist in Abbildung 32 eine detailliertere Analyse bzgl. des Verhaltens der durchschnittlichen Verspätung für einen Produktmix bei verschiedenen k_2-Faktoren unter steigender Auslastung detaillierter zu erkennen. Im Gegensatz zur ersten Parameterstudie wurde die Auflösung der Zwischenankunftszeit feiner und die so beobachtete Auslastung genauer analysiert. Es zeigt sich eindeutig, dass ein schrittweises Reduzieren des k_2-Faktors bei steigender Auslastung

durchschnittlich positiv auf die Leistung auswirkt. Aufgrund der benötigten Rechenleistung können nicht für alle Produktmixe vergleichbare und ausführliche Studien durchgeführt werden. Deshalb wurden im Rahmen der Arbeit lediglich eine kleine Menge an Produktmixen in dem beschriebenen Detailgrad betrachtet.

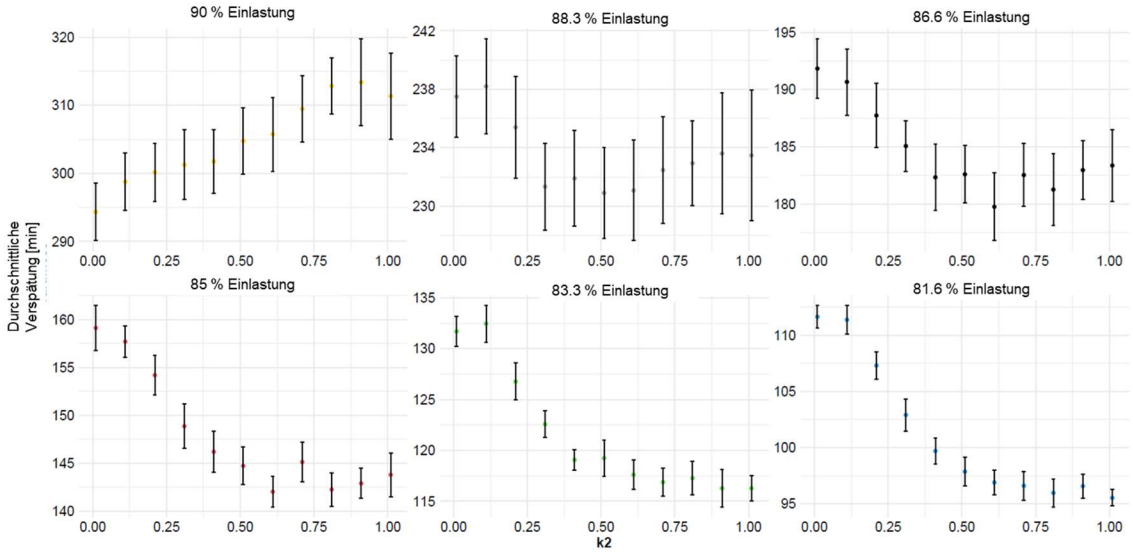


Abbildung 32: Detaillierte Analyse der k_2-Faktoren bei steigender Auslastung für einen Produktmix (eigene Darstellung)

Im Rahmen der Studie wurden neben der Zwischenankunftszeit auch die k_1-Faktoren mit einer kleineren Schrittweite betrachtet. Exemplarisch sind zwei der getesteten Produktmixe in Tabelle 9 zusammengefasst. Analog zum Vorgehen in 5.2.2 sind in der Tabelle die k-Faktoren angegeben, die zur niedrigsten durchschnittlichen Verspätung geführt haben. Der Mix auf der rechten Seite [0, 50, 50], welcher deutlich größere Rüstanteile hat, zeigt, dass bei gleicher Einlastung von 85 % unterschiedliche k-Faktoren zur niedrigsten durchschnittlichen Verspätung führten.

Tabelle 9: Betrachtung der besten k-Faktoren für zwei ausgewählte Produktmixe

	Mix [70, 30, 0]			Mix [0, 50, 50]		
$Einlastung_{plan}$	k_1	k_2	Auslastung [%]	k_1	k_2	Auslastung [%]
	1	1.01	85.6	1	0.61	87.6
	1	1.01	87.1	2	0.51	88.9
85 %	1	0.61	88.4	3	0.01	89.7
	2	0.61	89.9	6	0.11	91.1
	6	0.41	91.3	7	0.11	92.5
90 %	9	0.01	92.3	8	0.11	93.8

Die detaillierte Betrachtung macht ein weiteres Mal deutlich, dass produktmixspezifische k-Faktoren passend zum Systemzustand notwendig sind, um die Leistung zu optimieren. Der in Tabelle 9 beschriebene Datensatz wird im Folgenden verwendet, um die k-Faktoren dynamisch an die Systemzustände anzupassen und einen Vergleichswert für den RL-Agenten zu stellen. Auf Grund der benötigten Rechenzeit und aufwendigen Simulationsstudie wird dieser Datensatz im Folgenden auch als Brute Force Methode bezeichnet.

5.4.4. Evaluation der durchschnittlichen Durchlaufzeit

Um die Übertragbarkeit auf ein weiteres Zielkriterium zu prüfen, wurden dieselben Versuche mit dem Kriterium Durchlaufzeit durchgeführt. Wie bereits beschrieben, umfasst die Durchlaufzeit in der Fertigung die Rüstzeit, die Bearbeitungszeit pro Einheit und zusätzliche Zeiten, während derer die unfertige Ware zwischen den Arbeitsstationen bewegt wird oder auf die Bearbeitung wartet. Bei der Optimierung der durchschnittlichen Durchlaufzeit würde die Betrachtung von Rüstzeiten in den Hintergrund rücken und ein stärkerer Fokus auf die Prozesszeit gelegt werden. So wird die Verwendung von SPT als Reihenfolge zur Optimierung des Durchsatzes betrachtet. Da die ATCS-Regel aus drei Terminalen besteht (vergleiche Formel (2)), von denen einer die Bearbeitungszeit berücksichtigt, scheint die Verwendung der ATCS-Regel angemessen.

Ähnlich dem Verfahren zur Suche der optimaler k-Faktoren für die Reduzierung der mittleren Verspätung, können die besten k-Faktoren zur Reduzierung der mittleren Durchlaufzeit auf ähnliche Weise gefunden werden. Mit der Parameterkonfiguration des zweiten Experiments (vgl. Tabelle 9) wurde die Studie für den neuen Leistungsindikator wiederholt. Im Folgenden werden die besten k-Faktoren unter Berücksichtigung der mittleren Verspätung und der durchschnittlichen Durchlaufzeit für den Produktmix [70, 30, 0] verglichen. Wie in

Tabelle 10 zu erkennen, ist das allgemeine Muster ähnlich, dennoch unterscheiden sich die genauen Werte aufgrund der unterschiedlichen Zielsetzung leicht. Es ist davon auszugehen, dass der Agent in der Lage ist, unter Anpassung der Belohnungsfunktion, diese Zusammenhänge ebenfalls zu lernen.

Tabelle 10: Vergleich der k-Werte für denselben Produktmix unter der Betrachtung unterschiedlicher Leistungsindikatoren

$Einlastung_{plan}$	k_1	k_2	Durchlaufzeit [min]	k_1	k_2	Verspätung [min]
90 %	6	0.11	1278	5	0.01	299
	5	0.41	1216	4	0.41	232
	7	0.31	1160	1	0.61	181
85 %	9	0.41	1119	1	0.61	143
	4	0.71	1085	1	0.91	116
	1	0.61	1059	1	0.81	96
	1	1.01	1036	1	1.01	80

Es ist zu erkennen, dass zwischen Verspätung und Durchlaufzeit Szenario-bedingt eine gewisse Beziehung besteht. Ein entscheidender Faktor ist an dieser Stelle die Gleichverteilung der Prozesszeiten zwischen den Produktfamilien.

5.4.5. Evaluation der Anpassung mit bestärkendem Lernen

Im nächsten Schritt wird geprüft, ob der Agent in der Lage ist die Auswirkungen der k-Faktoren auf die Leistung zu nutzten. Wie auch in 5.3.4 wurde der Agent mit 70 Schritten pro Simulationslauf trainiert. Da aus dem letzten Abschnitt bereits bekannt ist, dass der Agent auf mehrere Produktmixe nur reagieren kann, wenn er sie im Training gesehen hat, wurde dies bereits berücksichtigt. Ausgehend von einem erfolgreichen Training mit 200 Episoden für einen einzelnen Produktmix in einer Vorstudie wurden dem finalen Agenten während des Trainings sechs unterschiedliche Produktmixe gezeigt und 1000 Episoden als Trainingsdauer festgelegt. Der Trainingsdatensatz umfasst also 70000 Datenpunkte. Im Training wurden die Belohnungen (wie in 5.3.4) basierend auf einer Woche berechnet. Durch die Verwendung der ATCS-Regel und die Entscheidung die k-Faktoren schrittweise anzupassen, benötigte der Agent 9 Wochen (~15 % der Simulationsdauer) um vom größten zum kleinsten Wert k-Faktor zu kommen. Aus diesem Grund wurden in der Evaluation zwei Möglichkeiten betrachtet, eine Handlung jeden Tag oder eine Handlung jede Woche. Da die Handlung im Live-System basierend auf den Vorhersagen des Modells lediglich abhängig von den Beobachtungen des Systemzustandes getroffen werden, ist diese Anpassung

problemlos möglich. Durch die tägliche Anpassung war es dem Agenten möglich, schneller auf dynamisch auftretende Änderungen im System zu reagieren und innerhalb von etwas mehr als einer Woche die komplette Spanne der k-Faktoren abzudecken. Für die Auswertung der Leistung des Agenten wurde schlussendlich die tägliche Anpassung gewählt.

In Abbildung 33 ist das Verhalten des RL-Agenten über die Laufzeit der Simulation zu sehen. Die x-Achse beschreibt die Laufzeit der Simulation in Minuten. Auf der linken Seite der Abbildung ist auf der y-Achse die durchschnittliche Verspätung in Minuten abgebildet, auf der rechten Seite auf der y-Achse ist der aktuelle k_2-Faktor gezeigt. Es ist klar zu erkennen, dass der Agent für den Produktmix eine hohen k_2-Faktor wählt und, wenn die durchschnittliche Verspätung steigt, der k_2-Faktor durch den Agenten später wieder reduziert wird. Dies ist besonders vor dem Hintergrund interessant, dass es sich um ein stabiles System mit konstanter Auslastung handelt. Die gepunktete Linie beschreibt die durchschnittliche Verspätung der letzten 200 Aufträge und demonstriert anschaulich den Effekt der stochastisch verteilten Zwischenankunftszeit und die schwankende Prozesszeit.

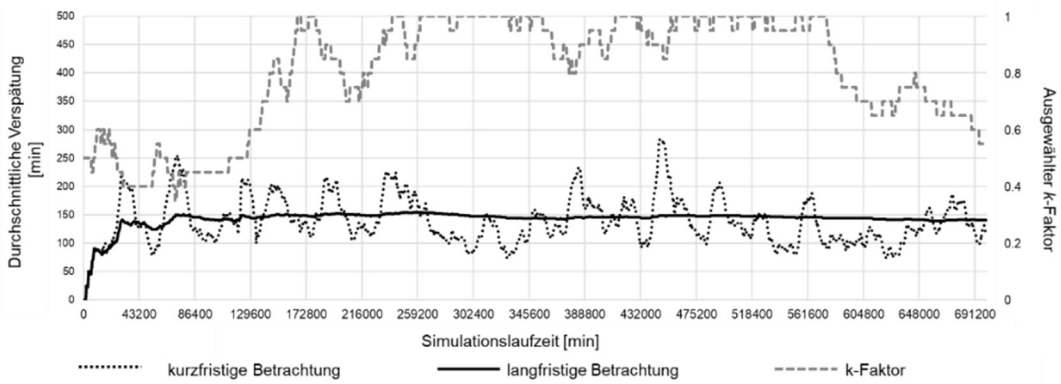


Abbildung 33:Der Agent passt den k_2-Faktor dynamisch an die Situation an (Heger und Voss, 2021)

Wird dieses Verhalten über 30 Replikationen betrachtet zeigt sich, dass der Agent in der Lage ist, die Leistung des Systems mit individuellen und statischen k-Faktoren zu schlagen. Weiterhin zeigt sich, dass die einfachen Prioritätsregeln wie SPT signifikant geschlagen werden. In diesem Szenario wird die in Abschnitt 5.4.3, mit Hilfe der Parameterstudie erzeugten Tabelle, ebenfalls geschlagen. Hierbei ist zu beachten, dass sowohl der Produktmix wie auch die genaue Einlastung des Systems bekannt sind und die k-Faktoren durch Simulation ausführlich evaluiert wurden. Die Verwendung von statischen Faktoren wie sie in Abbildung 31 gezeigt werden, würde hier ebenfalls unzureichende Ergebnisse bringen.

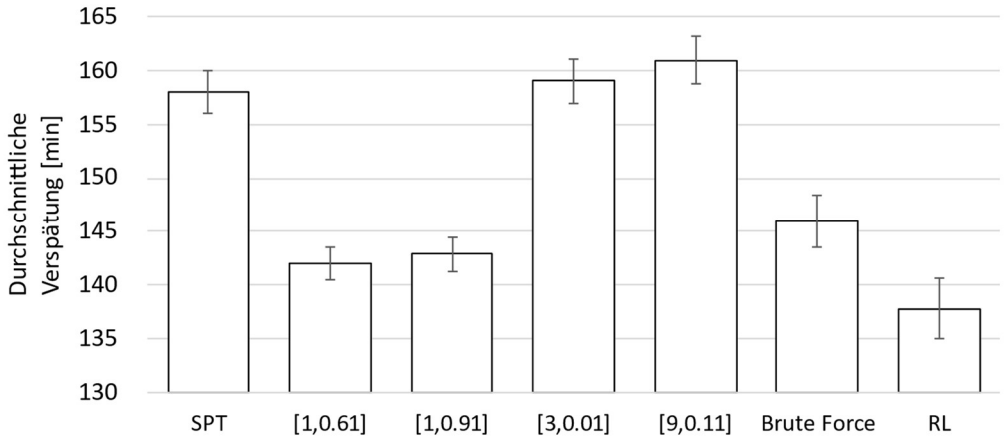


Abbildung 34: Der Agent bringt bessere Leistungen als die statischen k-Faktoren (Heger und Voss, 2021)

Analog zum vorherigen Vorgehen wird der Agent erneut in zwei verschiedenen Szenarien evaluiert. Im ersten Szenario ist der Produktmix über die komplette Laufzeit der Simulation bekannt. Im zweiten Teil der Evaluation ändert sich der Produktmix nach einer Zeit. So kann geprüft werden, inwieweit der Agent auf unbekannten Systemzustände reagiert.

Im nächsten Szenario (Abbildung 35) ist ein Produktmixwechsel nach einem ¾ Jahr zu erkennen. Durch die Verwendung von unpassenden k-Faktoren steigt die durchschnittliche Verspätung über einen sehr kurzen Zeitraum stark an. Der Agent beobachtet den Anstieg wie auch den geänderten Produktmix in System und kompensiert durch die Anpassung der Werte. Es fällt auf, dass der Anfang der Laufzeit von Abbildung 35 vergleichbar mit dem Beginn der Laufzeit von Abbildung 33 ist, es ist also davon auszugehen, dass dieses Verhalten typisch für den Produktmix ist. Weiterhin zeigt Abbildung 35 deutlich die Reduktion des k_2-Faktors als der Produktmixwechsel stattfindet. Nach dem Produktmix-Wechsel bleibt der k_2-Faktor auf mittlerem Niveau, da ein erneutes Anheben negative Auswirkungen auf die Leistung hätte.

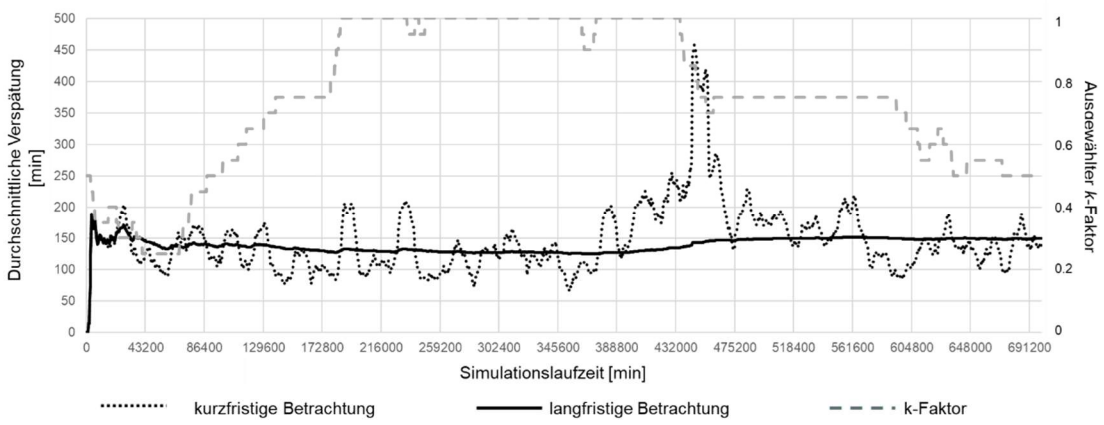


Abbildung 35: Der Agent erkennt den Produktmix-Wechsel passt den k_2-Faktor an (Heger und Voss, 2021)

Es zeigt sich, dass der Agent die dynamische Anpassung über 30 Replikationen sicher durchführen und die Leistung von statischen k-Faktoren sowie die dynamische Anpassung mit Hilfe der vorgelagerten Simulationsstudie schlagen kann.

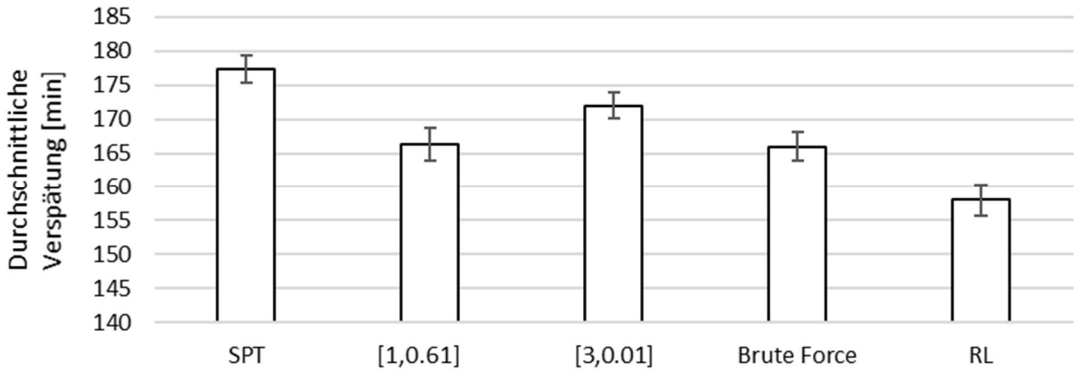


Abbildung 36: Der Agent reduziert die durchschnittliche Verspätung um bis zu 5 % (Heger und Voss, 2021)

Weiterhin wurde geprüft, wie der Agent auf eine Kombination aus einer unbekannten Verteilung der Zwischenankunftszeit und Wechsel im Produktmix reagiert. Besonders interessant ist dabei die Frage, wie die Auswirkungen der geänderten Verteilung kompensiert werden. In Abbildung 37 ist das Verhalten über die Zeit dargestellt.

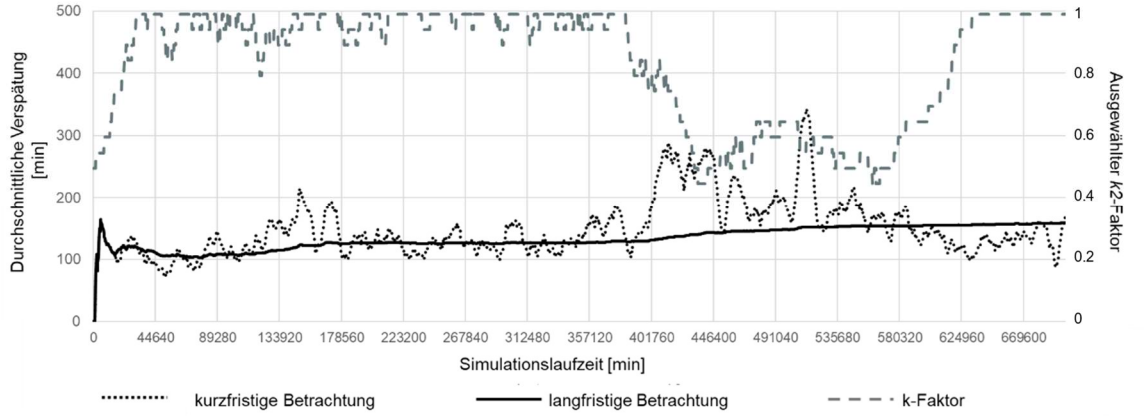


Abbildung 37: Verhalten des Agenten bei unbekannter Zwischenankunftszeit (eigene Darstellung)

Im Vergleich zu Abbildung 35 fällt auf, dass der Agent in Abbildung 37 deutlich schneller einen hohen k_2-Faktor anstrebt und diesen deutlich häufiger anpasst. Weiterhin ist er in der Lage den Produktmix-Wechsel zu erkennen und die k-Faktoren entsprechend anzupassen. Im Gegensatz zum bekannten Szenario wählt er zum Ende der Simulation allerdings einen hohen k_2-Faktor. Dies wirkt sich entsprechend auf die Leistung des Systems aus, wie in Abbildung 38 zu erkennen ist. Der Agent ist immer noch in der Lage die Leistung der besten k-Faktoren zu erreichen, allerdings schlägt er diese nicht mehr. Es ist an dieser Stelle darauf hinzuweisen, dass der Agent weder die Verteilung des Zwischenankunftszeit noch den Produktmixwechsel kannte. Somit lässt sich feststellen, dass der Agent in einem unbekannten Szenario immer noch in der Lage ist mit den besten statischen Werten mitzuhalten.

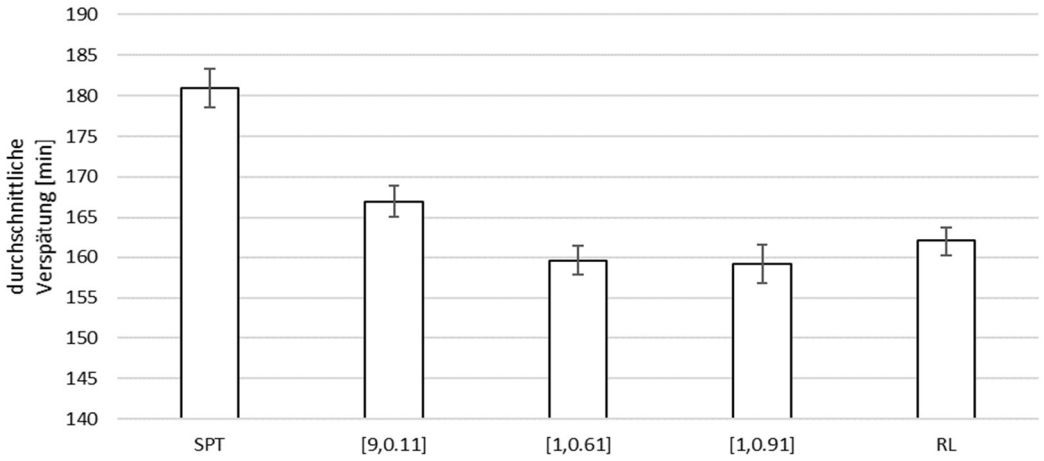


Abbildung 38: Leistung des RL-Agenten in einem unbekannten Szenario (eigene Darstellung)

5.4.6. Zusammenfassung

Die Verwendung der ATCS-Regel in einer flexiblen Fließfertigung mit reihenfolgeabhängigen Rüstzeiten benötigt produktmixspezifische k-Faktoren je nach Systemzustand. Das Finden dieser, mit Hilfe von Simulation, ist für bekannte Szenarien möglich. Ein Abschätzen der bestmöglichen k-Faktoren für unbekannte Systemzustände ist mit Hilfe von RL in diesem Szenario möglich. So ist die dynamische Anpassung der k-Faktoren als Methode zur dynamischen Anpassung, trainiert mit RL, ist in der Lage die logistische Leistung des Systems unter bekannten und unbekannten Systemzuständen zu verbessern.

Der Agent ist in der Lage die komplexen Zusammenhänge des Systems zu erfassen und zu nutzten. Vor allem in Szenarien, in denen die besten k-Faktoren auf Grund von dynamischen Änderungen variieren, zeigt die neue Methode gute Leistung. Die Verwendung des Agenten demonstriert eine gute dynamische Anpassung, die nachvollziehbar auf den Systemzustand reagiert und dabei Leistungen vergleichbar mit den optimalen k-Faktoren erreicht. Weiterhin demonstriert die Verwendung des RL in diesen Zusammenhang eine gute Leistung von zentral festgelegten Reihenfolgeregeln, die dann dezentral ausgeführt werden.

5.5. Vergleich unterschiedlicher Lernverfahren zur dynamischen Anpassung

Im Rahmen des Kapitels 3.3 wurden mehrere Verfahren zur Vorhersage von Systemverhalten und zur dynamischen Anpassung von Reihenfolgeregeln aufgeführt. Neben dem im Kapitel 5.3 und 5.4 beschriebenen Verwendung von RL wurden ebenfalls die Ansätze zur Verwendung von DT und NN beschrieben. Zum Vergleich der Eignung der unterschiedlichen Methoden für den in dieser Arbeit gezeigten Anwendungsfall werden die Methoden im folgenden Szenario spezifisch entwickelt und ihre Leistung über mehrere Szenarien hinweg verglichen.

Alle drei Methoden werden im Rahmen des Fertigungssystems mit reihenfolgeabhängigen Rüstzeiten, bekannt aus Kapitel 5.3, und der Verwendung von ATCS als Reihenfolgeregel (siehe Kapitel 5.4) getestet. Im Folgenden (Tabelle 11) ist das Szenario noch einmal detailliert beschrieben:

Tabelle 11: Szenario der flexiblen Fließfertigung mit 10 Maschinen

System	Maschinen: 10
	Maschinengruppen: 5
	Organisationsform: Fließfertigung
Job	Produktfamilien: 4
	Verteilung der Produktfamilien: nach Produktmix
	Operationen pro Auftrag: 10
	Verteilung der Zwischenankunftszeit: Poisson
	Prozessbearbeitungszeit: 1 – 99
	Verteilung der Prozesszeit: gleichverteilt
	Fälligkeitstermin: TWK Methode
Simulation	Einschwingphase: 2500 Aufträge
	Simulationsdauer: 12500 Aufträge
	Replikationen: 5
KPIs	Durchschnittliche Verspätung
	Durchschnittliche Durchlaufzeit

Wie bereits aus den vorherigen Kapiteln bekannt, sind die reihenfolgeabhängigen Rüstzeiten sowie die durchschnittliche Einlastung im System von entscheidender Bedeutung für die Leistung der ATCS-Regel. Die Betrachtung von 4 Produktfamilien in einem individuellen Produktmix führt zu unterschiedlich großen Rüstzeitanteilen. So ist davon auszugehen, dass ein Produktmix, der lediglich die ersten drei Produktfamilien enthält, deutlich weniger Rüstzeit im Mittel benötigt als ein Produktmix, der alle vier Produktfamilien enthält. Die Rüstzeitmatrix ist in Matrix (7) gezeigt, die bereits in Kapitel 5.3 beschrieben wurde. Weiterhin wurde für diese Studie der DueDate-Faktor von zwei auf drei erhöht, um mehr Zeit für die Fertigstellung einzuräumen.

Analog zu Kapitel 5.4 wurden als Beobachtungen die durchschnittliche Verspätung, die durchschnittliche Durchlaufzeit sowie die durchschnittliche Maschinenauslastung des Szenarios zusammen mit dem Produktmix und die jeweils aktuellen k-Faktoren dokumentiert. Dabei sind die k-Faktoren, die Auslastung sowie die Leistungsindikatoren kontinuierliche Variablen. Der Produktmix wird als kategorische Variable betrachtet. In Tabelle 12 ist eine mögliche Konfiguration des Systemzustandes gegeben.

Tabelle 12: Beispiel für die Beschreibung des Systemzustandes.

Features / Beobachtung [Ausprägung]	Beispielwert
Produktmix [kategorisch]	[5]
k-Faktoren [kontinuierlich]	[3, 0.81]
Maschinenauslastung [kontinuierlich]	[0.85]

Leistungskennwert [Ausprägung]	Beispielwert
Durchschnittliche Verspätung [kontinuierlich]	152

Wie bereits in den vorherigen Analysen wird die durchschnittliche Verspätung sowie die durchschnittliche Durchlaufzeit dokumentiert und zur Beurteilung der Leistung betrachtet.

5.5.1. Entwicklung der drei Regressionsmodelle

Analog zur Parameterstudie in 5.4 wurden in diesem Fall allen Kombinationen aus k_1-Werten von 1 bis 10 in Schrittweite 1 und k_2-Werten von 0.01 bis 1.01 mit Schrittweite 0.1 sowie 7 verschiedenen Einlastungen von 85 % bis 95 % und 12 verschiedenen Produktmixen mit unterschiedlichen Rüstanteilen durchgeführt. Für die so entstandenen 9240 individuellen Parameterkombinationen wurden jeweils 5 Replikationen durchgeführt, womit der initiale Datensatz 46200 Samples umfasst. Zusätzlich wurde aus dem ersten Datensatz mit 46200 Datenpunkte ein zweiter Datensatz mit 13860 Datenpunkte (entspricht 30 % des ersten Datensatzes) zufällig gezogen. Das Training der Regressionsmodelle (NN ebenso wie DT) wurde auf beiden Sätzen durchgeführt, um eine Aussage über die Verbesserung der Präzision mit mehr Datenpunkten machen zu können. Ausgehend von identischen Rohdaten wurden für das Training der NNs jedoch methodenspezifische Datenvorverarbeitung durchgeführt, um die Ergebnisse zu verbessern. Zusätzlich wurde eine gute Konfiguration für die NN sowie die DT mit Hilfe eines Grid-Search-Verfahrens ermittelt.

Neuronales Netz zur Vorhersage der Leistung

Das NN wurde mit scikit learn als Multi Layer Perceptron in Python implementiert. Das resultierende, zweilagige Netz mit 10 Neuronen in der ersten und 30 Neuronen in der zweiten Schicht hatte die Aktivierungsfunktion „relu“. In Kombination mit dem Solver „adam“ zeigte eine Minibatchgröße von 500 Samples gute Ergebnisse. Die initiale Lernrate wurde mit 0.01 festgelegt. Eine L2-Regulierung wurde durchgeführt. In Tabelle 13 sind die möglichen Konfigurationen für das Grid-Search-Verfahren gezeigt, wobei die endgültige Konfiguration in **fett** gedruckt ist.

Tabelle 13: Parameterkonfiguration des neuronalen Netztes

Parameter	Konfiguration
Mini Batch Size	[100, 200, 250, **500**, 50, 60]
Hidden layer sizes	[(4), (6), (10), (12), (15), (16), (100), (5, 10), (8, 12), (10, 5), (10, 20), (**10, 30**), (12, 16), (16, 32), (32, 32), (50, 75), (50, 50), (64, 32), (64, 128), (128, 64), (75, 50), (300, 300)]
Learning rate init	[1, 0.05, 0.1, **0.01**, 0.001]
Alpha	[0.001, 0.005, 0.01, **1**, 10, 0.1]

Für eine bessere Leistung im Training der NN wurden die k-Faktoren auf Werte von -1 bis 1 standardisiert. Weiterhin wurde die kategorische Ausprägung des Produktmixes als One-Hot-Encoding an das NN übergeben um diese als binäre Variable betrachten zu können. Bei diesem Verfahren werden die unterschiedlichen Ausprägungen von kategorischen Variablen binär als zusätzliche Beobachtungen kodiert – bei 11 verschiedenem Mixen wird aus Produktmix 5 der Vektor [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]. Somit ergeben sich als Übergabe für die Beobachtungen aus Tabelle 12 die folgenden Werte in Tabelle 14. Weiterhin sind die Leistungsindikatoren als Zielwert nicht mehr in den Beobachtungen enthalten:

Tabelle 14: Vorverarbeitete Beobachtungen für das neuronale Netz.

Features / Beobachtung [Ausprägung]	Beispielwert
Produktmix [binär]	[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
k-Faktoren [kontinuierlich]	[-0.87, 0.94]
Maschinenauslastung [kontinuierlich]	[0.85]

Entscheidungsbäume zur Vorhersage der Leistung

Die Entscheidungsbäume (engl. Decission Tree – DT) wurden mit scikit learn als Entscheidungsbaum Regressor in Python implementiert. Die endgültige Konfiguration wurde auch hier mit Hilfe eines Grid-Search-Verfahrens ermittelt. In diesem Rahmen zeigte sich (siehe Tabelle 15), dass die Verwendung einer maximalen Tiefe von 5, mit mindestens 4 Samples pro Blatt zu guten Ergebnissen führte.

Tabelle 15: Parameterkonfiguration für die Entscheidungsbäume

Parameter	Konfiguration
min_samples_split	[**4**, 5, 10, 20, 40]
max_depth	[1, 2, 3, 4, **5**, 6, 7, 8, 9, 10, None]
min_samples_leaf	[**4**, 5, 10, 20, 40, 100]
max_leaf_nodes	[**100**, 150, 200, 300, 400]

Da die Entscheidungsbäume auf Grund ihrer Struktur weniger Probleme mit kategorischen Variablen haben, wurden die Beobachtungen an dieser Stelle nicht vorverarbeitet und direkt für das Training verwendet.

Bestärkendes Lernen zur Vorhersage der Leistung

Wie bereits in Kapitel 5.4 gezeigt wurde der RL-Agent mit Hilfe der Pathmind Software-as-a-Service-Plattform mit 12000 Simulationsläufen innerhalb von 12 Stunden trainiert. Dabei wurden verschiedene Hyperparameterkonfigurationen im Rahmen von populationsbasiertem Training automatisch evaluiert und die beste Konfiguration für das Szenario gefunden. Pathmind verwendet Ray und

RLlib für das Training des Agenten. Die Strategie des Agenten wurde als Proximal Policy Optimization trainiert.

5.5.2. Evaluation der Regressionsmethoden in mehreren Szenarien

Analog zur Evaluation in 5.3 und 5.4 werden im Folgenden erst die Auswirkungen der k_2-Faktoren auf das Systemverhalten in einem statischen Szenario beschrieben. Anschließend werden die drei Methoden zur dynamischen Anpassung in der Onlineverwendung in einem statischen und einem dynamischen Szenario getestet und miteinander verglichen.

Evaluation der k-Faktoren und des Systemverhaltens

Für die erste Evaluation wird für ein bekanntes Szenario bei statischer Einlastung und bekanntem Produktmix die Leistung unter Veränderung der k-Faktoren dokumentiert. Vergleichbar mit Abbildung 29 und Abbildung 30 sind im Folgenden, für eine Einlastung von 85 %, zwei unterschiedliche Produktmix mit unterschiedlichen Rüstzeitanteilen zu erkennen (Abbildung 39).

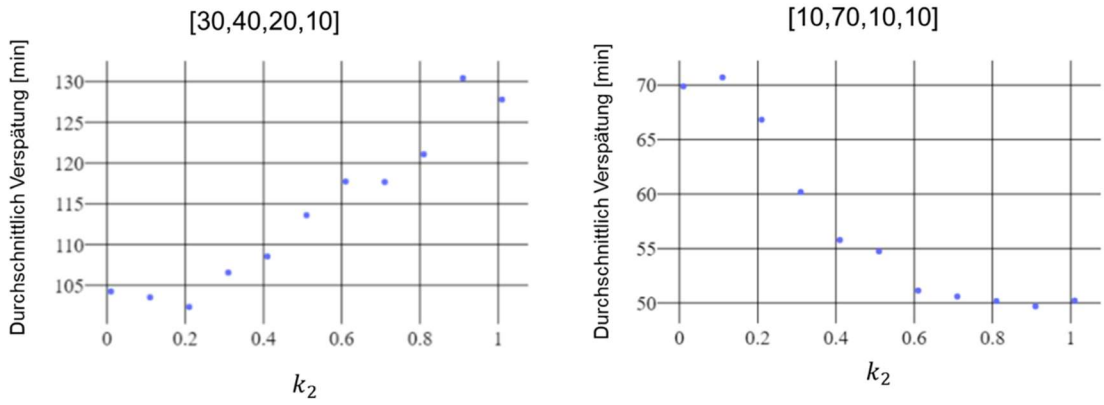


Abbildung 39: Bei gleicher Einlastung und gleichem k-Faktor zeigen die beiden Produktmixe unterschiedliche Leistung (Heger et al., 2021)

In den aggregieren Rohdaten (Abbildung 39) ist klar zu erkennen, dass unterschiedliche Produktmixe bei derselben Einlastung und der Verwendung von statischen k-Faktoren unterschiedliche Leistungen bringen. Ein kleiner k_2-Wert, welcher vorteilhaft für Produktmix [30, 40, 20, 10] ist, würde bei Produktmix [10, 70, 10, 10] zu einer Verschlechterung der Leistung um 30 % führen. Es kann festgestellt werden, dass die durchschnittliche Verspätung für Produktmix [10, 70, 10, 10] sehr gering ist. Eine Einlastung von 85 % und weniger würde dazu führen, dass alle k-Faktoren gleich wären und kein Verbesserungspotenzial in der dynamischen Anpassung mehr möglich wäre. Diesen Sachverhalt gilt es über mehrere Produktmixe zu erkennen und zur Verbesserung der Leistung zu nutzen.

Evaluation der dynamischen Anpassung im statischen Szenario

Wie bereits aus Kapitel 5.4.5 und Abbildung 33 bekannt, ist in Abbildung 40 das Verhalten des RL-Agenten für ein Szenario mit kleinem k_2 gezeigt. Wie bereits vorab erwähnt wurde und in Abbildung 39 erkenntlich war, ist für den Produktmix [30, 40, 20, 10] ein niedriger k_2-Wert von Vorteil. Es ist zu erkennen, dass bei einer Reduktion der durchschnittlichen Verspätung (linke Y-Achse) über die Zeit (X-Achse) die verwendeten k_2-Werte der ATCS-Regel (rechte Y-Achse) erhöht wird. Im Gegenzug fällt auf, dass bei steigender durchschnittlicher Verspätung, der k_2-Wert gesenkt wird.

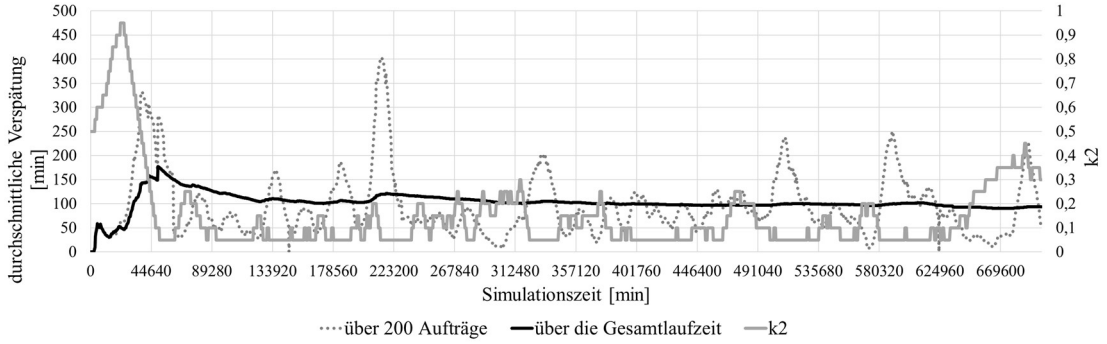


Abbildung 40: Dynamische Anpassung des niedrigen k_2-Faktors mit bestärkendem Lernen (Heger et al., 2021)

In der ausführlichen Evaluation über 30 Replikationen zeigt sich, dass bei einer statischen Einlastung und bei einem bekannten Produktmix die dynamische Anpassung der k-Faktoren mit RL einen positiven Einfluss hat, sich aber nicht signifikant zur statischen Auswahl von k-Faktoren unterscheidet. Die DTs, welche besonders gut geeignet sind, um bekannte Sachverhalte wiederzugeben erreichen ebenso wie das RL vergleichbare Werte mit den niedrigen statischen k-Faktoren. In der Abbildung 41 fällt auf, dass die NNs nicht in der Lage sind, die Leistung der anderen Verfahren zu erreichen. Es ist also davon auszugehen, dass das dynamische Umschalten und Anpassen durch das NN in diesem statischen Szenario einen negativen Einfluss auf die Leistung hat (vgl. Priore et al., 2006).

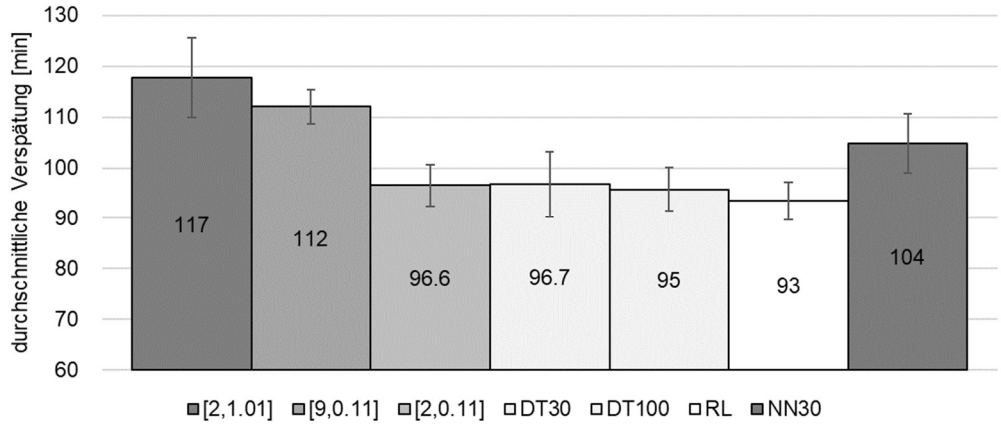


Abbildung 41: Vergleich der ML-Methoden zur dynamischen Parameteranpassung in einem statischen Szenario (Heger et al., 2021)

Evaluation der dynamischen Anpassung im dynamischen Szenario

Im Gegensatz zum statischen Szenario können die NNs ihre Vorteile im zweiten Szenario demonstrieren. Während der Simulation wird für ¼ der betrachteten Zeit ein neuer Produktmix (in diesem Fall Produktmix [30, 40, 20, 10] von oben) im System benötigt. Dieses Verhalten ist exemplarisch an das Saisongeschäft (z.B. Weihnachtszeit) angelehnt und bereits aus den vorherigen Simulationsstudien bekannt. Die Evaluation des dynamischen Szenarios (siehe Abbildung 42) zeigt, dass die Auswahl, der vorab ausgewählten und guten statischen k-Faktoren bereits zu robuster Leistung führt. Es ist ebenso zu erkennen, dass alternative k-Faktoren, die vorab weniger gute Leistung erzielten, im neuen Szenario bessere Ergebnisse bringen können, in diesem Fall die Kombination [2, 1.01].

Im Vergleich dazu bringen die DTs, die NNs ebenso wie der RL-Agent eine zusätzliche signifikante Verbesserung von bis zu 15 %. Dabei ist der RL-Agent zusätzlich 3 % besser als die Verwendung der DTs. Im Gegensatz zum statischen Szenario kann das NN seine Vorteile bzgl. der Generalisierbarkeit von Verhalten zeigen. Die vergleichbare Leistung von RL und NN ist nachvollziehbar, da RL zur Schätzung der Belohnung NNs verwendet.

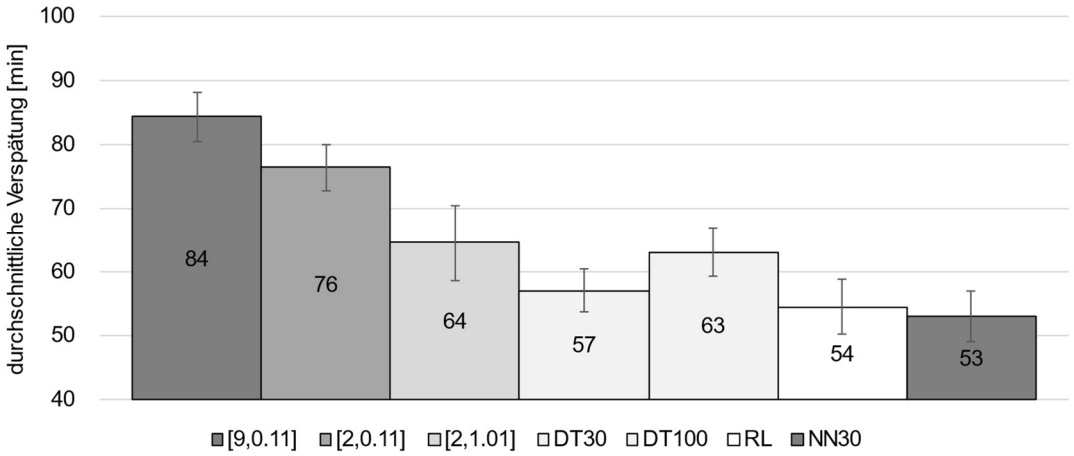


Abbildung 42: Vergleich der Methoden zur dynamischen Anpassung in einem dynamischen Szenario (Heger et al., 2021)

In der Evaluation zeigt sich, dass die Verwendung von DTs für bekannte Szenarien zusammen mit RL das Systemverhalten sehr gut wiedergeben können und bis zu einem gewissen Grad dynamische Verhalten nutzten können, um gute Leistung zu erreichen. Die Verwendung von NN und RL ist besonders in Szenarien mit unbekanntem Verhalten von Vorteil und kann zu einer Verbesserung der Leistung von bis zu 15 % führen.

5.5.3. Zusammenfassung

Die Verwendung von DT, NN und RL ist, je nach Szenario, für die dynamische Anpassung von Prioritätsregeln geeignet und kann zu einer Verbesserung der Leistung führen. Die Evaluation in den beschriebenen Szenarien zeigte die Vor- und Nachteile der drei Methoden auf. Es ist festzustellen, dass RL in beiden Szenarien mit einer flexiblen Fließfertigung den Zusammenhang zwischen k-Faktoren, Einlastung und Produktmix lernen und auf unbekannte Systemzustände übertragen konnte. Vor allem im Szenario mit unbekannten Situationen konnten NN und RL gute Leistungsvorhersagen treffen und so die Leistung um bis zu 15 % verbessern.

5.6. Anwendungsszenarien in der Produktion

Um die vollständige Prozesskette und eine potenzielle Anwendung aufzuzeigen, wurden lauffähige Beispiele im Rahmen der Leuphana Lernfabrik (Voß et al., 2021b) und im Rahmen eines industriellen Anwendungsfalles (Voß et al., 2021a) entwickelt. Zur Veranschaulichung wird im Folgenden die Applikation in der Lernfabrik näher erläutert. Die Verwendung von Lernfabriken kann die Hemmschwelle zur Anwendung von neuen Technologien senken, bei der

Einführung neuer Technologien helfen und bessere Lern-Erfolge als traditionelle Methoden zeigen (Baena et al., 2017). Auch wenn der Fokus dieser Arbeit auf der dynamischen Anpassung von Reihenfolgeregeln liegt, soll an diesem Beispiel das Vorgehen verdeutlicht werden. Das generelle Konzept der Pipeline von den Rohdaten über das Simulationsmodell bis zum endgültigen Agenten kann für andere Anwendungsfälle analog verwendet werden und soll an dieser Stelle gezeigt werden.

5.6.1. Beschreibung des Lernfabrik Szenarios

In dem betrachteten Szenario besteht die Fertigung aus einer kurzen Fließfertigung mit zwei Produktionsstufen in denen drei verschiedene Fahrzeugfamilien gefertigt werden. Die Fahrzeugfamilien haben individuelle Prozesszeiten und benötigen einen Rüstprozess am Anfang der Fertigung, bei dem die Rüstzeit reihenfolgeabhängig ist. In Abbildung 43 ist das Fertigungsprinzip für die Montage schematisch abgebildet. Dabei sind die in den Kästen angegeben Zahlen an der ersten und letzten Station die Montagezeiten für die drei unterschiedlichen Fahrzeugfamilien. Die zwei Zeiten an der Front-Station stehen für die zwei unterschiedlichen Varianten der Front, es ist nur eine Version des Hecks verfügbar.

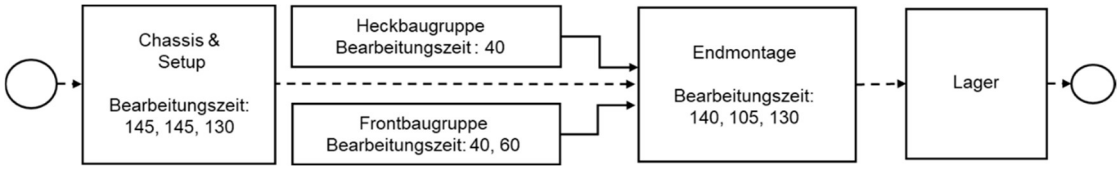


Abbildung 43: Montagestruktur in der Lernfabrik (Voß et al., 2021b)

Auch wenn eine grobe Schätzung des Kundenbedarfs bekannt ist, lässt sich dieser nicht genau vorhersagen und führt somit zu Unsicherheit in der Produktionsplanung. Alle Produkte haben eine knappe Kundenlieferzeit und es fallen Strafkosten an, sollten diese nicht eingehalten werden. Ein entscheidender Faktor ist, dass die benötigte Bearbeitungsdauer für die Montage der Fahrzeuge nur geringfügig niedriger ist als die geforderte Kundenlieferzeit und das Rüsten so zu einem Engpass wird.

Ausgehend von der geschätzten Zwischenankunftszeit ist davon auszugehen, dass die Warteschlangen in diesem Szenario sehr kurz sind und die Reihenfolgebildung vor den Maschinen kaum Einfluss auf die Leistung haben wird. Aufgrund der knappen Lieferzeit und den kurzen Warteschlangen ist es also entscheidend, wann welcher Auftrag mit welcher Losgröße im System eingesteuert wird.

Es ist daher notwendig, basierend auf dem Kundenbedarf, der Menge an Artikeln im Lager und der Anzahl an Artikeln in der Produktion dynamisch zu entscheiden, welches Produkt als nächstes gefertigt werden soll.

5.6.2. Methode zur Bestimmung von dynamischen Losgrößen

Im Rahmen vom Beitrag Maier et al. (2019) wurde für ein ähnliches Szenario ein mathematisches Modell entwickelt, welches für die optimale Losgrößenberechnung genutzt wurde. Weiterhin wurde die optimale Losgröße unter Unsicherheit wie schwankenden Prozesszeiten evaluiert. Es zeigt sich, dass das Model optimal für die gegebene Situation ist, sollte sich allerdings ein Kundenauftrag ändern oder eine Störung die Durchführung des Plans unmöglich machen, muss dieser erneut berechnet werden. Das im Beitrag verwendete Simulationsmodell wurde für dieses Szenario angepasst und mit Hilfe von Interviews und Beobachtung validiert und durch den Vergleich von simulierten und realen Daten verifiziert.

Zur Bestimmung der dynamischen Losgröße wurde ein RL-Agent mit der Simulation trainiert. In dem beschriebenen Szenario waren die Beobachtungen die Anzahl offener Kundenaufträge im System, die Anzahl der auf Lager liegenden Produkte sowie die Anzahl der Produkte in Bearbeitung. Die möglichen Handlungen, welche vom Agenten ausgeführt werden konnten, waren die Auswahl eines Typs und die dazugehörige Losgröße für den nächsten Fertigungsauftrag von Null bis vier. Der gemessene Leistungsindikator war der Gewinn nach einer definierten Zeit. Die Belohnungsfunktion ist definiert als die Veränderung des Gewinns zwischen zwei Aktionen.

In diesem Fall wurde das Training mit Hilfe der Pathmind Web App durchgeführt. Die mitgelieferte Toolbox wird in der Simulation eingebunden und bietet die Möglichkeit den Agenten als Blackbox mit der Simulation zu trainieren. Das Simulationsmodel wird auf die entsprechende Plattform geladen und der Trainingsprozess wird online auf bereitgestellten Servern durchgeführt. Da der Erfolg des RL-Trainings stark von der Modellstruktur (Breite und Tiefe) sowie der Lernrate und anderen Hyperparametern des NN abhängt werden nach dem Hochladen und Verbinden des Simulationsmodells mehrere Trainingsinstanzen mit individuellen Hyperparametern gestartet. Während des Trainingsprozesses werden die einzelnen Instanzen, die nicht konvergieren oder keine gute Leistung erbringen, zurückgelassen und neue Instanzen erzeugt. Der Prozess ist unter dem Namen Population Based Training (PBT) bekannt (Jaderberg et al., 2017). Das Verfahren ist bereits aus 5.3.2 und 5.5.1 bekannt.

Um die Leistung des Agenten mit individuellen Sätzen an Hyperparametern zu bewerten, werden während einer Iteration mehrere Simulationsläufe durchgeführt, um die Belohnung zu berechnen und so das NN mithilfe der Proximal Policy Optimization (PPO) zu trainieren (Schulman et al., 2017). Um die große Anzahl von Datenpunkten zu sammeln, die für den Ansatz benötigt werden, werden hunderte von Simulationsläufen für jeden Iterationsschritt durchgeführt. In dem gezeigten Beitrag verschiedene Sätze von Hyperparametern mit jeweils bis zu 200 Iterationen getestet, was zu mehr als 160000 einzelnen Simulationsläufen führte. Der leistungsstärkste Agent wurde mit 50460 Simulationen trainiert.

Sobald der beste Agent trainiert wurden, erstellt die Pathmind Web App einen Container, der dann wieder in der Simulation importiert und verwendet werden kann. Derselbe Container kann auch als Service online bereitgestellt und per REST-Schnittstelle angesprochen werden. Die Anpassung der Hyperparameter des Agenten wie die des NN werden durch Pathmind automatisch durchgeführt und sind für den Nutzer nicht zugänglich.

In Abbildung 44 sind die einzelnen Elemente des Vorgangs dargestellt. In der linken oberen Ecke ist das reale System zu finden. Das web-basierte Produktions- und Planungssystem der Lernfabrik erstellt die benötigten Aufträge und zeichnet die Daten der Produktion in einer Datenbank auf. Die aufgezeichneten Daten können ausgewertet und als Basis für das Simulationsmodell genutzt werden. Mit Hilfe der Simulation ist es dann möglich, den Agenten zu trainieren, welcher als Container auf der rechten Seite neben dem Online-Training zu sehen ist. Dieser kann sowohl in der Simulation wie auch über eine REST-Schnittstelle im realen System implementiert werden. So bietet der trainierte Agent die Möglichkeit den menschlichen Planer zu unterstützten oder zu ersetzten. Letzteres wurde im Rahmen des nachfolgenden Anwendungsfalles durchgeführt und wird im folgenden Abschnitt mit menschlichen Planern verglichen.

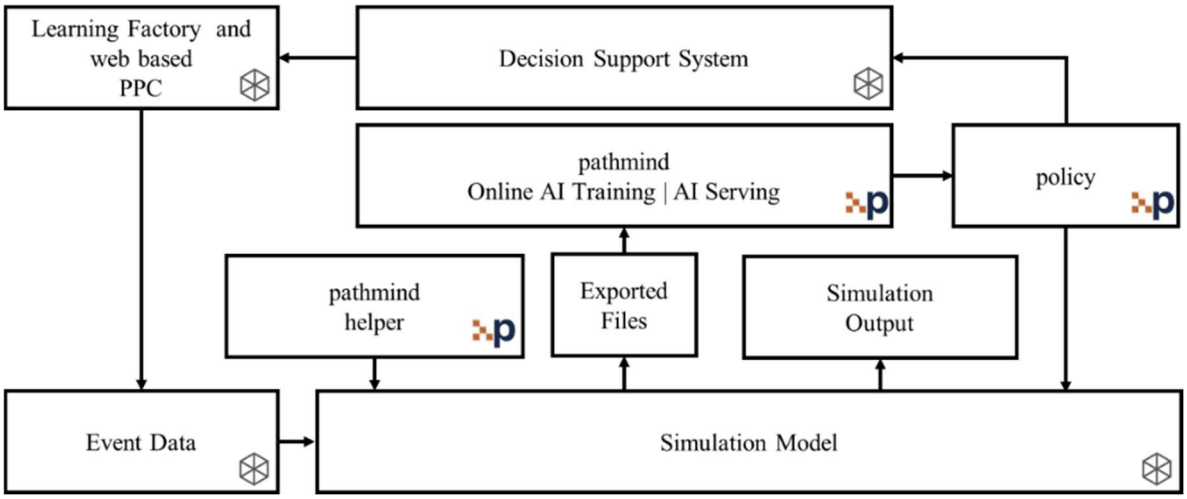


Abbildung 44: Die einzelnen Elemente für die Entwicklung eines RL-Agenten (Voß et al., 2021b)

5.6.3. Aufbau der Simulation

Die Simulation wurde auf Basis der über mehrere Läufe aufgezeichneten Daten modelliert. Dabei wurden die Fertigungsstruktur von zwei spezifischen Spielen exemplarisch ausgewählt und die sowie die Verteilung der Bearbeitungszeiten an allen Montagestationen aus dem jeweiligen Spiel berücksichtigt. Als Basiswert und zum Vergleich der Agentenleistung wurde eine stabile Basiskennlinie mit einer Make-to-Order (MTO)-Fertigungsstrategie implementiert. Diese wurde in der Simulation wie auch in der realen Welt getestet.

Die im Rahmen des Simulationsmodells verwendeten Kosten wurden auf der Grundlage von Voruntersuchungen gewählt, um bestimmte Aspekte zu betonen, zum Beispiel den Kompromiss zwischen Rüst- und Haltekosten. Generell gilt, dass für jeden Fertigungsauftrag definierte Work in Process (WIP)-Kosten anfallen, die sowohl die Materialbereitstellung als auch die Löhne berücksichtigen. Immer, wenn ein Montageauftrag in eine Arbeitsstation die Bearbeitung beginnt, wird der WIP auf der Grundlage des in der Arbeitsstation verwendeten Materials in Rechnung gestellt. Sollte ein Auftrag nicht rechtzeitig geliefert werden oder auf Lager liegen, werden Kosten für das Vorhalten von Produkten auf Lager sowie Strafkosten für die nicht rechtzeitige Lieferung fällig. Die Lagerkosten werden alle zwei Minuten auf der Grundlage der genauen Anzahl der auf Lager befindlichen Produkte berechnet. Die Strafkosten werden ebenfalls alle zwei Minuten berechnet und erhöhen sich proportional zur Verzögerung des Auftrags. Die Kosten für den Rüstvorgang fallen immer dann an, wenn eine Montagestation ihr Werkzeug wechseln muss. Sobald ein Produkt an den Kunden ausgeliefert wurde, wird der Umsatz auf Basis des Produkttyps

abgerechnet und der Gewinn verrechnet. Mit all den oben genannten Einschränkungen haben die Spieler ein klares Ziel: die Maximierung des Gewinns durch die Montage der richtigen Anzahl von Produkten zur richtigen Zeit. Es ist darauf hinzuweisen, dass auf Basis der zugrundeliegenden und aggregierten Informationen aus der Datenbank Montagezeit von ca. 145 Sekunden mit einer Standardabweichung von 30 Sekunden je nach Produkttyp gewählt wurden. So kommt es im System zu einer gewissen Unsicherheit bei der Montagezeit.

Während des Spiels mit der MTO-Strategie werden die Produkte montiert, sobald die Aufträge eintreffen. In den ersten zehn Minuten, die als Aufwärmphase betrachtet werden können, werden keine Bestellungen getätigt und somit auch keine Aufträge bearbeitet. Angesichts der Tatsache, dass die reine Montagezeit bei etwa sieben Minuten lag, konnte darauffolgend keines der Produkte den gewünschten Liefertermin einhalten und es mussten Strafgebühren gezahlt werden, was zu einem sehr geringen Gewinn führte. In Abbildung 45 zeigt die x-Achse die Zeit und die y-Achse zeigt den generierten Gewinn. Während die gepunktete Linie die Daten der Simulation zeigt, zeigt die durchgezogene Linie die Messpunkte des realen Systems. In beiden Verläufen ist zu erkennen, dass Kosten anfallen, wenn sich die Produkte entlang der Montagelinie bewegen. Die Dauer zwischen verschiedenen Zeitstempeln kann als die Zeit interpretiert werden, die für Montage, Lagerung und Bewegung benötigt wird. Die Abweichung der Simulation von den realen Daten kann durch die stochastischen Einflüsse der aufgezeichneten Montagezeiten erklärt werden. Da das Simulationsmodell und das reale Modell zu vergleichbaren Werten kommen, ist davon auszugehen, dass es sich bei der Simulation um ein vergleichbares Abbild des realen Systems handelt. So kann festgehalten werden, dass bei gleichbleibendem Beobachtungsraum und gleicher Systemkonfiguration der mit der Simulation trainierte RL-Agenten in das reale System transferiert werden kann. Weiterhin sollten die Ergebnisse des Agenten im realen System mit der Evaluation des Agenten in der Simulation vergleichbar sein.

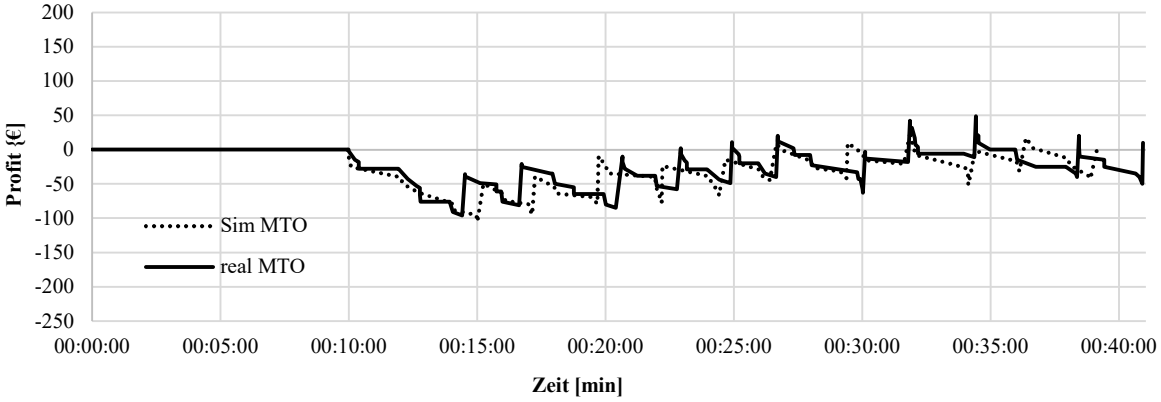


Abbildung 45: Vergleich der Leistung der MTO-Strategie in Simulation und realem System (Voß et al., 2021b)

5.6.4. Dynamische Auswahl der Losgrößen mit bestärkendem Lernen

Ausgehend von den aufgezeichneten Daten, die über mehrere Runden der Produktion gesammelt wurden, ist es möglich, die Leistung des Agenten direkt mit der menschlichen Leistung zu vergleichen. So sind der Abbildung 46 die Daten für die Runden 139 und 140 sowie die Aufzeichnungen für den RL-Agenten zu sehen. Es handelt sich um den Gewinn im System (auf der y-Achse) über die Zeit in der Produktion (auf der x-Achse).

Es fällt auf, dass der Profit für die ersten zehn Minuten in allen drei Fällen bei etwa -200 € liegt, was auf das proaktive Montieren von Fahrzeugen zurückzuführen ist. Da die Teilnehmer verstanden haben, dass der Rüstprozess an der ersten Maschine eine Schlüsselstelle ist, werden bestimmte Fahrzeuge vorab montiert und auf Lager gelegt, während der Rest nach Bedarf montiert wird. Es ist offensichtlich, dass der Agent dieses Verhalten in der Simulation ebenfalls erfahren hat und entsprechend darauf reagiert. Für die Daten aus Runde 140 zeigt sich ein niedriger Profit zwischen 25:00 und 30:00 Minuten, was durch einen Absprachefehler entstanden ist. Ähnlich verhält sich der Agent, welcher Fahrzeuge produzieren lässt, die erst später benötigt werden.

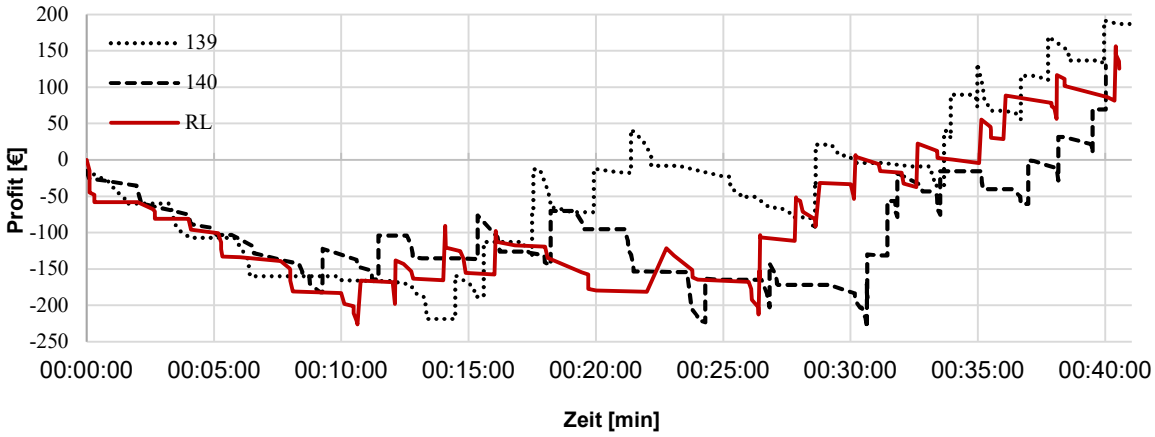


Abbildung 46: Profit der verschiedenen Runden über die Zeit (Voß et al., 2021b)

Um das Verhalten des Agenten zu bestätigen, wurde die Strategie durch die Simulation evaluiert und zeigte bei 100 Replikationen einen durchschnittlichen Profit von 127.8 ± 5.80 €. Es ist also davon anzunehmen, dass die gute Leistung des Agenten an dieser Stelle nicht auf eine günstige Kombination an Faktoren zurückzuführen ist.

Im direkten Vergleich fällt auf, dass die Runde 139 mit Abstand die besten Ergebnisse produziert hat. Der RL-Agent und die Runde 140 haben vergleichbare Ergebnisse erreicht und den Referenzwert der MTO-Strategie signifikant geschlagen. Es lässt sich also feststellen, dass der Agent menschenähnliche Leistung in diesem Anwendungsfall bringt. Die Runde 139 erreichte herausragende Ergebnisse und war in der Lage sich an die dynamische Situation in der Montage anzupassen. Ein Vergleich der Leistungskennzahl Profit (Tabelle 16), der in Arbeit befindlichen Fahrzeuge, der offenen Bestellungen und der im Lager liegenden Fahrzeuge kann weitere Einsichten in das genaue Verhalten bringen.

Tabelle 16: Runde 139 erwirtschaftet den höchsten Profit

Bezeichnung	Profit [€]
Runde ID 139	187
Runde ID 140	129.75
RL	125.5
MTO	10

5.6.5. Zusammenfassung

Die Anpassung der Losgröße an die aktuellen Systemzustände ist eine Herausforderung. Bestehende Strategien wie MTO sind nur bedingt geeignet, um dynamisch auf unbekannte Situationen zu reagieren. Die Anwendung eines RL-Agenten für die adaptive Losgrößenbestimmung kann zu einer guten Leistung führen und den Menschen bei komplexen Aufgaben unterstützten.

Der Prozess beginnt bei der Aufzeichnung von Daten, die dann zur Entwicklung eines Simulationsmodells genutzt werden. Mit diesem kann ein RL-Agent so trainiert und eingesetzt werden, dass er eine mit menschlichen Planern vergleichbare Leistung erbringt.

Der RL-Agent erzielte ein besseres Ergebnis als die MTO-Politik, wurde aber durch die besten menschlichen Planner übertroffen. Es kann geschlussfolgert werden, dass die Verwendung von RL als Werkzeug zur Entscheidungsunterstützung in diesem Szenario möglich ist, für größere und komplexere Szenarien aber noch evaluiert werden muss.

5.7. Zusammenfassung Konzept und Evaluation

Ausgehend von der der RoboCup Logistic Liga bis hin zur komplexen Fließfertigung mit parallelen Maschinen und reihenfolgeabhängigen Rüstzeiten, wurden unterschiedlich komplexe Systeme beschrieben und modelliert. Es zeigte sich, dass mit der mathematischen Modellierung für kleine Szenarien eine optimale Lösung errechnet werden kann. Um dynamische Einflüsse und komplexe Interaktionsbeziehungen evaluieren zu können ist die Methode allerdings nicht geeignet. Aus diesem Grund wurden Modelle eines komplexen und dynamischen Produktionssystems als ereignisdiskrete Simulation aufgebaut und in Kombination mit dezentralen Prioritätsregeln zur Reihenfolgebildung und Routenauswahl kombiniert.

Die erste durchgeführten Parameterstudien zeigen die Notwendigkeit zur Verwendung von produktmix- und systemzustands-spezifischen Kombinationen für Reihenfolge- und Routenregeln. Für Ausgehend bekannte Systemzustände können gute Entscheidungen bzgl. der Reihenfolge- und Routenregeln getroffen werden. Sollte es sich allerdings um unbekannte Szenarien handeln, so muss das Verhalten bei einer entsprechenden Kombination entweder nachsimuliert oder durch ein Regressionsverfahren geschätzt werden. Zu diesem Zweck wurde ein NN zur Schätzung des Systemverhaltens bei unterschiedlichen Regelkombinationen trainiert. Dieses kann als Wissensbasis für das System genutzt werden und bei der Auswahl der besten Regelkombination unterstützten.

Zur systematischen Nutzung des Wissens und zur dynamischen Anpassung wurde erfolgreich ein Agent mit RL, der vergleichbare Ergebnisse erreichen kann, trainiert. Im Rahmen des Trainings wurden gute Beobachtungs- und Aktionsräume sowie die Trainingsdauer evaluiert und optimiert. Die neue Methode zur dynamischen Anpassung mit RL wurde in verschiedenen Szenarien evaluiert und für tauglich befunden. Dabei ist der Agent in der Lage zwischen unterschiedlichen Reihenfolgeregeln zu wählen, wenn sich der Systemzustand ändert. Entscheidend ist hierbei, dass der Agent dynamisch auf minimale Schwankungen im vermeintlich stabilen System reagiert, welche durch leicht schwankende Ankunftszeiten ausgelöst sind, und so die Leistung verbessert.

Im nächsten Schritt wurde der Agent befähigt in einer kombinierten Reihenfolgeregel die Gewichte von zwei Termen schrittweise, an sich verändernde Systemzustände, anzupassen. Der Agent war so in der Lage in unbekannten Szenarien, wie zum Beispiel bei einem Produktmixwechsel während der Simulation, zu reagieren und die Leistung des Systems zu verbessern. Weiterhin war der Agent in der Lage Schwankungen durch die Änderung in der Zwischenankunftszeit zu erkennen und zu reagieren. In beiden Fällen zeigte sich, dass die dynamische Anpassung in der Lage war, die Systemleistung stabil zu halten und teilweise sogar zu verbessern. Die Methode wurde in Konkurrenz mit anderen Ansätzen des maschinellen Lernens wie NN und DT entwickelt und evaluiert. Es zeigte sich, dass RL immer mit einer anderen Methode vergleichbare Leistung bringt.

Schlussendlich wurde die Verwendung von RL im Rahmen eines realen Produktionssystems aufgezeigt. Im Anwendungsfall wurde der RL-Agent mit Hilfe einer Simulation, die basierend auf aufgezeichneten Daten erstellt wurde, trainiert. Der Agent wurde dann in der realen Fertigung zur Auswahl von Losgrößen eingesetzt und konnte menschenähnliche Leistung erbringen.

6. Fazit und Ausblick

In der Arbeit wird eine neue Methode zur dynamischen Auswahl und Anpassung von Reihenfolgeregeln in komplexen Fertigungssystemen mit bestärkendem Lernen untersucht.

Im Spannungsfeld der logistischen Zielgrößen, müssen Unternehmen bei möglichst niedrigen Beständen ihre Kunden schnell und sicher beliefern. Dazu ist es notwendig zur richtigen Zeit die richtigen Produkte auf der richtigen Maschine zu fertigen. Die Herausforderungen dabei reichen von neu konfigurierbaren Fertigungsorganisationsformen und der damit verbundenen Komplexität der Fertigung bis zur dynamischen Anpassung an sich verändernde Systemzustände. Weiterhin bietet die Vernetzung aller Akteure im Produktionssystem die Möglichkeit gute Entscheidungen auch ohne zentrale Planungsinstanz zu treffen.

Die bestehenden Methoden sind für die aktuellen Anforderungen nicht ausreichend. So kann durch optimierende Verfahren für kleine und eindeutig definierte Szenarien der für den Zeitpunkt bestmöglichen Plan errechnen werden. Bei größeren Systemen ist die allerdings nicht mehr möglich, hier müssen Heuristiken den zentralen Plan errechnen. Treten dynamische Ereignisse ein, müssen aber auch diese, neue Lösungen errechnen.

Als Alternative ist die Verwendung von Prioritätsregeln lange bekannt und ermöglicht es, dezentral und dynamisch eine Reihenfolgebildung vorzunehmen. Es ist bekannt, dass von den über 100 bekannten Regeln keine in allen Situationen die beste Regel ist. Weiterhin ist in den komplexen Fertigungssystemen die Interaktion aus Reihenfolge- und Routenauswahlregeln ein entscheidender Faktor.

Im Rahmen der Arbeit werden verschiedene Produktionsszenarien beschrieben und evaluiert. Da auf Grund der benötigten Rechenleistung nicht alle Systemzustände in Kombination mit den möglichen Reihenfolge- und Routenregel-kombination getestet werden können, wird eine begrenzte Menge an Systemzustände simuliert und ausgehend von dieser Wissensbasis, die Leistung für die nicht bekannten Zustände mit Hilfe eines NN geschätzt.

Da die Verwendung von RL im Rahmen der Produktionsplanung und -steuerung bereits gute Ergebnisse erzielt hat, wurde eine Methode zur dynamischen Anpassung von Prioritätsregeln mit eben dieser kombiniert. Diese wurde in den unterschiedlichen Szenarien trainiert und folgend mit unbekannten Situationen konfrontiert. Es zeigt sich, dass die neue Methode in der Lage ist, kleinste

Schwankungen im Systemzustand zu erkennen und darauf zu reagieren, was eine Verbesserung der Leistung zu Folge hat.

Im letzten Schritt wurde die Erprobung an einem Anwendungsbeispiel demonstriert. Dazu wurde ein Agent in einer Simulation zur dynamischen Auswahl von Losgrößen trainiert. Dieser war dann als Blackbox in der Simulation wie auch im realen Produktionssystem einsetzbar. Im realen System wurde der Agent in das web-basierte Produktionsplanungs- und Steuerungssystem integriert und es zeigte sich, dass der Agent in der Lage ist mit Menschen vergleichbare Lösungen zu erreichen.

Im Rahmen der Arbeit wurde eine Methode entwickelt, die zentral Informationen sammelt und in der Lage ist dynamisch auf Änderungen im Systemzustand zu reagieren. Die Änderungen ermöglichen die Anpassung von Prioritätsregeln, welche unabhängig von der Organisationsform und dezentral in der Lage sind gute Entscheidungen unter Unsicherheit zu treffen. Im Kontext der Reihenfolgeregelauswahl zeigte die neue Methode gute Leistungen bei der Verwendung in bekannten und unbekannten Szenarien. Sie war in der Lage die statische Verwendung von Regeln wie auch die Kombination aus Brute-Force Parameteroptimierung und dynamische Anpassung zu schlagen. Durch die Anpassung von bekannten Prioritätsregeln ist der Mensch teilweise in der Lage das Verhalten nachzuvollziehen.

Zu den weiterführenden Fragestellungen gehört der Aspekt der Vergleich mit weiteren Methoden des maschinellen Lernens zur dynamischen Anpassung von Reihenfolgeregeln. Auch wenn die Verwendung einer umfangreichen Datenbasis die Verhaltensvorhersage überflüssig machte, könnten andere Methoden ebenfalls gute und robuste Ergebnisse bringen. Weiterhin müssen die Hyperparameter des RL-Ansatzes optimiert werden, um die Präzision des Agenten zu untersuchen. In diesem Kontext gilt es auch zu überprüfen, inwieweit die Handlungen des Agenten generalisierbar und auf andere Anwendungsfälle übertragbar sind. Der Vergleich zwischen der Anpassung von Reihenfolgeregeln und der Auswahl konkreter Operationen steht ebenfalls aus.

Beginnend mit dem Szenario der RoboCup Logistik Liga waren zwei Entscheidungen zu treffen, die Reihenfolge- und die Routenplanung. Der bisherige Ansatz fokussiert sich auf den ersten Aspekt, die Betrachtung von zwei unterschiedlichen Entscheidungen – Reihenfolge- und Routenauswahl – in Kombination steht noch aus. Es ist zu evaluieren, ob ein zentraler Agent, der beide Entscheidungen trifft oder ein Agent pro Entscheidung eine bessere Wahl ist. Zur vollständigen Integration in die RoboCup Logistik Liga müsste jedes FTS

und jede Maschine mit einem individuellen Agenten zur dynamischen Auswahl von Operationen ausgestattet sein. Hier stellt sich die Frage ob individuelle Strategien oder eine zentrale Strategie zum Erfolg führen. Der Bereich der populationsbasierten Agententrainings ermöglicht hierbei noch etliche weitere Fragestellungen bezüglich der Belohnungsfunktion und Agenteninteraktion.

Literaturverzeichnis

Acatech Study: Industrie 4.0 Maturity Index: Managing the Digital Transformation of Companies. Hg. v. Günther Schuh, Reiner Anderl, Jürgen Gausemeier und Michael ten Hompel, 2017,

Acker, I.J.: Methoden zur mehrstufigen Ablaufplanung in der Halbleiterindustrie: Gabler Verlag 2011.

Alemão, D.; Rocha, A.D.; Barata, J.: Smart Manufacturing Scheduling Approaches—Systematic Review and Future Directions. Applied Sciences 11 (2021) 5, S. 2186.

Aydin, M.E.; Öztemel, E.: Dynamic job-shop scheduling using reinforcement learning agents. Robotics and Autonomous Systems 33 (2000) 2-3, S. 169–178.

Baena, F.; Guarin, A.; Mora, J.; Sauza, J.; Retat, S.: Learning Factory: The Path to Industry 4.0. Procedia Manufacturing 9 (2017), S. 73–80.

Baker, K.R.: The effects of input control in a simple scheduling model. Journal of Operations Management 4 (1984) 2, S. 99–112.

Bauernhansl, T.; Hompel, M. ten; Vogel-Heuser, B.: Industrie 4.0 in Produktion, Automatisierung und Logistik. Wiesbaden: Springer Fachmedien Wiesbaden 2014.

Botthof, A.; Gabriel, P.: AUTONOMIK für Industrie 4.0. Hg. v. Bundesministerium für Wirtschaft und Energie (BMWi)Berlin, 2016,

Branke, J.; Hildebrandt, T.; Scholz-Reiter, B.: Hyper-heuristic Evolution of Dispatching Rules. Evolutionary Computation 23 (2015) 2, S. 249–277.

Branke, J.; Nguyen, S.; Pickardt, C.W.; Zhang, M.: Automated Design of Production Scheduling Heuristics. IEEE Transactions on Evolutionary Computation 20 (2016) 1, S. 110–124.

Brucker, P.; Heitmann, S.; Hurink, J.; Nieberg, T.: Job-shop scheduling with limited capacity buffers. OR Spectrum 28 (2006) 2, S. 151–176.

Brucker, P.; Knust, S.: Complex Scheduling. Berlin, Heidelberg: Springer Berlin Heidelberg 2012.

Brucker, P.; Knust, S.; Cheng, T.E.; Shakhlevich, N.V.: Complexity Results for Flow-Shop and Open-Shop Scheduling Problems with Transportation Delays. Annals of Operations Research 129 (2004) 1-4, S. 81–106.

Buxmann, P.; Schmidt, H.: Künstliche Intelligenz. Berlin, Heidelberg: Springer Berlin Heidelberg 2019.

Conway, R.W.: An Experimental Investigation of Priority Assignment in a Job Shop. Santa Monica, CA: RAND Corporation 1964.

Dangelmaier, W.: Theorie der Produktionsplanung und -steuerung. Berlin, Heidelberg: Springer Berlin Heidelberg 2009.

Dantzig, G.B.: Origins of the simplex method. In: Nash, S.G. (Hrsg.): A history of scientific computing. New York, NY, USA: ACM 1990, S. 141–151.

Doh, H.-H.; Yu, J.-M.; Kim, J.-S.; Lee, D.-H.; Nam, S.-H.: A priority scheduling approach for flexible job shops with multiple process plans. International Journal of Production Research 51 (2013) 12, S. 3748–3764.

Domschke, W.; Drexl, A.; Klein, R.; Scholl, A.: Einführung in Operations Research. Berlin, Heidelberg: Springer Berlin Heidelberg 2015.

El-Bouri, A.; Shah, P.: A neural network for dispatching rule selection in a job shop. The International Journal of Advanced Manufacturing Technology 31 (2006) 3-4, S. 342–349.

Eley, M.: Simulation in der Logistik: Einführung in die Erstellung ereignisdiskreter Modelle unter Verwendung des Werkzeuges "Plant Simulation". Berlin, Heidelberg: Springer Berlin Heidelberg 2012.

Fourer, R.; Gay, D.M.; Kernighan, B.: AMPL: A Mathematical Programming Language: Boyd & Fraser Danvers, MA 1993.

Frazzon, E.M.; Hartmann, J.; Makuschewitz, T.; Scholz-Reiter, B.: Towards Socio-Cyber-Physical Systems in Production Networks. Procedia Cirp 7 (2013), S. 49–54.

Fries, C.; Wiendahl, H.-H.; Assadi, A.A.: Design concept for the intralogistics material supply in matrix productions. Procedia Cirp 91 (2020), S. 33–38.

Gabel, T.: Multi-agent reinforcement learning approaches for distributed job-shop scheduling problems. Osnabrück, Universität Osnabrück, Fachbereich Mathematik/Informatik, 2009.

Gabel, T.; Riedmiller, M.: Adaptive reactive job-shop scheduling with reinforcement learning agents. International Journal of Information Technology and Intelligent Computing 24 (2008) 4, S. 14–18.

Gere Jr, W.S.: Heuristics in job shop scheduling. Management Science 13 (1966) 3, S. 167–190.

Gomes, M.C.; Barbosa-Povoa, A.P.; Novais, A.Q.: Optimal scheduling for flexible job shop operation. International Journal of Production Research 43 (2005) 11, S. 2323–2353.

Greschke, P.: Matrix-Produktion als Konzept einer taktunabhängigen Fließfertigung. [Essen]: Vulkan Verlag 2016.

Greschke, P.; Schönemann, M.; Thiede, S.; Herrmann, C.: Matrix Structures for High Volumes and Flexibility in Production Systems. Procedia Cirp 17 (2014), S. 160–165.

Grill-Kiefer, G.: Logistik in der Automobilindustrie. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 115 (2020) 9, S. 595–601.

Gröflin, H.; Klinkert, A.: A new neighborhood and tabu search for the blocking job shop. Discrete Applied Mathematics 157 (2009) 17, S. 3643–3655.

Gronau, N.: Determinants of an Appropriate Degree of Autonomy in a Cyber-physical Production System. Procedia Cirp 52 (2016), S. 1–5.

Grundstein, S.; Schukraft, S.; Görges, M.; Scholz-Reiter, B.: Interlinking central production planning with autonomous production control. Advances in Production, Automation and Transportation Systems (2013), S. 326–332.

Gurobi Optimization, I.Gurobi Optimizer Reference Manual, 2016: Gurobi Optimizer Reference Manual. Online verfügbar unter http://www.gurobi.com.

Gutenschwager (Hg.): Simulation in Produktion und Logistik. Berlin, Heidelberg: Springer Berlin Heidelberg 2017.

Hackstein, R.: Produktionsplanung und -steuerung (PPS): Ein Handbuch für die Betriebspraxis. Düsseldorf: VDI - Verlag 1984.

Heger, J.: Dynamische Regelselektion in der Reihenfolgeplanung. Wiesbaden: Springer Fachmedien Wiesbaden 2014.

Heger, J.; Branke, J.; Hildebrandt, T.; Scholz-Reiter, B.: Dynamic adjustment of dispatching rule parameters in flow shops with sequence-dependent set-up times. International Journal of Production Research 54 (2016) 22, S. 6812–6824.

Heger, J.; El Abdine, M.Z.; Sekar, S.; Voß, T.: Entscheidungsbäume und bestärkendes Lernen zur dynamischen Auswahl von Reihenfolgeregeln in einem flexiblen Produktionssystem. Simulation in Produktion und Logistik 2021: Erlangen, 15.-17. September 2021 (2021), S. 337.

Heger, J.; Voss, T.: Optimal Scheduling for Automated Guided Vehicles (AGV) in Blocking Job-Shops. In: Lödding, H.; Riedel, R.; Thoben, K.-D.; Cieminski, G. von; Kiritsis, D. (Hrsg.): Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing. Cham: Springer International Publishing 2017, S. 151–158.

Heger, J.; Voss, T.: Optimal Scheduling of AGVs in a Reentrant Blocking Job-shop. Procedia Cirp 67 (2018), S. 41–45.

Heger, J.; Voß, T.: Dynamic priority based dispatching of AGVs in flexible job shops. Procedia Cirp 79 (2019), S. 445–449.

Heger, J.; Voss, T.: Reducing mean tardiness in a flexible job shop containing AGVs with optimized combinations of sequencing and routing rules. Procedia Cirp 81 (2019), S. 1136–1141.

Heger, J.; Voss, T.: Dynamically Changing Sequencing Rules with Reinforcement Learning in a Job Shop System With Stochastic Influences. In: Winter Simulation Conference (WSC), Orlando, FL, USA, 14.12.2020 - 18.12.2020, 2020, S. 1608–1618.

Heger, J.; Voss, T.: Dynamically adjusting the k -values of the ATCS rule in a flexible flow shop scenario with reinforcement learning. International Journal of Production Research (2021), S. 1–15.

Ho, N.B.; Tay, J.C.: Evolving dispatching rules for solving the flexible job-shop problem. In: 2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005. Proceedings 2005, S. 2848–2855.

Hofmann, C.; Brakemeier, N.; Krahe, C.; Stricker, N.; Lanza, G.: The Impact of Routing and Operation Flexibility on the Performance of Matrix Production Compared to a Production Line. In: Schmitt, R.; Schuh, G. (Hrsg.): Advances in Production Research. Cham, Switzerland: Springer 2019, S. 155–165.

Hofmann, C.; Krahe, C.; Stricker, N.; Lanza, G.: Autonomous production control for matrix production based on deep Q-learning. Procedia Cirp 88 (2020), S. 25–30.

Holland, J.H.: Genetic Algorithms and Adaptation. In: Selfridge, O.G.; Rissland, E.L.; Arbib, M.A. (Hrsg.): Adaptive Control of Ill-Defined Systems. Boston, MA: Springer US 1984, S. 317–333.

Holthaus, O.; Rajendran, C.: Efficient jobshop dispatching rules. Production Planning & Control 11 (2000) 2, S. 171–178.

Homberger, J.; Bauer, H.; Preissler, G.: Operations Research und Künstliche Intelligenz: utb GmbH 2019.

Hornik, K.; Stinchcombe, M.; White, H.; others: Multilayer feedforward networks are universal approximators. Neural Networks 2 (1989) 5, S. 359–366.

Jaderberg, M.; Czarnecki, W.M.; Dunning, I.; Marris, L.; Lever, G.; Castañeda, A.G.; Beattie, C.; Rabinowitz, N.C.; Morcos, A.S.; Ruderman, A.; Sonnerat, N.; Green, T.; Deason, L.; Leibo, J.Z.; Silver, D.; Hassabis, D.; Kavukcuoglu, K.; Graepel, T.: Human-level performance in 3D multiplayer games with population-based reinforcement learning. Science (New York, N.Y.) 364 (2019) 6443, S. 859–865.

Jaderberg, M.; Dalibard, V.; Osindero, S.; Czarnecki, W.M.; Donahue, J.; Razavi, A.; Vinyals, O.; Green, T.; Dunning, I.; Simonyan, K.; others: Population based training of neural networks. arXiv preprint arXiv:1711.09846 (2017).

Joshi, A.V.: Machine learning and artificial intelligence. Cham: Springer 2020.

Jun, S.; Lee, S.: Learning dispatching rules for single machine scheduling with dynamic arrivals based on decision trees and feature construction. International Journal of Production Research 59 (2021) 9, S. 2838–2856.

Jun, S.; Lee, S.; Chun, H.: Learning dispatching rules using random forest in flexible job shop scheduling problems. International Journal of Production Research 57 (2019) 10, S. 3290–3310.

Kagermann, H.; Wahlster, W.; Helbig, J. (2013): Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0: Abschlussbericht des Arbeitskreises Industrie 4.0.

Kim, C.W.; Tanchoco, J.A.; Koo, P.-H.: AGV dispatching based on workload balancing. International Journal of Production Research 37 (1999) 17, S. 4053–4066.

Koren, Y.; Hill, R.: The global manufacturing revolution: Product-process-business integration and reconfigurable systems. Hoboken, N.J.: Wiley 2010.

Koza, J.R.: Genetic programming: on the programming of computers by means of natural selection: MIT Press 1992.

Kruse, R.; Borgelt, C.; Braune, C.; Klawonn, F.; Moewes, C.; Steinbrecher, M.: Computational Intelligence-Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze. 1. Auflage. Wiesbaden: Vieweg+ Teubner (2011).

Kuhnle, A.: Adaptive order dispatching based on reinforcement learning: Application in a complex job shop in the semiconductor industry. Düren: Shaker Verlag GmbH 2020.

Kuhnle, A.; Kaiser, J.-P.; Theiß, F.; Stricker, N.; Lanza, G.: Designing an adaptive production control system using reinforcement learning. Journal of Intelligent Manufacturing (2021) 32, S. 855–876.

Kuhnle, A.; Röhrig, N.; Lanza, G.: Autonomous order dispatching in the semiconductor industry using reinforcement learning. Procedia Cirp 79 (2019), S. 391–396.

Kumar, R.: Simulation of Manufacturing System at Different Part Mix Ratio and Routing Flexibility. Global Journal of Enterprise Information System 8 (2016) 1, S. 10–14.

Land, A.H.; Doig, A.G.: An Automatic Method of Solving Discrete Programming Problems. Econometrica 28 (1960) 3, S. 497.

Lee, W.-J.; Kim, B.-H.; Ko, K.; Shin, H.: Simulation Based Multi-Objective Fab Scheduling by Using Reinforcement Learning. In: Mustafee, N.; Bae, K.-H.G.; Lazarova-Molnar, S.; Rabe, M.; Szabo, C.; Haas, P.; Son, Y.-J. (Hrsg.): Proceedings of the 2019 Winter Simulation Conference, 2019, S. 2236–2247.

Lee, Y.H.; Jeong, C.S.; Moon, C.: Advanced planning and scheduling with outsourcing in manufacturing supply chain. Computers & Industrial Engineering 43 (2002) 1, S. 351–374.

Lödding, H.: Verfahren der Fertigungssteuerung. Berlin, Heidelberg: Springer Berlin Heidelberg 2016.

Lorenz, U.: Reinforcement Learning. Berlin, Heidelberg: Springer Berlin Heidelberg 2020.

Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.-I.: From Local Explanations to Global Understanding with Explainable AI for Trees. Nature machine intelligence 2 (2020) 1, S. 56–67.

Maier, J.T.; Voß, T.; Heger, J.; Schmidt, M.: Simulation Based Optimization of Lot Sizes for Opposing Logistic Objectives. In: Ameri, F.; Stecke, K.E.; Cieminski, G. von; Kiritsis, D. (Hrsg.): Advances in Production Management Systems. Towards Smart Production Management Systems. Cham: Springer International Publishing 2019, S. 171–179.

Martins, L.; Varela, M.L.; Fernandes, N.O.; Carmo–Silva, S.; Machado, J.: Literature review on autonomous production control methods. Enterprise Information Systems 14 (2020a) 8, S. 1219–1231.

Martins, L.M.; Fernandes, N.O.; Varela, M.L.; Dias, L.M.; Pereira, G.A.; Silva, S.C.: Comparative study of autonomous production control methods using simulation. Simulation Modelling Practice and Theory 104 (2020b), S. 102142.

Mati, Y.; Xie, X.: A genetic-search-guided greedy algorithm for multi-resource shop scheduling with resource flexibility. IIE Transactions 40 (2008) 12, S. 1228–1240.

Mattfeld, D.C.; Bierwirth, C.: An efficient genetic algorithm for job shop scheduling with tardiness objectives. European Journal of Operational Research 155 (2004) 3, S. 616–630.

Matzka, S.: Künstliche Intelligenz in den Ingenieurwissenschaften. Wiesbaden: Springer Fachmedien Wiesbaden 2021.

Michalewicz, Z.: Genetic algorithms+ data structures= evolution programs: Springer Science & Business Media 2013.

Mittal, S.; Khan, M.A.; Romero, D.; Wuest, T.: A critical review of smart manufacturing & Industry 4.0 maturity models: Implications for small and medium-sized enterprises (SMEs). Journal of Manufacturing Systems 49 (2018), S. 194–214.

Mönch, L.; Zimmermann, J.: Simulation-based assessment of machine criticality measures for a shifting bottleneck scheduling approach in complex manufacturing systems. Computers in Industry 58 (2007) 7, S. 644–655.

Mönch, L.; Zimmermann, J.; Otto, P.: Machine learning techniques for scheduling jobs with incompatible families and unequal ready times on parallel batch machines. Engineering Applications of Artificial Intelligence 19 (2006) 3, S. 235–245.

Mouelhi-Chibani, W.; Pierreval, H.: Training a neural network to select dispatching rules in real time. Computers & Industrial Engineering 58 (2010) 2, S. 249–256.

Nguyen, S.; Mei, Y.; Zhang, M.: Genetic programming for production scheduling: a survey with a unified framework. Complex & Intelligent Systems 3 (2017) 1, S. 41–66.

Nguyen, S.; Zhang, M.; Johnston, M.; Tan, K.C.: A Computational Study of Representations in Genetic Programming to Evolve Dispatching Rules for the Job Shop Scheduling Problem. IEEE Transactions on Evolutionary Computation 17 (2013) 5, S. 621–639.

Nunes, I.; Jannach, D.: A systematic review and taxonomy of explanations in decision support and recommender systems. User Modeling and User-Adapted Interaction 27 (2017) 3-5, S. 393–444.

Nyhuis, P.; Wiendahl, H.-P.: Logistische Kennlinien: Grundlagen, Werkzeuge und Anwendungen. Berlin, Heidelberg: Springer 2012.

Ouelhadj, D.; Petrovic, S.: A survey of dynamic scheduling in manufacturing systems. Journal of Scheduling 12 (2009) 4, S. 417–431.

Panwalkar, S.S.; Iskander, W.: A Survey of Scheduling Rules. Operations Research 25 (1977) 1, S. 45–61.

Park, I.-B.; Huh, J.; Kim, J.; Park, J.: A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities. IEEE Transactions on Automation Science and Engineering (2020), S. 1–12.

Pezzella, F.; Morganti, G.; Ciaschetti, G.: A genetic algorithm for the Flexible Job-shop Scheduling Problem. Computers & Operations Research 35 (2008) 10, S. 3202–3212.

Pickardt, C.W.; Hildebrandt, T.; Branke, J.; Heger, J.; Scholz-Reiter, B.: Evolutionary generation of dispatching rule sets for complex dynamic scheduling problems. International Journal of Production Economics 145 (2013) 1, S. 67–77.

Poppenborg, J.; Knust, S.; Hertzberg, J.: Online scheduling of flexible job-shops with blocking and transportation. European Journal of Industrial Engineering (EJIE) 6 (2012) 4, S. 497–518.

Priore, P.; Gómez, A.; Pino, R.; Rosillo, R.: Dynamic scheduling of manufacturing systems using machine learning: An updated review. AI EDAM-ARTIFICIAL INTELLIGENCE FOR ENGINEERING DESIGN ANALYSIS AND MANUFACTURING 28 (2014) 1, S. 83–97.

Priore, P.; La Fuente, D. de; Puente, J.; Parreno, J.: A comparison of machine-learning algorithms for dynamic scheduling of flexible manufacturing systems. Engineering Applications of Artificial Intelligence 19 (2006) 3, S. 247–255.

Rai, A.: Explainable AI: from black box to glass box. Journal of the Academy of Marketing Science 48 (2020) 1, S. 137–141.

Rebala, G.; Ravi, A.; Churiwala, S.: An Introduction to Machine Learning. Cham: Springer International Publishing 2019.

Rehse, J.-R.; Mehdiyev, N.; Fettke, P.: Towards Explainable Process Predictions for Industry 4.0 in the DFKI-Smart-Lego-Factory. KI - Künstliche Intelligenz 33 (2019) 2, S. 181–187.

Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review 65 (1958) 6, S. 386.

Runkler, T.A.: Data Mining. Wiesbaden: Vieweg+Teubner 2010.

Sastry, K.; Goldberg, D.; Kendall, G.: Genetic algorithms. In: Search methodologies: Springer 2005, S. 97–125.

Schacht, S.; Lanquillon, C.: Blockchain und maschinelles Lernen. Berlin, Heidelberg: Springer Berlin Heidelberg 2019.

Scholz-Reiter, B.; Dashkovskiy, S.; Görges, M.; Naujok, L.: Stability analysis of autonomously controlled production networks. International Journal of Production Research 49 (2011) 16, S. 4857–4877.

Scholz-Reiter, B.; Freitag, M.: Autonomous processes in assembly systems. CIRP Annals 56 (2007) 2, S. 712–729.

Scholz-Reiter, B.; Görges, M.; Philipp, T.: Autonomously controlled production systems—Influence of autonomous control level on logistic performance. CIRP Annals 58 (2009) 1, S. 395–398.

Schönemann, M.; Herrmann, C.; Greschke, P.; Thiede, S.: Simulation of matrix-structured manufacturing systems. Journal of Manufacturing Systems 37 (2015), S. 104–112.

Schuh, G.: Produktionsplanung und -steuerung: Grundlagen, Gestaltung und Konzepte. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg 2006.

Schuh, G.; Potente, T.; Wesch-Potente, C.; Weber, A.R.; Prote, J.-P.: Collaboration Mechanisms to Increase Productivity in the Context of Industrie 4.0. Procedia Cirp 19 (2014), S. 51–56.

Schuh, G.; Stich, V.; Gützlaff, A.; Reschke, J.; Cremer, S.; Steinlein, F.; Liu, Y.: Verbesserung der Liefertermintreue durch Simulation. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 114 (2019) 12, S. 819–822.

Schukraft, S.; Grundstein, S.; Scholz-Reiter, B.; Freitag, M.: Evaluation approach for the identification of promising methods to couple central planning and autonomous control. International journal of computer integrated manufacturing 29 (2016) 4, S. 438–461.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017).

Shahzad, A.; Mebarki, N.: Learning Dispatching Rules for Scheduling: A Synergistic View Comprising Decision Trees, Tabu Search and Simulation. Computers 5 (2016) 1, S. 3.

Sharma, P.; Jain, A.: Effect of routing flexibility and sequencing rules on performance of stochastic flexible job shop manufacturing system with setup times. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 231 (2016) 2, S. 329–345.

Shiue, Y.-R.; Lee, K.-C.; Su, C.-T.: Real-time scheduling for a smart factory using a reinforcement learning approach. Computers & Industrial Engineering 125 (2018), S. 604–614.

Shiue, Y.-R.; Lee, K.-C.; Su, C.-T.: A Reinforcement Learning Approach to Dynamic Scheduling in a Product-Mix Flexibility Environment. IEEE Access 8 (2020), S. 106542–106553.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; others: Mastering the game of go without human knowledge. Nature 550 (2017) 7676, S. 354.

Stricker, N.; Kuhnle, A.; Sturm, R.; Friess, S.: Reinforcement learning for adaptive order dispatching in the semiconductor industry. CIRP Annals 67 (2018) 1, S. 511–514.

Suhl, L.; Mellouli, T.: Optimierungssysteme. Berlin, Heidelberg: Springer Berlin Heidelberg 2013.

Sutton, R.S.; Barto, A.G.: Reinforcement learning: An introduction: MIT Press 2018.

Tay, J.C.; Ho, N.B.: Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. Computers & Industrial Engineering 54 (2008) 3, S. 453–473.

ten Hompel, M.; Vogel-Heuser, B.; Bauernhansl, T.: Handbuch Industrie 4.0. Berlin, Heidelberg: Springer Berlin Heidelberg 2020.

Trabs, M.; Jirak, M.; Krenz, K.; Reiß, M.: Statistik und maschinelles Lernen. Berlin, Heidelberg: Springer Berlin Heidelberg 2021.

Usuga Cadavid, J.P.; Lamouri, S.; Grabot, B.; Pellerin, R.; Fortin, A.: Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0. Journal of Intelligent Manufacturing 31 (2020) 6, S. 1531–1558.

Vepsalainen, A.P.; Morton, T.E.: Priority rules for job shops with weighted tardiness costs. Management Science 33 (1987) 8, S. 1035–1047.

Vinyals, O.; Babuschkin, I.; Czarnecki, W.M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D.H.; Powell, R.; Ewalds, T.; Georgiev, P.; Oh, J.; Horgan, D.; Kroiss, M.; Danihelka, I.; Huang, A.; Sifre, L.; Cai, T.; Agapiou, J.P.; Jaderberg, M.; Vezhnevets, A.S.; Leblond, R.; Pohlen, T.; Dalibard, V.; Budden, D.; Sulsky, Y.; Molloy, J.; Paine, T.L.; Gulcehre, C.; Wang, Z.; Pfaff, T.; Wu, Y.; Ring, R.; Yogatama, D.; Wünsch, D.; McKinney, K.; Smith, O.; Schaul, T.; Lillicrap, T.; Kavukcuoglu, K.; Hassabis, D.; Apps, C.; Silver, D.: Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature 575 (2019) 7782, S. 350–354.

Vinyals, O.; Ewalds, T.; Bartunov, S.; Georgiev, P.; Vezhnevets, A.S.; Yeo, M.; Makhzani, A.; Küttler, H.; Agapiou, J.; Schrittwieser, J.; (Keine Angabe): Starcraft ii: A new challenge for reinforcement learning. arXiv:1708.04782 (2017).

Voß, T.; Bode, C.; Heger, J.: Dynamische Losgrößenoptimierung mit bestärkendem Lernen. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 116 (2021a) 11, S. 815–819.

Voß, T.; Heger, J.; Meier, N.; Georgiadis, A.: Optimal robot scheduling of an AGV in the RCLL and introduction of team Leuphana. In: Robocup 2016, 2016,

Voß, T.; Rokoss, A.; Maier, J.T.; Schmidt, M.; Heger, J.: Outperformed by a Computer? - Comparing Human Decisions to Reinforcement Learning Agents, Assigning Lot Sizes in a Learning Factory. SSRN Electronic Journal (2021b).

Wang, H.: Flexible flow shop scheduling: optimum, heuristics and artificial intelligence solutions. Expert Systems 22 (2005) 2, S. 78–85.

Wang, Y.-C.; Usher, J.M.: Application of reinforcement learning for agent-based production scheduling. Engineering Applications of Artificial Intelligence 18 (2005) 1, S. 73–82.

Waschneck, B.; Reichstaller, A.; Belzner, L.; Altenmuller, T.; Bauernhansl, T.; Knapp, A.; Kyek, A.: Deep reinforcement learning for semiconductor production scheduling. In: 29th Annual SEMI Advanced 2018a, S. 301–306.

Waschneck, B.; Reichstaller, A.; Belzner, L.; Altenmüller, T.; Bauernhansl, T.; Knapp, A.; Kyek, A.: Optimization of global production scheduling with deep reinforcement learning. Procedia Cirp 72 (2018b) 1, S. 1264–1269.

Welch, P.D.: The statistical analysis of simulation results. The computer performance modeling handbook 22 (1983), S. 268–328.

Wiendahl; Lehnert: Variantenbeherrschung in der Montage. Berlin: Springer Berlin Heidelberg 2004.

Wiendahl, H.-P.: Fertigungsregelung: Logistische Beherrschung von Fertigungsabläufen auf Basis des Trichtermodells. München: Hanser 1997.

Wiendahl, H.-P.: Betriebsorganisation für Ingenieure. München: Hanser 2010.

Wiswede, G.: Motivation und Verbraucherverhalten: Grundlagen der Motivforschung. München, Basel.: Reinhardt 1973.

Zhang, F.; Mei, Y.; Zhang, M.: Genetic Programming with Multi-tree Representation for Dynamic Flexible Job Shop Scheduling. In: Mitrovic, T.; Xue, B.; Li, X. (Hrsg.): AI 2018. Cham: Springer 2018, S. 472–484.

Zhang, F.; Mei, Y.; Zhang, M.: A two-stage genetic programming hyper-heuristic approach with feature selection for dynamic flexible job shop scheduling. In: GECCO 2019: Proceedings of the Genetic and Evolutionary Computation Conference 2019, S. 347–355.