

Matthias Klar

**Automatisierte und
simulationsbasierte
Fabriklayoutplanung
mittels Reinforcement
Learning**

Produktionstechnische Berichte aus dem FBK
Band 2 | 2024

Herausgeber: Prof. Dr.-Ing. Jan C. Aurich

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Bibliographic information published by Die Deutsche Bibliothek

Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

Produktionstechnische Berichte aus dem FBK

Wissenschaftliche Schriftenreihe des
Lehrstuhls für Fertigungstechnik und Betriebsorganisation
der Rheinland-Pfälzischen Technischen Universität Kaiserslautern-Landau

Herausgeber: Lehrstuhl für Fertigungstechnik und Betriebsorganisation
Prof. Dr.-Ing. Jan C. Aurich
Postfach 3049
Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
67653 Kaiserslautern

Verlag: Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau

Druck: Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
Dezernat 5 Technik
Abteilung 5.6 Foto-Repro-Druck

D-386

© Lehrstuhl für Fertigungstechnik und Betriebsorganisation, 2024
Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
Erwin-Schrödinger-Straße
67663 Kaiserslautern

Alle Rechte vorbehalten, auch das des auszugsweisen Nachdrucks, der auszugsweisen oder vollständigen Wiedergabe (Photographie, Mikroskopie), der Speicherung in Datenverarbeitungsanlagen und das der Übersetzung.

Als Manuskript gedruckt. Printed in Germany.

ISBN 978-3-95974-226-9
ISSN 0937-9061

Automatisierte und simulationsbasierte Fabriklayoutplanung mittels Reinforcement Learning

Vom Fachbereich Maschinenbau und Verfahrenstechnik
der Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte

Dissertation

von

Herrn

Matthias Klar, M. Sc.

aus Mainz

Tag der mündlichen Prüfung: 06. Mai 2024

Promotionskommission:

Vorsitzende: Prof. Dr.-Ing. Kristin de Payrebrune

1. Berichterstatter: Prof. Dr.-Ing. Jan C. Aurich

2. Berichterstatter: Prof. Dr. Bahram Ravani

3. Berichterstatter: Prof. Dr.-Ing. Jürgen Fleischer

Kaiserslautern, 2024

Vorwort des Verfassers

"Progress cannot be generated when we are satisfied with existing situations."

/Taiichi Ohno/

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Fertigungstechnik und Betriebsorganisation der RPTU Kaiserslautern (FBK).

An dieser Stelle möchte ich mich bei Prof. Dr.-Ing. Jan C. Aurich für die Möglichkeit bedanken, meine Dissertation am FBK anzufertigen und als Oberingenieur des Teams Produktionssysteme den Lehrstuhl mitzugestalten und wertvolle Erfahrungen zu sammeln. Bedanken möchte ich mich auch für die persönlichen und fachlichen Entwicklungsmöglichkeiten, die mir durch zahlreiche Forschungsaufenthalte an der UC Davis und UC Berkeley im Rahmen des IRTG 2057 ermöglicht wurden. Hier gilt mein besonderer Dank Prof. Dr. Bahram Ravani für die intensive Zusammenarbeit. Weiterhin danke ich Prof. Dr.-Ing. Kristin de Payrebrune für den Vorsitz der Promotionskommission und Prof. Dr.-Ing. Jürgen Fleischer für die Übernahme der Berichterstattung.

Darüber hinaus möchte ich mich bei meinen Kolleginnen und Kollegen am FBK für die intensive und erfüllende Zeit bedanken, die mich persönlich geprägt und stets mit Begeisterung erfüllt hat. Insbesondere möchte ich mich beim Team Produktionssysteme für die hervorragende Arbeitsatmosphäre, die Freude an der Forschung und den offenen sowie engagierten Austausch bedanken, der maßgeblich zum Gelingen meiner Arbeit beigetragen hat. Weiterer Dank gilt meinem ehemaligen Teamleiter Dr.-Ing. Moritz F. Glatt, der mich jederzeit mit seiner ganzen persönlichen und fachlichen Erfahrung unterstützt und mir zusammen mit PD Dr.-Ing. habil. Benjamin Kirsch einen reibungslosen Übergang in die Teamleitung ermöglicht hat. Nicht zuletzt möchte ich mich stellvertretend bei Jonas Pollak, Michael Lutzke, Rosemarie Schleret und Britta Ziehmer für die wichtige Unterstützung im technisch-administrativen Lehrstuhlbetrieb und den persönlichen Austausch bedanken. Zusammenfassend hatte ich am FBK und im IRTG 2057 eine ganz besondere Zeit, in der viele Kolleginnen und Kollegen zu Freunden geworden sind.

Zum größten Dank bin ich jedoch meinen Eltern Beate und Dietmar sowie meinem Bruder Martin verpflichtet, deren Rückhalt und Verständnis mich dazu ermutigt haben, immer wieder über mich hinauszuwachsen. Ganz besonders möchte ich mich auch bei meiner Freundin Caro für ihre einfühlsame Unterstützung bedanken. Ihr habt einen großen Beitrag für meinen Werdegang und das Gelingen dieser Arbeit geleistet.

Kaiserslautern, im Mai 2024

Matthias Klar

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Zusammenfassung	III
Abstract	IV
English Summary	V
Abkürzungen.....	VI
1 Einleitung und Motivation.....	1
2 Stand von Wissenschaft und Technik.....	2
2.1 Grundlagen der Fabrikplanung.....	2
2.2 Fabriklayoutplanung.....	7
2.2.1 Ablauf und Zielsetzung der Fabriklayoutplanung.....	7
2.2.2 Simulationen als Bestandteil der Fabriklayoutplanung.....	12
2.2.3 Manuelle Fabriklayoutplanungsansätze.....	16
2.2.4 Automatisierte exakte und approximative Fabriklayoutplanungsansätze..	18
2.3 Bewertung von automatisierten Layoutplanungsansätzen	22
2.3.1 Anforderungsdefinition an automatisierte Layoutplanungsansätze	22
2.3.2 Zusammenfassende Bewertung.....	24
2.4 Reinforcement Learning	28
2.4.1 Grundlagen des maschinellen Lernens.....	28
2.4.2 Grundlagen des Reinforcement Learnings	31
2.4.3 Reinforcement Learning Agenten	34
2.4.4 Erklärbares Reinforcement Learning	38
2.4.5 Zusammenfassung der Leistungspotentiale von Reinforcement Learning Ansätzen	43
3 Zielsetzung und Aufgabenstellung.....	45
3.1 Erkenntnisse aus dem Stand von Wissenschaft und Technik.....	45
3.2 Forschungsziele und Aufbau der Arbeit.....	46
4 Modellierung des Fabriklayoutplanungsproblems	49
4.1 Definition der Systemgrenzen und zugehöriger Annahmen	49
4.2 Modellierung des Aktionsspielraums	50
4.3 Modellierung der Zustandsinformationen	52
4.4 Modellierung des Reinforcement Learning Agenten	53
4.5 Integration der Optimierungszielgrößen und Modellierung der Belohnungsfunktion	54
4.5.1 Integration analytisch formulierter Zielgrößen	56
4.5.2 Integration von EDS-Ergebnissen	60
5 Experimentelle Entwicklung des Reinforcement Learning Agent	64
5.1 Umgang mit ungültigen Aktionen	64
5.2 Festlegung des Reinforcement Learning Agent Typs	66
5.3 Festlegung der Agentenstruktur zur multikriteriellen Optimierung	73
6 Erklärbares Reinforcement Learning für die Fabriklayoutplanung	79
6.1 Definition der Zielgrößen.....	79
6.2 Entwicklung der XRL-Ansätze	80
6.2.1 Policy Summarization	80
6.2.2 Occlusion-Sensitivitätsansätze	82
7 Zusammenfassendes Konzept zur Reinforcement Learning basierten Fabriklayoutplanung.....	86
8 Validierung.....	93
8.1 Umplanungsszenario	93
8.2 Neuplanungsszenario.....	105
8.3 Diskussion und Schlussbetrachtung	109

9	Zusammenfassung und Ausblick.....	112
10	Quellenverzeichnis	115
11	Anhang	129
11.1	Anhang A: Schematischer Aufbau des Simulationsmoduls	129
11.2	Anhang B: Abbildungen zur Anwendung des RL-Ansatzes auf ein Neuplanungsproblem	130
11.3	Anhang C: Übersicht über die wichtigsten Python-Bibliotheken.....	136
11.4	Anhang D: Übersicht über die realisierbaren Ergebnisse durch Transfer Lernen	137

Zusammenfassung

Die Fabriklayoutplanung ist ein auf Expertenwissen basierender, kreativer und zeitintensiver Prozess, der einen erheblichen Einfluss auf die Wirtschaftlichkeit und den Betrieb des Produktionssystems hat, sodass in der frühen Layoutplanungsphase häufig automatisierte Methoden als Unterstützungswerkzeug eingesetzt werden. Diese erfordern jedoch häufig eine analytische und vereinfachende Beschreibung des Produktionssystems, wodurch dynamische und layoutbestimmende Effekte unberücksichtigt bleiben. Mittels einer Materialflusssimulation (EDS) können diese Einflüsse realitätsnah abgebildet werden. Entwicklungen im Bereich des Reinforcement Learning (RL) zeigen, dass RL in der Lage ist, selbstständig Lösungsstrategien für komplexe Problemstellungen zu erlernen, deren Systemverhalten analytisch nicht vollständig beschreibbar ist. Die Kopplung von RL und EDS bietet somit das Potential, durch eine realitätsnahe Planungsgrundlage automatisiert optimierte Layoutvarianten zu erzeugen. Außerdem können Erklärbarkeitsansätze (XRL) genutzt werden, um Einblicke in das Lösungsverhalten des RL-Ansatzes zu erlangen. Ziel dieser Dissertation ist es, durch die kombinierte Nutzung von RL, EDS und XRL die Qualität in der frühen Layoutplanungsphase zu erhöhen. Hierzu findet eine konzeptionelle Entwicklung des RL-Ansatzes statt, die eine Modellierung des Entscheidungsproblems, der Aktionsräume, der Zustandsinformationen, der Belohnungsfunktion und der XRL-Ansätze umfasst. Anschließend wird der RL-Ansatz implementiert und es werden Experimente zur Leistungssteigerung durchgeführt, bevor er auf zwei Validierungsszenarien angewendet wird. Der Vergleich mit konventionellen Planungsansätzen zeigt, dass die Qualität der generierten Lösung durch RL gesteigert werden kann und dass die XRL-Ansätze eine detaillierte Analyse des Lösungsprozesses ermöglichen.

Abstract

Factory layout planning is a creative and time-consuming process based on expert knowledge that has a significant impact on the profitability and operation of the resulting manufacturing system. As a result, automated approaches are often used to support the early layout planning phase. However, these often require an analytical and simplified description of the production system, so that dynamic and layout-determining effects are not considered. These effects can be represented with the help of a discrete-event material flow simulation (DES). Besides, developments in the field of reinforcement learning (RL) show that RL can independently learn solution strategies for complex decision-making problems even in cases in which the system behavior cannot be fully described analytically. The coupling of RL and DES thus offers the potential to automatically generate realistic and optimized layout variants. In addition, explainability approaches (XRL) can be used to gain insights into the solution strategy of the RL approach. This dissertation aims to increase the quality of the early layout planning phase through the combined use of RL, EDS, and XRL. For this purpose, a conceptual development of the RL approach takes place first, which includes modeling the decision-making problem, the action space, the state information, the reward function, and the XRL approaches. The RL approach is implemented and experiments are conducted to improve the performance before it is applied to two validation scenarios. The comparison with conventional layout planning approaches shows that the quality of the generated solution can be increased using RL and that the XRL approaches allow a detailed analysis of the training and solution process.

English Summary

Factory layout planning requires experienced personnel as it is a creative and time-consuming process in which various alternatives must be generated taking into account multiple, often conflicting, planning objectives. The result has a direct impact on the operational characteristics of the factory. Therefore, it is important to ensure high quality planning. Due to the shortage of planners who are up to the task, automated approaches aim to support the planner in the early planning phase by generating optimized layouts that serve as a planning basis. However, approximate automated methods often require an analytical and simplified description, which means that dynamic and layout-determining effects of a production system are not taken into account. These effects can be realistically represented by a discrete event material flow simulation.

Developments in the field of artificial intelligence have shown that reinforcement learning (RL) is a powerful approach that can independently learn solution strategies even for complex problems whose system behavior cannot be fully described analytically. The coupling of RL and discrete event simulation thus offers the unexplored potential to automatically generate optimized layout variants, taking into account dynamic and layout-determining effects. In addition, it is expected that the trustworthiness of the generated layouts can be increased by methods of explainable artificial intelligence.

This dissertation aims to address these research gaps by presenting a multi-objective, simulation, and RL based factory layout planning approach. In the first step, the factory layout planning problem is modeled as an RL compatible sequential decision-making process, including the system boundaries, the agent, and the environment structure. Subsequently, the objective functions are defined, and a discrete event simulation model is developed, resulting in an operational RL approach. To improve the performance of the approach, the agent type, its hyperparameters, and the structure to perform multi-objective optimization are defined based on experiments. The main body of this dissertation concludes with the introduction of explainability methods and the validation of the approach based on two industrial application scenarios. The results demonstrate the capabilities of RL for factory layout planning. Specifically, RL can be used to generate different layout variants while taking into account discrete event simulation results. Furthermore, it is shown that the quality of the generated solutions exceeds the quality of conventional metaheuristics. Another advantage compared to conventional metaheuristics is the ability of RL to learn a policy that can be analyzed after training. This makes it possible to use explainability methods to evaluate trustworthiness and to analyze the training and solution process in detail, which is not possible using conventional metaheuristics.

Abkürzungen

A2C	Advantage Actor Critic	NSGA	Non-dominated Sorting Genetic Algorithm
AR	Augmentierte Realität		
bzw.	beziehungsweise	PER	Prioritized Experience Replay
ca.	circa	PPO	Proximal Policy Optimization
CNN	Convolutional Neural Network	PSFU	Funktionseinheitenbezogen Policy Summarization
CORELAP	Computerized Relative Layout Planning	PSL	Layoutbezogene Policy Summarization
CRAFT	Computerized Relative Allocation of Facilities		
d	diskret	Rainbow	Rainbow Deep Q Learning
DDQN	Double Deep Q Learning	DQN	
DIN	Deutsches Institut für Normung e.V.	RL	Reinforcement Learning
EDS	Ereignisdiskrete Materialflusssimulation	SA	Simulated Annealing
FiFo	First-In-First-Out	SAC	Soft Actor Critic
FLP	Facility Layout Problem	SDG	Stochastic Gradient Decent
GA	Genetischer Algorithmus	SL	Supervised Learning
GANN	Graph Attention Neural Network	TCR	Total Closeness Rating
GCN	Graph Convolutional Network	TD-Fehler	Temporal-Differenz Fehler
GNN	Graph Neural Network	TRPO	Trust Region Policy Optimization
GUI	Graphical User Interface	TS	Tabu Suche
KI	Künstliche Intelligenz	u. a.	unter anderem
KNN	Künstliches Neuronales Netz	USL	Unsupervised Learning
KVP	Kontinuierlicher Verbesserungsprozess	vgl.	vergleiche
md	multidiskret	VPG	Vanilla Policy Gradient
mk	multikontinuierlich	VR	Virtuelle Realität
ML	Maschinelles Lernen	WA	Warenausgang
MLP	Multilayer Perceptron	WE	Wareneingang
		XAI	Erklärbare Künstliche Intelligenz
		XRL	Erklärbares Reinforcement Learning
		z. B.	zum Beispiel

1 Einleitung und Motivation

Die Fabriklayoutplanung als zentraler Bestandteil der Fabrikplanung befasst sich mit den grundlegenden Aspekten der Gestaltung und Optimierung von Produktionssystemen. Im Kern umfasst dies die geometrische Anordnung der Funktionseinheiten einer Fabrik, wodurch die Fabriklayoutplanung einen erheblichen Einfluss auf die Wirtschaftlichkeit und den Betrieb des resultierenden Produktionssystems hat. Folglich ist eine hohe Qualität des Planungsprozesses von Relevanz [Grun21].

Zur Sicherstellung der Planungsqualität werden in der frühen Fabriklayoutplanung häufig unterschiedliche Layoutvarianten entworfen. In diesem Zusammenhang ist es notwendig, Konflikte zwischen Planungszielen und Positionierungsoptionen zu berücksichtigen. Traditionell werden dazu manuelle, auf Erfahrungswissen basierende Verfahren eingesetzt [Pawe14]. Da die Durchführung einer manuellen Planung häufig zeitintensiv ist und die generierten Layoutvarianten meist nach subjektiven Aspekten bewertet werden, haben automatisierte Verfahren an Bedeutung gewonnen. Diese ermöglichen die Generierung von Layoutvarianten in einer frühen Planungsphase, wodurch die Planungszeit reduziert und eine bessere Planungsgrundlage sichergestellt werden kann [Hoss18].

Approximative, automatisierte Verfahren führen auch bei größeren Problemdimensionen zu praktikablen Lösungen, erfordern jedoch häufig eine analytische und vereinfachende Beschreibung, wodurch dynamische und layoutbestimmende Effekte eines Produktionssystems unberücksichtigt oder unbewertet bleiben. Mit Hilfe von ereignisdiskreten Materialflusssimulationen (EDS) können diese Einflüsse realitätsnah simuliert werden [Burg21a].

Entwicklungen im Bereich der Künstlichen Intelligenz (KI) haben gezeigt, dass sich Reinforcement Learning (RL) als ein leistungsfähiger Ansatz erweist, der selbstständig Lösungsstrategien auch für komplexe Problemstellungen erlernt, deren Systemverhalten in der Praxis analytisch nicht oder nur mit erheblichem Zeitaufwand vollständig beschreibbar ist [Sutt18, Silv16]. Die Kopplung von RL und EDS bietet das Potential, durch eine realitätsnahe Planungsgrundlage automatisiert Layoutvarianten zu generieren, die eine hohe Wirtschaftlichkeit des Produktionssystembetriebs ermöglichen. Darüber hinaus ist zu erwarten, dass die Nachvollziehbarkeit der generierten Layoutvarianten durch den Einsatz von Methoden der Erklärenden Künstlichen Intelligenz (XAI) signifikant erhöht werden kann. Die Entwicklung eines RL-basierten Fabriklayoutplanungsansatzes und die Untersuchung der erlernten Strategie mit Methoden der XAI sind Gegenstand dieser Dissertation.

2 Stand von Wissenschaft und Technik

In diesem Kapitel werden die Grundlagen im Zusammenhang mit dem Forschungsschwerpunkt dieser Dissertation beschrieben. Als thematische Einführung werden in Kapitel 2.1 die Grundlagen der Fabrikplanung erörtert. Ein zentraler Bestandteil der Fabrikplanung ist die Fabriklayoutplanung, die in Kapitel 2.2 charakterisiert wird. Dabei wird neben den Planungszielen auf die Simulation als Bewertungskriterium eingegangen und bestehende Ansätze zur manuellen und automatisierten Fabriklayoutplanung werden beschrieben. Letztere werden in Kapitel 2.3 anhand von Anforderungskategorien bewertet und Forschungsbedarfe abgeleitet. Darauf aufbauend werden in Kapitel 2.4 die Grundlagen und Leistungspotentiale von RL erläutert und die zu adressierenden Entwicklungsschritte für diese Dissertation definiert.

2.1 Grundlagen der Fabrikplanung

Die Fabrikplanung kann als „systematischer, zielorientierter, in aufeinander aufbauende Phasen strukturierter und unter Zuhilfenahme von Methoden und Werkzeugen durchgeführter Prozess zur Planung einer Fabrik von der Zielfestlegung bis zum Hochlauf der Produktion“ definiert werden [VDI 11a]. Das Ziel der Fabrikplanung besteht darin, die Fabrik unter Berücksichtigung der veränderlichen Rahmenbedingungen so zu planen [Warn95], dass eine wirtschaftliche, effiziente, flexible und wandlungsfähige Produktion realisiert werden kann [Grun21]. Eine Fabrik umfasst im Kern ein Produktionssystem, das aus systemtechnischer Sicht alle Elemente, die zur Herstellung eines Produktes notwendig sind, und deren Relationen, Abhängigkeiten und Wechselwirkungen umfasst (siehe Abbildung 2-1) [Ever99, Ever96].

Die Struktur der Fabrik wird durch die Gesamtheit aller Relationen definiert und umfasst u. a. die räumlichen Relationen zwischen Bereichen, die zeitlichen Relationen von Abläufen, die inhaltlich-sachlichen Relationen von Produkten und die inhaltlich-logistischen Relationen von Prozessen [Pawe14]. Der Fabrikplanungsprozess ist kein einmalig durchzuführender Planungsprozess, sondern kann als ein fortlaufender Verbesserungs- und Anpassungsprozess angesehen werden, der die iterativ ablaufende Analyse-, Planungs- und Realisierungsphase umfasst [Domb10, Warn95]. Während des Fabrikplanungsprozesses müssen vielfältige Planungsaufgaben aus unterschiedlichen Planungsfeldern [Feli98] unter Nutzung von Werkzeugen, Methoden und Modellen durchgeführt werden [VDI 11a]. Je nach Planungsinhalt ist die Expertise verschiedener Abteilungen im Unternehmen erforderlich, weshalb die Zusammensetzung des Planungsteams während des Fabrikplanungsprozesses variiert [Burg21b]. Ferner sind die Planungsinhalte durch teilweise widersprüchliche Zielsetzungen gekennzeichnet, deren Erreichung über die Lebensdauer der Fabrik sichergestellt werden soll [Wien14a]. Da der spätere Produktionsbetrieb maßgeblich durch den Fabrikplanungsprozess bestimmt wird, ist eine strukturierte Vorgehensweise unerlässlich [Burg21b].

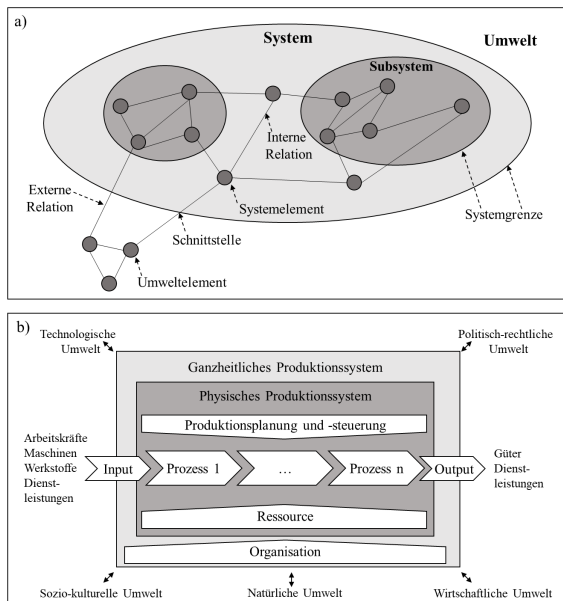


Abbildung 2-1: Definition eines Systems gemäß der Systemtheorie (a) und Definition des Produktionssystems (b) [Habe12, Günt09]

Übergeordnet können die fünf Planungsgrundfälle Neubau, Umgestaltung, Erweiterung, Rückbau und Revitalisierung definiert werden (siehe Abbildung 2-2). Im Rahmen der *Neuplanung*, auch Greenfield-Planung genannt, wird eine Fabrik vollständig neu geplant [Grun21]. Da das Gelände für den Neubau der Fabrik ungenutzt ist, bestehen hohe Freiheitsgrade in der Fabrikgestaltung. Allerdings müssen bei diesem Planungsfall zusätzliche Aufgaben, wie die Standortplanung, durchgeführt werden. Bei der *Umplanung*, auch Brownfield-Planung genannt, wird eine bestehende Fabrik umgeplant. Daher müssen bereits definierte Fabrikstrukturen und bestehende räumliche, technologische und organisatorische Rahmenbedingungen bei der Umgestaltung einbezogen werden, um die Fabrik an veränderte Anforderungen anzupassen [Burg21b]. Während bei der Erweiterungsplanung neue Kapazitäten geschaffen und in den bestehenden Fertigungsablauf integriert werden, fokussiert sich die Rückbauplanung auf eine Kapazitätsreduktion und eine Verringerung der Fertigungstiefe [Kett84]. In beiden Fällen steht eine Steigerung der Wirtschaftlichkeit der Produktion im Vordergrund. Im Rahmen der Revitalisierung werden Sanierungsprozesse durchgeführt, um stillgelegte Bereiche und Anlagen erneut zu nutzen [Grun21].

Zur zielgerichteten Umsetzung des Fabrikplanungsprozesses wurden verschiedene Vorgehensmodelle entwickelt. Diese können in klassische, sequentielle Fabrikplanungsmodelle und in moderne, nichtsequentielle Fabrikplanungsmodelle unterteilt werden [Sinn20]. Tabelle 2-1 gibt eine Übersicht über zugehörige Ansätze.

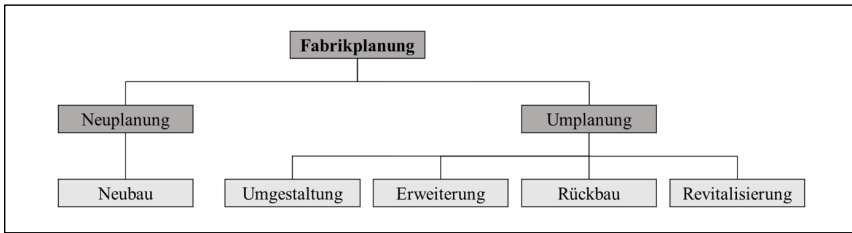


Abbildung 2-2: Planungsgrundfälle nach [Burg21b]

Tabelle 2-1: Übersicht über klassische und moderne Fabrikplanungsverfahren

Klassische Fabrikplanungsmodelle	Moderne Fabrikplanungsmodelle
Leitfaden der systematischen Fabrikplanung [Kett84]	Objektorientierte Fabrikplanung [Berg06]
Fabrikplanung [Grun21]	Fabrikplanung im Gegenstromverfahren [Schu07]
Fabrikplanung [Aggt87]	Zustandsbasierte Fabrikplanung [Nöck12]
Handbuch Fabrikplanung [Wien14a]	Das Aachener Fabrikplanungsverfahren [Kamp14]
Systematic design of assembly systems [Bell98]	Architecture Framework for Manufacturing System Design [Benk14]
Unternehmens- und Fabrikplanung [Feli98]	Synergetische Fabrikplanung [Wien14b]
Systematic planning of industrial facilities [Muth99]	Integrative Fabrikplanung durch effiziente Koordinationsmodelle [Meck15]
Planung und Gestaltung komplexer Produktionssysteme [REFA90]	
Projektierung von Industriebetrieben [Rock63]	
Produktionsstrategie im Produktlebenszyklus [Auri04]	
Planung und Betrieb wandlungsfähiger Logistiksysteme in der variantenreichen Serienproduktion [Dürr01]	
PDCA als Methode der qualitäts- und zielorientierten Fabrikplanung [Urba09]	
Facilities planning [Tomp10]	
Objektorientierte Fabrikplanung [Berg06]	
Verteilte kooperative Fabrikplanung [Auri05]	
Innovative und effiziente Fabrikplanung [Kerk14]	
Ganzheitliche Fabrikplanung [Pawe14]	
Planung von Materialflusssystemen [Amo19]	
Fabrikplanung – Planungsverfahren [VDI 11a]	
Improving Factory Planning by Analyzing Process Dependencies [Büsc14]	

Klassische Fabrikplanungsmodelle

Die klassischen, analytischen Fabrikplanungsmodelle sind in aufeinanderfolgende Phasen aufgeteilt. Die genaue Aufteilung und Bezeichnung der Phasen unterscheiden sich quellenabhängig, beinhalten aber ähnliche Planungsaufgaben mit den übergeordneten Phasen: Vorbereitung, Grobplanung, Feinplanung, Ausführungsplanung und Ausführung [Schu07].

Da die Inhalte der klassischen Fabrikplanungsmodelle ähnlich sind, wird nachfolgend beispielhaft das Modell nach KETTNER ET AL. [Kett84] und GRUNDIG [Grun21] dargestellt. Die Autoren unterteilen das Fabrikplanungsvorgehen in sieben Teilschritte (siehe Abbildung 2-3), die als Grundlage für die Erklärungen der Fabriklayoutplanung in Kapitel 2.2 dienen.

In der *Zielplanung* werden hierarchie- und abteilungsübergreifende Ziel- und Aufgabenstellungen für das Fabrikplanungsprojekt auf Basis von Analysen des Marktes und interner sowie externer Rahmenbedingungen definiert. Die Ergebnisse der Zielplanung werden in der *Vorplanung* aufgegriffen, in der der aktuelle Zustand der Fabrik analysiert wird. Hierbei werden u. a. das Produktionsprogramm, Bedarfe und Aufwände analysiert, um die Zielsetzung des Planungsprojekts zu konkretisieren und mögliche Lösungsprinzipien erstmalig zu definieren. Als Ergebnis sind am Ende der Phase die Rahmenbedingungen für alle Folgeaktivitäten festgelegt. Die Grobplanungsphase wird in die Ideal- und Realplanung unterteilt. Die Zielsetzung der Grobplanungsphase besteht u. a. in der Erstellung eines ersten groben Fabriklayouts, das in Form eines Blocklayouts dargestellt werden kann. Bei der *Idealplanung* werden Flächen-, Raum- und Personalbedarfe dimensioniert und strukturiert. Zudem werden Bereiche gebildet und ein idealisiertes flächenmaßstäbliches Funktionsschema (Ideallayout) erstellt. Die *Realplanung* berücksichtigt im Vergleich zur Idealplanung die Restriktionen des Realsystems. Das Ideallayout wird unter Berücksichtigung von Grundstück- und Gebäuderestriktionen, Raum- und Flächenbedarfe, funktioneller Kriterien und des Budgets in ein Reallayout, auch Groblayout genannt, überführt. Während der Grobplanungsphase werden unterschiedliche Varianten des Reallayouts generiert und miteinander verglichen, um eine bestmögliche Ausgangssituation für die anschließende Feinplanung zu gewährleisten. Folglich werden bereits im Rahmen der Grobplanung wichtige Charakteristika des zukünftigen Produktionssystems festgelegt. Die *Feinplanung* baut auf der Grobplanung auf und berücksichtigt zusätzlich alle Details, die während der Grobplanungsphase nicht berücksichtigt wurden. Zu den Aufgaben der Feinplanung gehört die Erstellung des final umzusetzenden Fabriklayouts. U. a. werden Arbeitsplätze detailliert ausgeplant, Betriebsmittel final angeordnet und eine Feinabstimmung zwischen Bearbeitungs-, Förder- und Lagertechnik durchgeführt. Des Weiteren werden in der Feinplanungsphase bereits erste Vorbereitungen zur Umsetzung der Planung umgesetzt, indem der finale Planungsstand dokumentiert und Genehmigungsverfahren angestoßen werden. Während der *Ausführungsplanung* wird die Umsetzung aller Bau-, Montage-, Installations-, Einrichtungs- und Inbetriebnahmetätigkeiten entsprechend der zur Verfügung stehenden Kapazitäten terminiert und ein Projektteam mit Projektleitung gebildet. Die Projektleitung führt und überwacht in der *Ausführungsphase* die Umsetzung des Projekts, lässt Probelaufe, Abnahmeprüfungen und Mitarbeiterschulungen durchführen und steuert das Projekt zu einer erfolgreichen Inbetriebnahme der Fabrik [Grun21, Kett84].

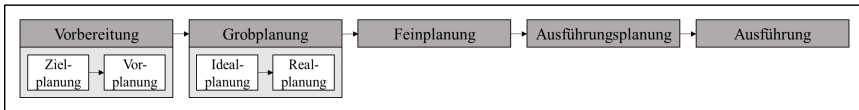


Abbildung 2-3: Fabrikplanungsmodell nach [Grun21] und [Kett84]

Moderne Fabrikplanungsmodelle

Moderne, nichtsequentielle Fabrikplanungsmodelle unterscheiden sich von den klassischen Fabrikplanungsmodellen in der grundsätzlichen Vorgehensweise. Die durchzuführenden Planungsaufgaben sind jedoch ähnlich. Bei klassischen Fabrikplanungsmodellen bilden die Ergebnisse der ersten Planungsschritte den Rahmen für nachfolgende Planungsschritte. Die Planung der Fabrik erfolgt anhand eines Top-Down-Ansatzes, in dem der Detailgrad während des Planungsvorgangs schrittweise zunimmt [Pawe14]. In modernen Fabrikplanungsmodellen wird die strenge sequentielle Abfolge von Planungsschritten teilweise aufgelöst, sodass der Planungsprozess auch während der Bearbeitung an die Dynamik im Fabrikumfeld angepasst werden kann [Kamp14]. Daher werden neben Top-Down- auch Bottom-Up-Ansätze genutzt. Hierbei werden einzelne Teilbereiche detailliert geplant und anschließend in eine Gesamtlösung eingebracht. Die Lösungen für die Teilbereiche können ideal ausgeplant werden. Der Aufwand und die Komplexität der Schnittstellenbetrachtung zwischen allen Teilbereichen steigt jedoch mit der Größe des Planungsprojekts. Daher sind reine Bottom-Up Ansätze für ganzheitliche Planungsaufgaben nicht geeignet [Pawe14].

Das *Fabrikplanungsverfahren im Gegenstromverfahren* nutzt eine Kombination des Top-Down- und Bottom-Up-Ansatzes und ist in Abbildung 2-4 dargestellt. Hierbei wird ähnlich zu den klassischen Fabrikplanungsmodellen, basierend auf den gegebenen Rahmenbedingungen, ein Idealkonzept generiert, das eine übergreifende Strukturierung der Prozesse und Funktionsbereiche umfasst. Dieses Idealkonzept wird durch teilbereichsbezogene Planungskonzepte ergänzt, die durch eine Bottom-Up-Methodik erstellt wurden. Für jeden Teilbereich werden Stärken-Schwächen Analysen durchgeführt und Veränderungsbedarfe abgeleitet, sodass ein Soll-Konzept erarbeitet werden kann. Das Idealkonzept aus dem Top-Down-Ansatz und die Soll-Konzepte des Bottom-Up-Ansatzes werden anschließend unter Berücksichtigung der Planungsrestriktionen aggregiert und in Fabriklayoutvarianten überführt. Die Layoutvarianten werden miteinander verglichen und das vielversprechendste Layout wird anschließend im Roll-Out bis zur finalen Umsetzung begleitet. Durch die kombinierte Betrachtung von Top-Down- und Bottom-Up-Ansätzen sollen die Vorteile beider Ansätze genutzt werden, um übergeordnete Zielsetzungen und optimierte Teillösungen in gleicher Weise zu berücksichtigen [Schu07].

Ein weiteres modernes Planungsmodell ist das *Aachener Fabrikplanungsverfahren*, bei dem der Planungsprozess unter Nutzung standardisierter Planungsmodule frei konfiguriert werden kann [Schu11, Schu09]. Die Konfiguration des Planungsprozesses erfolgt im Kontext des Lean Managements, sodass Planungsmodule nur dann verwendet werden, wenn sie für die Fabrikplanungsaufgabe benötigt werden [Kamp14]. Die Bearbeitung der einzelnen Planungsmodule erfolgt nicht immer sequentiell, sondern auch parallel. Planungsinformationen werden zwischen den Modulen bereits während der Bearbeitung iterativ mittels Schnittstellen ausgetauscht [Nöck12]. Weitere Planungsmodelle wie die *integrative Fabrikplanung durch*

effiziente Koordinationsmodelle nach [Meck15] oder die zustandsbasierte Fabrikplanung nach [Nöck12], bauen auf dem Aachener Fabrikplanungsvorgehen auf.

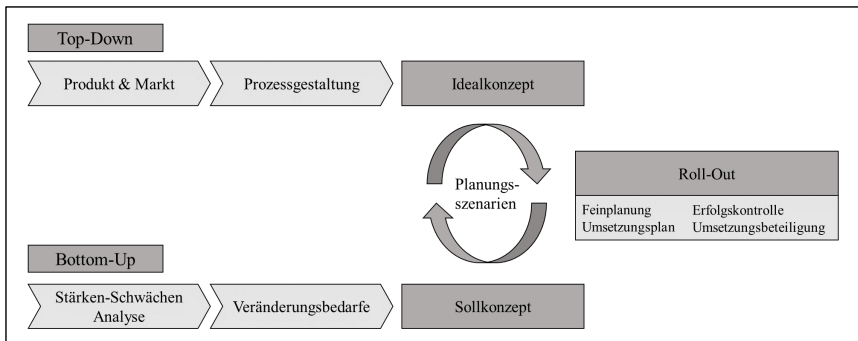


Abbildung 2-4: Fabrikplanungsvorgehen im Gegenstromverfahren nach [Schu07]

2.2 Fabriklayoutplanung

Die Fabriklayoutplanung kann als ein zentraler Bestandteil der Fabrikplanung angesehen werden, greift die Ergebnisse der vorgegangenen Planungsphasen, wie z. B. die Flächenbedarfsermittlung und -dimensionierung, auf, ordnet einzelne Funktionsbereiche, auch Funktionseinheiten genannt, räumlich an und verknüpft diese materialflussgerecht [Pawe14]. Diese Anordnung wird als Fabriklayout bezeichnet [VDI 08, Schm96]. Die Qualität des geplanten Fabriklayouts entscheidet maßgeblich über die Leistungsfähigkeit des zukünftigen Produktionssystems [Burg21b].

Da die Fabriklayoutplanung ein zentraler Schwerpunkt dieser Dissertation ist, werden in Kapitel 2.2.1 der Ablauf und die Zielsetzungen der Fabriklayoutplanung erläutert. Der Materialfluss und dessen Einfluss auf die Leistungsfähigkeit des Produktionssystems ist für die Fabriklayoutplanung von besonderer Bedeutung. Daher wird in Kapitel 2.2.2 dargestellt, wie Materialflusssimulationen zur Verbesserung der Ergebnisqualität der Fabriklayoutplanung genutzt werden können. In den Kapiteln 2.2.3 und 2.2.4 werden Ansätze zur Layoutgenerierung in der frühen Phase der Fabriklayoutplanung beschrieben.

2.2.1 Ablauf und Zielsetzung der Fabriklayoutplanung

Wie zuvor beschrieben wurde, ist das Ziel der Grobplanungs- und Feinplanungsphase die Erstellung des Fabriklayouts [Grun21, Kett84]. In Analogie zur Fabrikplanung ist die Anpassung oder Neugestaltung eines Fabriklayouts immer dann notwendig, wenn sich die Rahmenbedingungen bzw. Einflussfaktoren ändern. Zu den Einflussfaktoren zählen u. a. die Einführung neuer Produkte oder Produktvarianten, technologische Weiterentwicklungen, Änderungen der Markt- und Kundenanforderungen, Arbeitszeitveränderungen oder strategische Anpassungen, die eine Veränderung des Beschaffungsprinzips oder der Fertigungstiefe bedingen [Pawe14].

Die im Rahmen des Fabriklayoutplanungsprozesses generierten Ergebnisse lassen sich gemäß Abbildung 2-5 in Groblayoutvarianten, die Ideal- und Reallayoutvarianten umfassen, und Feinlayoutvarianten unterteilen. Der Detaillierungsgrad der Ausarbeitung nimmt im Verlauf

der Fabriklayoutplanung zu, sodass am Ende des Layoutplanungsprozesses alle Gebäude, Funktionseinheiten und Arbeitsplätze materialflussgerecht angeordnet sind [Burg21b].

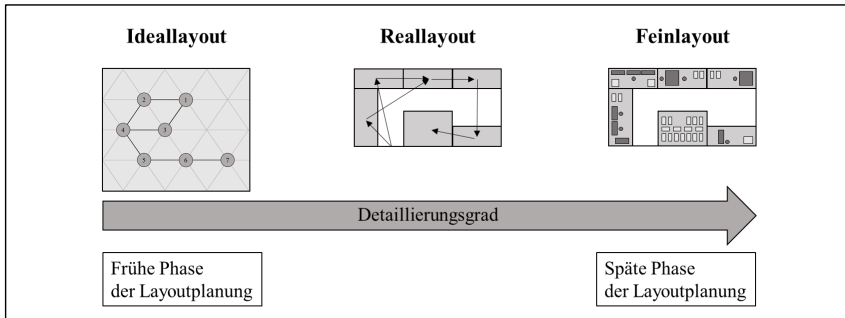


Abbildung 2-5: Steigerung des Detaillierungsgrad im Fabriklayoutplanungsprozess nach [Grun21]

Groblayout

Die Generierung des Groblayouts, auch Strukturierung genannt, ist Bestandteil der Grobplanungsphase [Kett84]. Als Ausgangspunkt dienen Informationen und Rahmenbedingungen aus verschiedenen Planungsbereichen. Erste Rahmenbedingungen werden bereits während der Werkstrukturplanung festgelegt, in der neben dem Generalbebauungsplan und dem Organisationskonzept der Fabrik auch das Werksstrukturlayout definiert wird. Das Werksstrukturlayout stellt eine grobe Abbildung der Werksbebauung dar und umfasst Grundstücks- und Gebäudegrenzen sowie Wege und Versorgungsleitungen. Zur Grobplanung des Layouts für ein ausgewähltes Werksgebäude werden u. a. die Ergebnisse der Flächenbedarfsplanung, auch Dimensionierung genannt, mit einbezogen. In dieser werden die Flächenbedarfe für alle zu positionierenden Funktionseinheiten wie Lagerflächen oder Flächenbedarfe für Maschinen definiert. Darüber hinaus werden Informationen aus dem Produktionsprogramm bei der Fabriklayoutplanung berücksichtigt. Dies umfasst die Menge der geplanten Produktion und kann genutzt werden, um die Kapazitätsbedarfe an Betriebsmitteln und Personal zu ermitteln [Burg21b]. Die Ergebnisse der Dimensionierung und das geplante Produktionsprogramm fließen in die Materialflussplanung ein, in der berücksichtigt wird, wie einzelne Bereiche diesbezüglich zu verknüpfen sind. Die Ergebnisse der Materialflussplanung sind eine wichtige Grundlage für die Grobplanung von Fabriklayouts, da Funktionseinheiten so angeordnet werden sollen, dass der Materialfluss bestmöglich optimiert wird [Pawe14] und ein funktionsgerechtes, kostengünstiges und menschengerechtes Fabriksystem entsteht [Enge84]. Eine Herausforderung besteht darin, dass die Materialflussplanung, die Flächenbedarfsplanung und die Grobplanung des Layouts teilweise parallel ablaufen, weshalb aufgrund gegenseitiger Abhängigkeiten der Planungsfelder und fehlender Informationen teils ungenaue Zwischenergebnisse anfallen, die in Verbesserungsschleifen konkretisiert werden [Burg21b].

Üblicherweise wird zur Erstellung des Groblayouts ein zweistufiges Vorgehen gewählt, bei dem zunächst ein Ideallayout generiert wird, das anschließend in ein Reallayout überführt wird. Zur Generierung eines *Ideallayouts* werden weder räumliche Restriktionen, die z. B. durch das Gebäude vorgegeben sind, noch Dimensionen der Funktionseinheiten selbst berücksichtigt

[Grun21]. Ein beispielhaftes Ideallayout, das mittels des Dreieckprobiervfahrens erstellt wurde, ist in Abbildung 2-5 dargestellt.

Zur Erstellung eines Ideallayouts können verschiedene Methoden genutzt werden, die in Kapitel 2.2.3 näher erläutert werden. Grundsätzlich werden die Funktionseinheiten der Produktions- und Logistikabschnitte hinsichtlich ihrer Materialfluss- und Transportbeziehungen analysiert und so angeordnet, dass eine möglichst wirtschaftliche Gesamtlösung entsteht [Pawe14]. Hierfür werden Transportmatrizen analysiert und Funktionseinheiten mit starker materialflussesseitiger Verknüpfung nebeneinander platziert [Schm70]. Ferner können weitere formelle und informelle Beziehungen aus den Kategorien Informations-, Material- und Personalflüsse berücksichtigt werden [VDI 11a]. Eine weitere Möglichkeit zur groben Strukturierung stellt die Gestaltung des Layouts entsprechend des Fertigungsprinzips und basierend auf dem Fertigungstyp (Einzel-, Serien- oder Massenfertigung) dar. In Abhängigkeit vom Fertigungstyp, den damit einhergehenden Stückzahlen und der Anzahl an Produktvarianten kann ein passendes Fertigungsprinzip zugeordnet werden, das mit den zugehörigen räumlichen Strukturierungs-, zeitlichen Bewegungs- und grundlegenden Organisationsprinzipien verknüpft ist (vgl. Tabelle 2-2 und Abbildung 2-6) [Günt09]. Da nur selten ein einziges, optimales Ideallayout existiert, werden alternative Ideallayoutvarianten entwickelt, die den Ausgangspunkt für die Generierung eines oder mehrerer Reallayouts darstellen [Harm04].

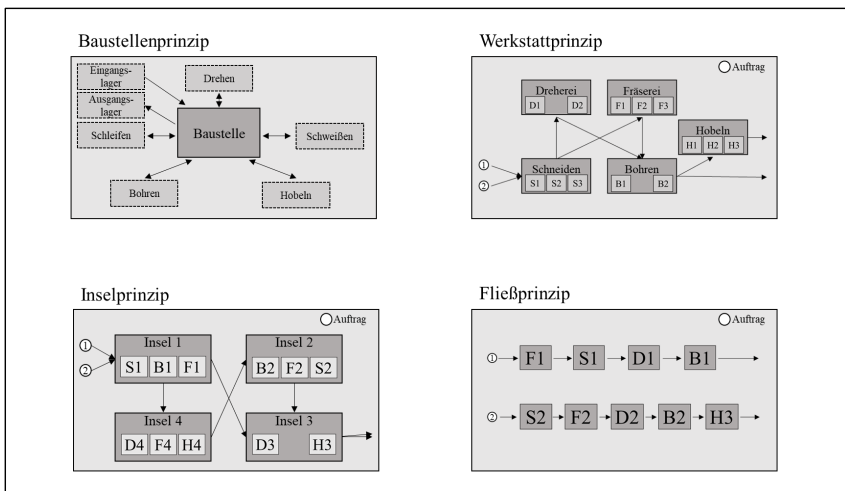


Abbildung 2-6: Überblick über die Fertigungsprinzipien nach [Wien14a]

Tabelle 2-2: Zuordnung von Fertigungstypen zu Fertigungsprinzipien nach [Günt09]

	Fertigungstyp		
	Einzelfertigung	Serienfertigung	Massenfertigung
Produktionsaufträge	Wenige	Mittel	Viel
Auftragsgröße	Gering	Mittel	Groß
Produkte	Individuell	Variable	Homogen
Materialfluss	Unregelmäßig	Auftragsweise	Kontinuierlich
Fertigungsprinzip	Baustellen, Werkstattprinzip	Baustellen, Werkstattprinzip, Inselprinzip, Fließprinzip	Inselprinzip, Fließprinzip

Für das *Reallayout* werden Restriktionen wie Transport- und Wegeflächen, Flächenbedarfe der Funktionseinheiten und gebäudebedingte Einschränkungen berücksichtigt. Das Ziel besteht darin, das Ideallayout bestmöglich in ein Reallayout zu überführen, das üblicherweise in Form eines Blocklayouts dargestellt wird (siehe Abbildung 2-5) [Pawe14]. Das Blocklayout umfasst eine schematische Abbildung der Funktionseinheiten, die vereinfacht als geometrische Elemente (Rechtecke oder Blöcke) dargestellt werden [Burg21b]. Das Ideallayout dient während der Überführung als Referenz, an der das maximale Verbesserungspotential abgeleitet werden kann [Grun21]. Da während der Grobplanung unterschiedliche Layoutvarianten entstehen und die Generierung zu großen Teilen auf der Expertise und Problemlösungsfähigkeit der Planungsbeteiligten beruht, müssen die Alternativen abschließend bewertet werden. Hierfür werden qualitative und quantitative Zielgrößen genutzt, die bereits bei der Generierung der Ideallayouts berücksichtigt werden sollten [Pawe14]. Hierzu zählen die folgenden Bewertungsgrößen [Burg21b, Wien14a, VDI 11a, VDI 99]:

- ☐ Geringe Transportkosten und Transportzeiten
- ☐ Geringe Durchlaufzeiten
- ☐ Hohe Lieferbereitschaft
- ☐ Möglichst geringe Kapitalbindung in Form von Beständen
- ☐ Hohe Qualität und Verschwendungsfreiheit in der Produktion
- ☐ Hohe Maschinenauslastungen zur optimalen Nutzung der vorhandenen Kapazitäten
- ☐ Transparenz über die Lager- und Pufferbestände
- ☐ Hohes Maß an Flexibilität
- ☐ Reduzierung der Störanfälligkeit durch entkoppelte Funktionseinheiten
- ☐ Stau- und störungsfreier Materialfluss
- ☐ Transparenter, richtungsorientierter Material- und Informationsfluss
- ☐ Hohe Wandlungsfähigkeit durch Erweiterungs- bzw. Verkleinerungsmöglichkeit
- ☐ Gute Flächennutzung

Zusätzlich müssen Rahmenbedingungen wie die erforderlichen Bodenlasten, Raumhöhen und Sicherheitsvorschriften berücksichtigt werden [Pawe14]. Im Fall einer Umplanungsaufgabe ist außerdem der Umstellungsaufwand zu beachten, der bei der Umsetzung des Layouts anfällt. Das neue Layout sollte die zuvor genannten Ziele bestmöglich erfüllen, ohne dabei zu stark von der bestehenden Lösung abzuweichen.

Feinlayout

Bei der Feinplanung wird das am besten bewertete Groblayout aufgegriffen und die Arbeitsumgebung und die Arbeitsplätze im Detail geplant. Die Zielsetzungen der Feinlayoutplanung entsprechen den zuvor aufgeführten Zielen. Zusätzlich werden jedoch auch Anforderungen im Hinblick auf Lichtverhältnisse und Emissionsgrenzwerte, wie Lärmemissionen, berücksichtigt [Pawe14]. All diese Faktoren haben einen Einfluss auf das Klima am Arbeitsplatz [Burg21b]. Während der Feinplanung sollten im Kontext der Mitarbeiterorientierung auch Aspekte der Attraktivität betrachtet werden [Wien14a]. Hierzu zählen Aspekte wie eine Verbesserung des Unfallschutzes und eine Steigerung der Leistungsmotivation durch die Senkung der physischen Beanspruchung des Personals, die durch eine ergonomiegerechte Gestaltung des Layouts realisiert werden kann [Pawe14]. Bei der Feinplanung können unterschiedliche Methoden genutzt werden, wie Spaghettidiagramme, um die tatsächlichen Bewegungen von Werkern und Materialien zu optimieren [Niem21].

Eine Herausforderung, die während der Layoutplanung berücksichtigt werden muss, sind mögliche Zielkonflikte zwischen einzelnen Zielgrößen. Wie zuvor geschildert besteht ein Ziel darin, den Materialfluss durch eine optimale Positionierung der Funktionseinheiten so effizient wie möglich zu gestalten. Dies kann durch kurze Wege realisiert werden. Kurze Wege können erreicht werden, indem Flächen klein dimensioniert und kompakt positioniert werden. Kurze Transportwege und eine kompakte Positionierung können sich jedoch negativ auf die Wandlungsfähigkeit des Fabriklayouts auswirken. Weitere Zielkonflikte bestehen zwischen Effizienz bzw. Wirtschaftlichkeit und der Flexibilität sowie Wandlungsfähigkeit, oder zwischen der Maschinenauslastung und -verfügbarkeit [Burg21b].

Zielkonflikte müssen bereits während der frühen Phase der Layoutplanung berücksichtigt werden, da hier die grundlegende Struktur des Layouts definiert wird und diese einen entscheidenden Einfluss auf die Ausprägung der Zielgrößen hat [Grun21]. Die frühe Phase der Layoutplanung entscheidet folglich maßgeblich über den Erfolg eines Fabrikplanungsprojekts [Wien14a]. Schätzungen zufolge kann eine günstig gewählte Anordnung von Funktionseinheiten den Betrieb eines Produktionssystems optimieren und die Selbstkosten eines Produkts um bis zu 50 % reduzieren [Dri07]. Dies betrifft auch den Materialfluss, der zwischen 20 - 50 % der Betriebskosten eines Produktionssystems ausmacht. Durch eine materialflussgerechte Anordnung im Rahmen der Layoutplanung lässt sich dieser Kostenanteil um bis zu 30 % verringern [Tomp10]. Der Materialfluss stellt folglich eines der wichtigsten Kriterien in der Layoutplanung dar [Burg21b].

Der Layoutplanungsprozess beruht zusammenfassend auf der Erfahrung der Planungsbeteiligten und erfordert Kreativität. Aufgrund der vielen Positionierungsmöglichkeiten der Funktionseinheiten, der Abhängigkeiten und Wechselwirkungen, der unterschiedlichen Rahmenbedingungen und der Notwendigkeit, eine hochwertige Lösung zu generieren, ist die Layoutplanung ein zeitintensiver und komplexer Prozess, der maßgeblich über die Leistungsfähigkeit und Betriebskosten des Produktionssystems entscheidet.

2.2.2 Simulationen als Bestandteil der Fabriklayoutplanung

In diesem Kapitel werden die Grundlagen der Simulation, der Modellbildung, die Potentiale von Simulationen für die Fabriklayoutplanung und Vorgehensmodelle zur Erstellung einer Simulation erläutert.

Wie zuvor geschildert, ist der Materialfluss ein zentraler Betrachtungsgegenstand der Layoutplanung, der einen maßgeblichen Einfluss auf die Leistungsfähigkeit des Produktionssystems und dessen Betriebskosten hat. Dabei wird der Materialfluss als "die Verkettung aller Vorgänge beim Gewinnen, Be- und Verarbeiten, sowie bei der Verteilung von stofflichen Gütern innerhalb festgelegter Bereiche" definiert [VDI 07]. Somit verknüpft der Materialfluss die Funktionseinheiten und umfasst neben Transportprozessen auch Lagerungs-, Umschlag- und Handhabungsprozesse [Burg21b]. Die Komplexität der Optimierung des Materialflusses und die damit einhergehende Planung von Materialflusselementen wie Förder-, Lager- und Kommissioniertechnik wird durch vielfältige Einflussfaktoren gesteigert. Hierzu zählen u. a. [VDI 14a]:

- ☐ Anstieg der Produktkomplexität und Variantenvielfalt
- ☐ Anstieg der Komplexität von Produktionsnetzwerken
- ☐ Anstieg der Anforderungen an Servicezeit und Servicegrad
- ☐ Anstieg des Kostendrucks
- ☐ Kleinere Losgrößen
- ☐ Kürzere Produktlebenszyklen

Bei der Planung des Materialflusssystems müssen daher unterschiedliche, teilweise dynamisch auftretende Wirkzusammenhänge berücksichtigt werden, wie der Einfluss von auftragsbedingten Schwankungen auf verkettete Prozesse [Hilb17]. Ohne eine Berücksichtigung der dynamischen Effekte können Störungen im Materialfluss wie Staus erst während des Betriebs erkannt und behoben werden. Folglich kann der Materialfluss im Produktionssystem nur dann optimal ausgelegt werden, wenn dynamische Aspekte bereits während der Planung berücksichtigt werden [Tomp10].

Zur Planungsunterstützung werden daher häufig Materialflusssimulationen genutzt. Die Simulation kann in diesem Zusammenhang als das „Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind“ definiert werden. Für die Durchführung von Simulationsexperimenten ist folglich ein Simulationsmodell notwendig, welches das existierende oder geplante System, alle relevanten Prozesse und Wirkzusammenhänge vereinfacht abbildet [VDI 14a].

Simulationen können anhand der gewählten Repräsentationsform (diskret oder kontinuierlich) und durch die Modellierung des Zufallsverhaltens (deterministisch oder stochastisch) klassifiziert werden [Bung13]. In Abbildung 2-7 sind die Repräsentationsformen exemplarisch dargestellt. Die verschiedenen Repräsentationsformen weisen einen unterschiedlichen Detailgrad hinsichtlich der Zustands- und Zeitrepräsentation auf. Die Modellierungsform sollte so einfach wie möglich und so komplex, wie zur Zielerfüllung nötig, gewählt werden [Schü98]. Diskrete Modellierungsformen nutzen diskrete und kombinatorische Größen wie binäre oder ganzzahlige Werte. Dagegen können in kontinuierlichen Modellierungsformen zusätzlich reelle

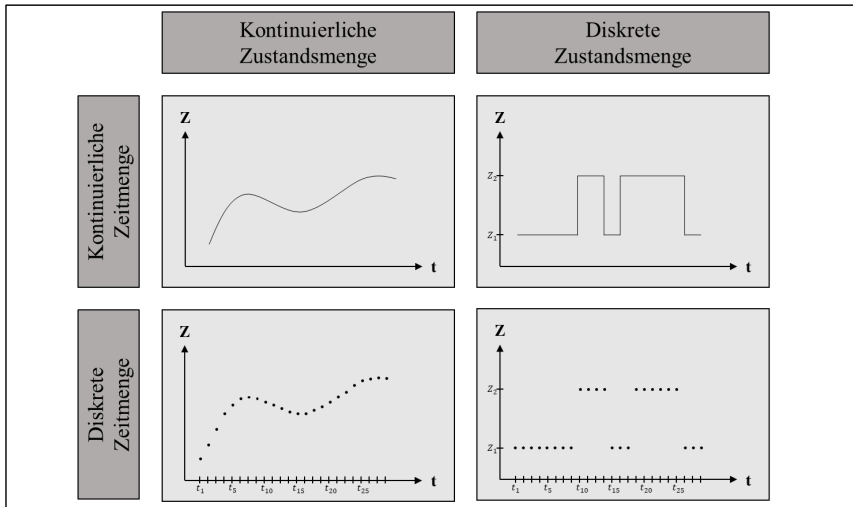


Abbildung 2-7: Klassifikation von Simulationsmodellen nach [Page91]

und kontinuierliche Repräsentationen integriert werden (z. B. Differentialgleichungen), die häufig mit höheren Rechenaufwänden verbunden sind [Bung13].

Das Zufallsverhalten kann für alle zuvor vorgestellten Repräsentationsformen zusätzlich integriert werden. Im Fall einer stochastischen Modellierungsform wird das Zufallsverhalten explizit in Form von stochastischen Prozessen in das Modell integriert. So können bspw. Unsicherheiten bzgl. Nachfrageschwankungen in die Simulation integriert werden. Bei deterministischen Modellen wird die Zufallskomponente hingegen nicht berücksichtigt [Law13, Bank10].

Für die Abbildung des Materialflussverhaltens werden in der Fabriklayoutplanung vorrangig EDS genutzt. Ereignisdiskret bedeutet, dass sich die Zustandsvariablen, z. B. der Zustand des Produktionssystems, aufgrund von Ereignissen wie dem Eintreffen von Material verändern [Regg11]. Die Zustands- und Zeitachse werden in diskretisierter Form und nicht als kontinuierliche Flussgröße dargestellt [Prit95]. Folglich werden im Modell nur die Zeitpunkte betrachtet, in denen eine Veränderung des Systemzustands auftritt. Da die Variablen des Modells nur zu diesen Zeitpunkten aktualisiert werden müssen, kann der Rechenaufwand für einen Simulationslauf im Vergleich zu einer kontinuierlichen Modellierung reduziert werden [Robi04, Matt89]. Das Produktionssystem und der zugehörige Materialfluss werden in der Modellierung als Gesamtsystem, bestehend aus mehreren verknüpften Subsystemen, z. B. Funktionseinheiten und Transportmittel, angesehen [Mart11]. Jedes Subsystem besitzt Eigenschaften, die vor der Simulation konfiguriert werden. So werden für Fördermittel u. a. die Geschwindigkeiten, Kapazitäten und Handhabungszeiten festgelegt. Anschließend werden die Subsysteme logisch miteinander zu einem Gesamtsystem verknüpft, indem die Materialflussteuerung und die Produktionsplanung und -steuerung unter Berücksichtigung des Produktionsprogramms, der Ablauforganisation, der Arbeitszeitorganisation und der Ressourcenzuordnung definiert werden. Zusätzlich werden durch das Fabriklayout die Position der Funktionseinheiten, Verkehrswege und Restriktionen vorgegeben, sodass Zielgrößen auf

der Gesamtsystemebene, wie die Durchlaufzeit, oder Zielgrößen auf der Subsystemebene, wie die Maschinenauslastung, bestimmt werden können [Bang08].

Eine Alternative zur simulationsbasierten Planung von Materialflusssystemen stellen analytisch modellierte Optimierungsprobleme dar [Regg11]. Analytische Modelle erfordern eine exakte mathematische Beschreibung des zu lösenden Problems in Form von Zielfunktionen und Nebenbedingungen. Für Problemformulierungen bis zu einer gewissen Komplexität können diese exakt oder approximativ mittels (Meta-) Heuristiken gelöst werden. Bei steigender Komplexität nimmt jedoch die Lösungsqualität ab. Die Komplexität kann in die zwei Bereiche der dynamischen und kombinatorischen Komplexität unterteilt werden [Robi14]. Die *dynamische Komplexität* umfasst die Wechselwirkung zwischen Entitäten des Systems [Ster00]. Produktions- und Materialflusssysteme umfassen eine Vielzahl dynamischer Prozesse, die teilweise durch stochastische Wirkzusammenhänge beeinflusst werden [Hart91]. Daher können häufig nicht alle Größen analytisch durch Gleichungen beschrieben werden [Law13]. Zudem erlauben Graphenmodelle und Ansätze des Operations Research lediglich die Lösung von Zuweisungs- und Transportproblemen und sind zur Bestimmung des Gesamtsystemverhaltens ungeeignet [Hart91]. Simulationen ermöglichen im Gegensatz zu analytischen Modellierungsverfahren die Berücksichtigung von dynamischen Effekten [Hilb17], ohne dass komplexe Wirkzusammenhänge unter Nutzung von Annahmen und Vereinfachungen in analytischer Form beschrieben werden müssen [Arno08]. Die *kombinatorische Komplexität* bezieht sich auf die Problemgröße, also auf die Anzahl der Entitäten im System [Robi14]. Das Layoutplanungsproblem gehört zu den NP-schweren Problemen [Amar06]. Das bedeutet, dass die kombinatorische Komplexität exponentiell mit der Problemgröße anwächst [Gare79]. Dies erschwert eine analytische Lösung des Problems, falls die Problemgröße zu groß wird. Zusätzlich können bei einer höheren Problemkomplexität geschlossene, analytische Formulierungen nicht oder nur durch Annahmen und Vereinfachungen gefunden werden. Dieses Defizit kann durch den Einsatz von Simulationsmodellen adressiert werden, da sie eine Abbildung der Produktionssystemeigenschaften auch bei zunehmender Problemgröße ermöglichen [Gute17].

Der Einsatz von Simulationen ermöglicht in Bezug auf die Layoutplanung insbesondere, dass Experimente mit dem Gesamtsystem oder mit Subsystemen durchgeführt werden können, ohne dass aktiv in das bestehende System eingegriffen wird. Im Fall einer Neuplanung können die Prozesse analysiert und ausgelegt werden, ohne dass ein reales Produktionssystem existiert [Bank10]. Fehler können hierdurch frühzeitig erkannt und behoben werden [Krüg19].

Somit können generierte Layoutvarianten hinsichtlich der Leistungsfähigkeit und der Dauer von Einschwingvorgängen quantitativ bewertet werden [Hilb17]. Darüber hinaus kann der Einfluss externer Faktoren und Parameteränderungen in Form von Sensitivitätsanalysen gezielt analysiert werden, sodass ein möglichst leistungsfähiges Produktionssystem realisiert werden kann [Pegd85]. Simulationen zählen zu den wichtigsten Methoden, um dynamische Aspekte und komplexe Zusammenhänge abzubilden [Brac18b] und eignen sich daher zur Planungsunterstützung in der Fabrikplanung [Homp18, Domb15]. Es sollte jedoch hervorgehoben werden, dass das Erstellen einer Simulation meist mit großen Modellierungsaufwänden verbunden ist und die Analyse und Ergebnisinterpretation zeitaufwändig sind [Bank10].

Das Kosten-Nutzen-Verhältnis einer Simulation kann lediglich abgeschätzt werden und variiert quellenabhängig. Während [VDI 14a] von zwei bis vier Prozent Einsparungen in Bezug auf die Investitionssumme des Gesamtprojekts spricht, weist [Etsp96] ein Verhältnis von 1:6 zwischen Aufwand und Erträgen aus. Trotz dieser Unterschiede kann der Effekt einer Simulation auf die Fabrikplanung als positiv angesehen werden, da dieser zu einem Sicherheitsgewinn, kostengünstigeren Lösungen, effizienteren Prozessen und einem besseren Systemverständnis führen kann [VDI 14a].

Für die Umsetzung einer Simulationsstudie, die eine Durchführung mehrerer Simulationsläufe mit unterschiedlichen Parameterkonfigurationen umfasst, existieren diverse Vorgehensmodelle [Gute17]. In [VDI 14a] wird das Vorgehen nach [ASIM97] aufgezeigt, das in die sequentiell ablaufenden Phasen Aufgabendefinition, Systemanalyse, Modellformalisierung, Implementierung, Experimente und Analyse aufgeteilt wird. Parallel zu diesen Phasen werden für Analyse- und Abstraktionszwecke Daten generiert und aufbereitet. Während der Analyse wird das Gesamtsystem strukturiert und auf unterschiedlichen Untersuchungsebenen unter Berücksichtigung der Projektziele analysiert. Hierbei kann ein Bottom-Up- oder Top-Down-Ansatz genutzt werden. Der Top-Down-Ansatz ermöglicht ein schnelles Verständnis der Zusammenhänge, begrenzt den Modellierungsaufwand auf den problemrelevanten Anteil, erfordert aber ein hohes Abstraktionsvermögen. Bei einem Bottom-Up-Ansatz sind die Anforderungen an das Abstraktionsvermögen zu Beginn geringer und es können vorgefertigte Strukturen genutzt werden. Jedoch steigt der Modellierungsaufwand bei komplexen Gesamtsystemen [VDI 14a]. Alternativen sind die Simulationspipeline nach [Bung13], das Vorgehen nach [Bank10] und der Simulationskreislauf nach [Kühn06]. Im zuletzt genannten Ansatz wird das reale System (z. B. Materialflusssystem oder Produktionssystem) im Anschluss an die Analyse und Abstraktion in Analogie zur Vorgehensweise in [VDI 14a] modelliert und implementiert. Das Ergebnis stellt ein Simulationsmodell dar, das den Ausgangspunkt für Simulationsexperimente bildet. Die Ergebnisse der Experimente werden ausgewertet und interpretiert, sodass Verbesserungen im realen System realisiert werden können [Kühn06].

Bei der Verwendung von Simulationen muss außerdem die Validierung von Modellen und Ergebnissen sichergestellt werden [Krüg19]. Dabei wird in einem iterativen Überprüfungs- und Korrekturprozess sichergestellt, dass das Simulationsmodell das Verhalten des realen Systems hinreichend genau abbildet. Zur Validierung können die Simulationsergebnisse mit dem Verhalten des realen Systems verglichen werden. Außerdem wird im Rahmen der Verifizierung sichergestellt, dass keine Abweichungen durch Programmierfehler entstanden sind [VDI 14a].

In Abhängigkeit von der Zielsetzung des Fabrikplanungsprojekts können die Planung, die Optimierung und die Durchführung der Simulationsstudie in vier unterschiedlichen zeitlichen und organisatorischen Verknüpfungen zueinanderstehen (siehe Abbildung 2-8). Für die ersten beiden Fälle sind die Optimierung und Simulation sequentiell verkettet, während in den anderen Fällen eine hierarchische Verknüpfung vorliegt [VDI 14b]. Welche Verknüpfung für die Fabriklayoutplanung am zweckmäßigsten ist, hängt vom Planungsproblem ab. Im Hauptteil dieser Arbeit wird eine Verknüpfung gewählt, in der die Simulation hierarchisch in die Optimierung integriert ist. Die Simulation wird somit als ein Werkzeug zur Ermittlung von Zielfunktionswerten für die Optimierung des Fabriklayouts genutzt. Herausforderungen stellen laut [VDI 14b] die Gestaltung eines Simulationsmodells dar, das mit einem geringen

Rechenaufwand intelligent gestaltet und gesteuert werden kann. Folglich muss die Simulationsumgebung modular gestaltet sein, sodass ein Fabriklayout ausschließlich über definierte Schnittstellen initialisiert und simuliert werden kann [VDI 14b].

Der Materialfluss ist zusammenfassend ein wichtiger Gesichtspunkt für die Layoutplanung, der einen entscheidenden Einfluss auf die Leistungsfähigkeit und Betriebskosten des Produktionssystems hat. Wesentliche Merkmale des Materialflusses und der zukünftigen Fabrik werden bereits in der frühen Phase der Layoutplanung definiert, in der die Struktur durch das Positionieren der Funktionseinheiten bestimmt wird. Für den Materialfluss sind insbesondere dynamische Aspekte von Relevanz. Auch diese müssen bereits in der frühen Phase der Layoutplanung berücksichtigt werden, um Störungen im Materialfluss frühzeitig vorzubeugen. Zur Berücksichtigung der dynamischen Aspekte stellen EDS ein leistungsfähiges Unterstützungswerkzeug dar, das es ermöglicht, Layoutvarianten und verschiedenen komplexe Materialflusssysteme quantitativ zu bewerten. Hierdurch ist es möglich, den Materialfluss effizient zu gestalten und dabei Durchlaufzeiten und Verschwendungen im Materialfluss zu senken.

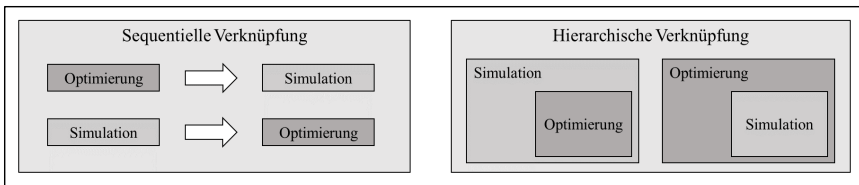


Abbildung 2-8: Verknüpfung von Optimierung und Simulation nach [VDI 14b]

2.2.3 Manuelle Fabriklayoutplanungsansätze

Zur Generierung von Fabriklayoutvarianten in der frühen Phase der Layoutplanung existieren verschiedene Ansätze. Diese können in manuelle, automatisierte exakte und automatisierte approximative Fabrikplanungsansätze unterteilt werden. In diesem Kapitel wird ein Überblick über manuelle Fabrikplanungsansätze gegeben. Diese basieren auf den Erfahrungswerten und der Problemlösungskreativität der Planenden, die das Layout durch methodisches Ausprobieren generieren [Schu10]. Der Ausgangspunkt der manuellen Planung ist häufig ein Ideallayout oder Funktionsschema. Diese können mittels graphischer oder heuristischer Verfahren generiert werden, die manuell oder rechnergestützt ausgeführt werden können [Burg21b]. Zu den graphischen Verfahren gehören neben dem in Kapitel 2.2.1 erklärten Dreieckprobiervorgehen nach SCHMIGALLA [Schm70] auch das Kreisverfahren nach SCHWERTFEGGER oder das Sankey-Diagramm (bzw. mengenmaßstäbliches Flussdiagramm). Heuristische Verfahren sind u. a. das Aufbauverfahren Computerized Relative Layout Planning (CORELAP) sowie das Vertauschverfahren Computerized Relative Allocation of Facilities (CRAFT) [Burg21b]. Weitere Verfahren werden in [Burg21b, Pawe14] aufgezählt. In Abbildung 2-9 sind Ergebnisse der vier zuvor genannten Verfahren beispielhaft dargestellt.

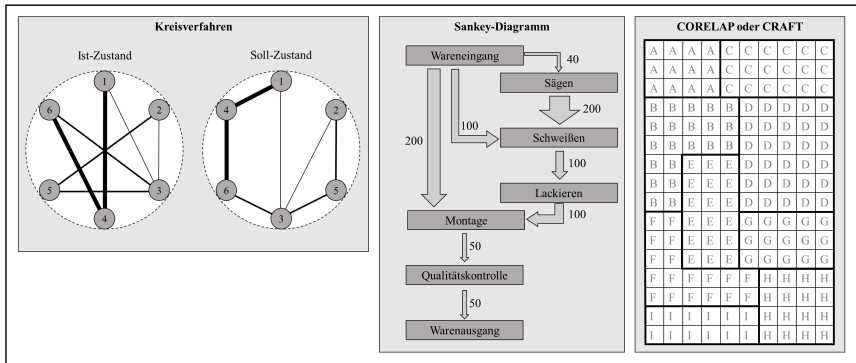


Abbildung 2-9: Planungsverfahren in der frühen Phase der Layoutplanung

Beim Kreisverfahren nach SCHWERTFEGGER werden die Funktionseinheiten kreisförmig angeordnet und schrittweise vertauscht. Dabei sollen Funktionseinheiten, die starke Materialflussbeziehungen aufweisen, benachbart platziert werden, sodass möglichst keine oder nur schwach ausgeprägte Materialflussbeziehungen durch das Innere des Kreises laufen [Grun21]. Die Stärke der Materialflussbeziehung wird durch angepasste Linienstärken dargestellt [Heis11]. Das Sankey-Diagramm stellt in Analogie die Materialflussbeziehungen zwischen den Funktionseinheiten mengenmaßstäblich anhand der Pfeilbreite dar [Lieb16, Arno08]. Der Ausgangspunkt für die Erstellung kann eine Transportmatrix sein. Das Sankey-Diagramm ermöglicht eine Analyse der Materialflussbeziehungen, ohne die tatsächliche räumliche Anordnung zu beachten. Bei einer hohen Materialflusskomplexität kann das Sankey-Diagramm unübersichtlich werden [Grun21].

CORELAP ist ein heuristisches Verfahren, das zum Generieren einer Ausgangslösung genutzt werden kann, ohne dass ein initiales Layout vorliegen muss. Hierfür werden die Beziehungen der Funktionseinheiten initial durch den TCR-Wert (Total Closeness Rating) bewertet. Anschließend werden die Funktionseinheiten sequentiell unter Berücksichtigung des TCR-Werts und der gewünschten Art der Nachbarschaftsbeziehung (gewünscht bis unerwünscht) mittels einer Platzierungsbewertung, die alle bisher positionierten Funktionseinheiten berücksichtigt, platziert [Lee67]. Das CRAFT Verfahren benötigt im Vergleich zum CORELAP-Ansatz eine Ausgangslösung, die anschließend weiter optimiert wird. CRAFT ist ein Vertauschungsverfahren, das durch iteratives Vertauschen von zwei oder drei Funktionseinheiten versucht das Layout zu verbessern. Hierbei werden benachbarte Funktionseinheiten oder Funktionseinheiten gleicher Größe vertauscht. Die möglichen Vertauschungen werden anhand des erwarteten Verbesserungspotentials bewertet und umgesetzt, bis die zuvor festgelegte Anzahl an maximalen Vertauschungen erreicht wird und keine weitere Verbesserung des Layouts erreicht werden kann [Buff64]. Es existieren diverse Modifikationen des CRAFT-Ansatzes wie CRAFT-M [Hick76], CRAFT-3D [Cina75] oder die Biased Sampling Technique [Nuge68].

Anschließend wird das Ideallayout unter Berücksichtigung der Transport- und Wegeflächen sowie gebäudebedingter Restriktionen manuell in mehrere Reallayoutvarianten überführt. Hierbei werden die Optimierungszielgrößen, die teilweise konfliktäre Beziehungen aufweisen können, berücksichtigt (vgl. Kapitel 2.2.1) [Pawe14]. Als Unterstützungswerkzeug können in

dieser Phase der Fabriklayoutplanung auch Ansätze der virtuellen und augmentierten Realität (VR, AR) genutzt werden. Diese ermöglichen einen interaktiven und informationsangereicherten Planungsprozess [Nee12]. So können virtuelle Planungsobjekte mittels AR in der realen Umgebung positioniert und verschoben werden. Dabei erhält der Planer von der AR-Anwendung Informationen zu Positionen im Layout oder zu relevanten Zielgrößen [Herr18]. Die Erstellung einer solchen Planungsumgebung ist häufig zeitintensiv und erfordert Expertenwissen, weshalb AR- und VR-Anwendungen lediglich selten genutzt werden [Nee12]. Abschließend werden die generierten Layoutvarianten bewertet und miteinander verglichen. Wie in Kapitel 2.2.2 werden hierzu häufig Materialflusssimulationen genutzt, da diese dynamische Effekte abbilden können und so eine realitätsnahe Beurteilung der Layoutvarianten ermöglichen [Tomp10]. Als Bewertungsmethodik werden bspw. Nutzwertanalysen oder der analytische Hierarchieprozess genutzt. Diese ermöglichen es, sowohl qualitative als auch quantitative Aspekte miteinander zu vergleichen. Die Quantifizierung qualitativer Aspekte ist jedoch mit subjektiven Entscheidungsprozessen des Planers verbunden [Schn91, Saat89].

Zusammenfassend bringen manuelle Planungsverfahren in der Regel zulässige Lösungen hervor. Da die manuelle Layoutgenerierung auf kreativen Prozessen basiert, werden aus Kapazitätsgründen meist nur wenige alternative Layoutvarianten erstellt [Schu10]. Ein weiteres Problem, das aus manuellen Verfahren resultiert, ist die Tendenz der menschlichen Planer, eine als geeignet erscheinende räumliche Anordnung der Funktionseinheiten zunächst festzulegen (z. B. Maschinen aufeinanderfolgender Bearbeitungsschritte) und weitere Planungsgegenstände, wie den Materialfluss, hinsichtlich der Kompatibilität an das festgelegte Layout anzupassen [Tomp10]. Dies bewirkt, dass diese Planungsgegenstände häufig nicht optimal ausgestaltet werden. Es ist anzunehmen, dass sich diese Tendenz zur initialen Festlegung der räumlichen Anordnung auch negativ auf andere Planungsaspekte auswirkt. Die nachgelagerte Bewertung der generierten Varianten findet häufig unter der Verwendung qualitativer Methoden statt und hängt von der Kompetenz und Erfahrung der beteiligten Planer ab [Grun21]. Neben der damit einhergehenden teilweisen Subjektivität der Entscheidungsfindung bewirkt dies auch eine Einschränkung des bewältigbaren Lösungsraumes. So zeigen kognitionspsychologische Untersuchungen, dass die Fähigkeit eines Menschen zum Vergleich mehrerer Alternativen ab einer Alternativenzahl von sieben stark abnimmt [Mill94]. Zur Überwindung der Nachteile der manuellen Methoden wird die frühe Phase der Layoutplanung häufig als kombinatorisches Optimierungsproblem modelliert und mittels verschiedener exakter und approximierender Berechnungsverfahren rechnergestützt gelöst [Hoss18].

2.2.4 Automatisierte exakte und approximative Fabriklayoutplanungsansätze

Die hierdurch beschriebene Fragestellung zur Findung eines optimalen Layouts ist als „Facility Layout Problem“ (FLP) bekannt [Kusi87]. Zur automatisierten rechnerbasierten Lösung des FLP ist es notwendig, ein kombinatorisches Optimierungsproblem zu formulieren. Hierfür wird das Layoutplanungsproblem mittels Zielfunktionen und Nebenbedingungen mathematisch modelliert [Mell96]. Das FLP kann als einzelkriterielles oder multikriterielles Optimierungsproblem modelliert werden [Sing06]. Mögliche Zielgrößen können, in Analogie zu Kapitel 2.2.1, Messgrößen des Materialflusses (z. B. Durchlaufzeit oder Transportkosten) [Rama09], Flächenbedarf und -nutzung, Umsetzungsaufwand [Pawe14], der Energiebedarf des

Produktionssysteme [Yang13], die Flexibilität der technischen Anlagen [Khan13] oder die Wandlungsfähigkeit sein [Wien14a].

Daneben existieren weitere Planungscharakteristika, die in das FLP mit einbezogen werden können und dieses charakterisieren. Eine Übersicht ist in Abbildung 2-10 gegeben. Die Ansätze können danach unterscheiden werden, ob der Entscheidungsraum diskret oder kontinuierlich dargestellt ist. Im ersten Fall wird das Layout oftmals rasterartig in rechteckige Positionen (Elementarzellen) unterteilt. Funktionseinheiten werden dann einer oder mehreren Positionen zugeordnet [Frug06]. Im kontinuierlichen Fall kann eine Funktionseinheit auch allen Zwischenpositionen zugeordnet werden [DAS93]. Im diskreten Fall kann das Layoutplanungsproblem als quadratisches Zuordnungsproblem formuliert werden, während für kontinuierliche Repräsentationen ein gemischt ganzzahliges Optimierungsproblem genutzt werden kann [Hoss18]. Ein weiteres Charakteristikum ist die Layoutkonfiguration. Die Layoutkonfiguration hat einen Einfluss auf die Größe des Lösungsraums. Im Fall des Single Row Layouts werden Funktionseinheiten ausschließlich direkt benachbart zur Materialflusslinie platziert [Anjo17]. Einen Kontrast hierzu stellt das Open Field Layout dar. In diesem können Funktionseinheiten auch ohne direkte Nachbarschaftsbeziehung zur Materialflusslinie positioniert werden. Demnach umfasst ein Open Field Layout üblicherweise mehr Positionen, die für die Platzierung von Funktionseinheiten genutzt werden können, als ein Single Row Layout. Ferner wird im Fall eines Open Field Layouts angenommen, dass ungenutzte Flächen im Anschluss an die Optimierung für den Materialtransport genutzt werden können. Daher ist der Lösungsraum im Open Field Layout deutlich größer als im Fall eines Single oder Multi Row Layouts [Yang05]. Die Geometrie der Funktionseinheiten ist ein weiteres Charakteristikum und kann in 2 oder 3 Dimensionen berücksichtigt [Besb21] und für alle Funktionseinheiten als einheitlich oder uneinheitlich angenommen werden [Kim00]. Die Komplexität des FLPs kann weiter gesteigert werden, wenn neben den Materialflusseigenschaften zusätzliche Nebenbedingungen wie Bodenlasten, Anforderungen an die Medienversorgung und -entsorgung oder die Emissionen der Funktionseinheiten mit in

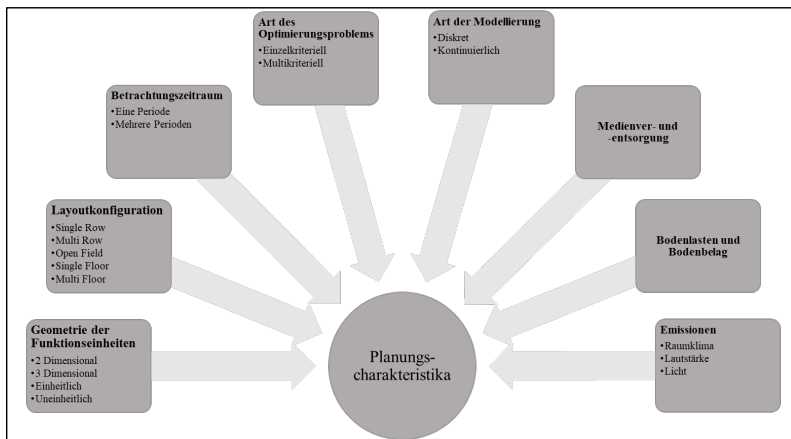


Abbildung 2-10: Planungscharakteristika des FLP nach [Hoss18, VDI 11b, Drir07]

die Optimierung einbezogen werden [Süße21]. Eine grundsätzliche Herausforderung bei der mathematischen Modellierung des FLPs stellt der heterogene Charakter der Planungscharakteristika dar. Demnach müssen in Abhängigkeit vom Planungsproblem verschiedene Annahmen und Approximationen definiert werden, um z. B. Materialflussbeziehungen durch analytische Gleichungen darzustellen [Auri20].

Auf der mathematischen Formulierung aufbauende **exakte Verfahren** nutzen bspw. Branch-and-Bound- [Soli08] oder Dynamic-Programming-Algorithmen [Dunk05], um die mathematisch bestmögliche Lösung eines Layouts zu identifizieren. Da das FLP zu den NP-schweren Problemen gehört, steigt die Anzahl der Lösungsmöglichkeiten exponentiell mit der Problemgröße an [Amar06]. Die Berechnung der exakten Lösung für FLPs mit einer industrierelevanten Problemgröße ist daher mit einem hohen Berechnungsaufwand verbunden, weshalb sich exakte Verfahren ausschließlich für kleine Problemgrößen eignen [Drir07].

Zur Lösung praxisnaher Problemdimensionen können **approximative Verfahren**, sog. Metaheuristiken, genutzt werden. Hierfür kann eine ähnliche mathematische Problemmodellierung wie bei den exakten Lösungsverfahren genutzt werden. Anstelle der Berechnung des Optimums wird bei einer Metaheuristik jedoch eine dem Optimum nahe Lösung generiert [Tuzk13]. Es existiert eine Vielzahl an möglichen Metaheuristiken, die auch in Kombination miteinander eingesetzt werden können. Dazu gehören u. a. das Simulated Annealing, Adaptive Large Neighborhood Search [Sout21], Particle Swarm Optimization [Sama10], oder die Ant Colony Optimization [McKe06a]. Die am stärksten verbreiteten Ansätze für das FLP sind der Genetische Algorithmus (GA) und die Tabu Suche (TS) [Hoss18, Al-S99]. Der GA- und TS-Ansatz wird daher in Kapitel 8 als Vergleichsmaßstab genutzt. Der Ablauf des GA und TS ist in Abbildung 2-11 dargestellt. Der GA beginnt mit einer Menge von Ausgangslösungen (sog. Generation). Jede Layoutvariante der 1. Generation wird durch die Fitnessfunktion bewertet. Die besten Lösungen stellen die sog. Eltern dar und bilden den Ausgangspunkt für die Generierung der nächsten Generation [Forr96]. Hierfür wird das sog. Crossover durchgeführt. Bei diesem wird ein Teil der Funktionseinheiten entsprechend der ersten Elternlösung positioniert und der zweite Teil entsprechend der zweiten Elternlösung. Einzelne Funktionseinheiten können durch die anschließende Mutation zufällig umplatziert werden. Die neue Generation umfasst folglich die n_E Eltern und $n - n_E$ neu generierte Lösungen. Dieses Vorgehen wird wiederholt, bis z. B. die maximale Anzahl an Episoden erreicht wurde. Die Lösungen des GA werden nicht schlechter, da immer mindestens die besten beiden Lösungen in die nächste Generation übergehen [Kram17].

Eine Alternative zum GA stellt die TS dar. Der Ausgangspunkt für die TS ist ein einzelnes Fabriklayout. Ausgehend von dieser Lösung wird eine Nachbarylösung generiert, indem z. B. einzelne Funktionseinheiten umplatziert oder getauscht werden. Dieser Vorgang wird mehrfach wiederholt, sodass eine Menge an Nachbarschaftslösungen generiert wird. Für den Fall, dass die hierdurch generierten Lösungen besser als die bestehende Lösung sind, wird die beste Lösung angenommen. Im Fall einer ausschließlichen Verschlechterung wird mittels Gleichung 2-1 die Annahmewahrscheinlichkeit $P(x)$ für diese Lösung berechnet. Je größer die Differenz zwischen der Bewertung der alten Lösung $f(x_{old})$ und der neuen Lösung $f(x_{new})$ ist, desto kleiner ist die Annahmewahrscheinlichkeit. Der Temperaturkoeffizient T_k sinkt in Abhängigkeit von der Modellierungsform linear oder exponentiell, sodass die Annahmewahrscheinlichkeit über die Anzahl der Episoden abnimmt. Im Vergleich zum GA ist

folglich eine Verschlechterung der aktuellen Lösung möglich und soll das Überwinden eines lokalen Optimums erleichtern. Darüber hinaus wird eine Tabu Liste angelegt. Diese beinhaltet alle bislang angenommenen Lösungen. Angenommene Lösungen dürfen in einem definierten Zeitintervall nicht erneut angenommen werden. Demnach stellt die Tabu Liste sicher, dass dauerhaft neue Lösungen generiert werden [Lian08, Chia96, Glov90].

$$P(x) = \exp\left(-\frac{f(x_{new}) - f(x_{old})}{T_k}\right) \quad (2-1)$$

Zur weiteren Steigerung der Leistungsfähigkeit können die genannten Metaheuristiken miteinander kombiniert werden, sodass eine Metaheuristik eine globale Suche durchführt, während die zweite Metaheuristik gefundene Lösungen lokal optimiert. Durch diese Kombination stellt die integrierte Form aus GA und TS eine der leistungsfähigsten Metaheuristiken zur Lösung des FLPs dar [Pour14].

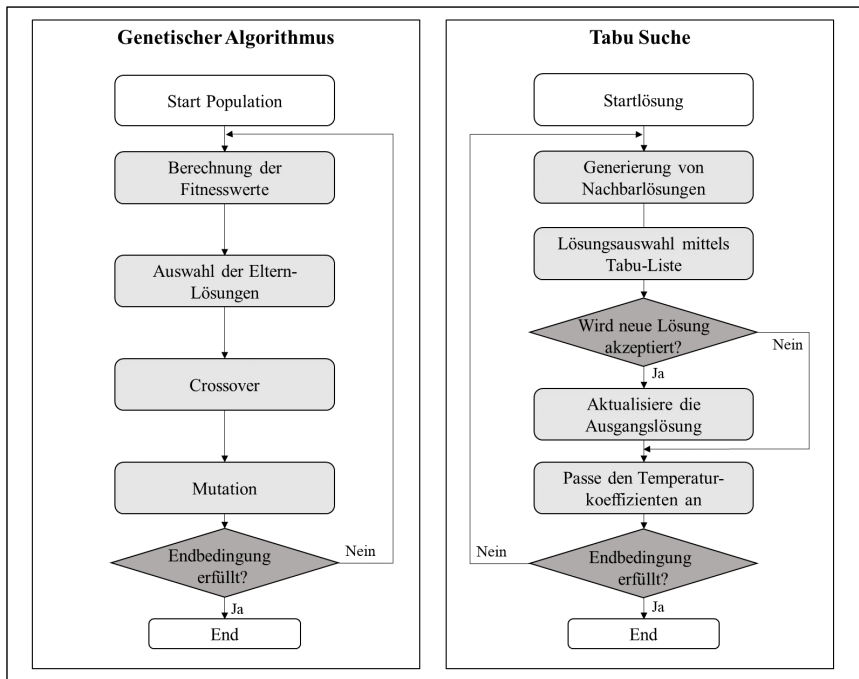


Abbildung 2-11: Ablauf des Genetischen Algorithmus und der Tabu Suche nach [Kram17, Forr96, Bert93, Rute89, Laar87]

Metaheuristiken führen unter Betrachtung der quantitativen Randbedingungen und Zielgrößen je nach Algorithmus zu guten Lösungen. Grundsätzlich lassen sich sowohl einzelkriterielle als

auch multikriterielle FLPs lösen, indem einzelne Zielgrößen gewichtet werden, so dass eine Priorisierung der Zielgrößen möglich ist [Turg18]. Die Modellierung des Planungsproblems stellt jedoch eine zentrale Herausforderung von metaheuristischen Ansätzen dar, da die Qualität des Modells einen direkten Einfluss auf die Realitätsnähe der Lösung hat. Daher muss bei der Nutzung von Metaheuristiken sichergestellt werden, dass die gewählten Annahmen in der Modellierung das Problem nicht zu stark vereinfachen.

2.3 Bewertung von automatisierten Layoutplanungsansätzen

Aufbauend auf den Inhalten der vorangegangenen Kapitel werden in Kapitel 2.3.1 zunächst Anforderungen an automatisierte Planungsansätze in der frühen Phase der Layoutplanung definiert. Diese werden im Anschluss in Kapitel 2.3.2 genutzt, um die bestehenden Planungsansätze zu bewerten.

2.3.1 Anforderungsdefinition an automatisierte Layoutplanungsansätze

Anforderung 1: Anwendbarkeit für die Neu- und Umplanung

Die beiden verbreitetsten Planungsgrundfälle sind die Neu- und die Umplanung. Daher sollte ein Planungsansatz nicht lediglich einen der beiden Fälle adressieren, sondern für beide Fälle integriert genutzt werden können. Diese Anforderung kann daher erfüllt oder nicht erfüllt werden.

Anforderung 2: Multikriterielle Optimierung

Während der Fabriklayoutplanung können verschiedene Zielgrößen als Optimierungskriterium betrachtet werden. Dabei ist die Berücksichtigung einer Zielgröße häufig nicht ausreichend. Vielmehr erfordert es die Fabriklayoutplanung, dass verschiedene Optimierungsgrößen zur gleichen Zeit berücksichtigt werden. Die Priorisierungsmöglichkeit von quantifizierbaren Zielgrößen ermöglicht es zudem, verschiedene Fabriklayoutvarianten zu erzeugen, die objektiv miteinander verglichen werden können. Daher sollten Planungsansätze gleichzeitig verschiedene Zielgrößen durch eine Priorisierung berücksichtigen können.

Ein Teil der möglichen Zielgrößen kann durch analytische Formeln beschrieben werden. Hierzu zählen z. B. die Transportintensität, Transportkosten oder der Flächenbedarf. Diese Zielgrößen sollten von ganzheitlichen Planungsansätzen berücksichtigt werden. Neben den analytischen Zielgrößen sind dynamische Aspekte im Materialfluss von großer Bedeutung. Eine Einbeziehung ist durch die Nutzung von EDS möglich, die eine Bewertung von dynamischen Zielgrößen wie Durchlaufzeiten, Auslastungen und Einfluss von Staus im Materialfluss ermöglichen.

Zur Beurteilung der bestehenden Ansätze wird zuerst bewertet, ob eine einzelkriterielle oder multikriterielle Optimierung durchgeführt wird. Falls lediglich eine einzelkriterielle Optimierung vorgestellt wurde, gilt diese Anforderung als nicht erfüllt. Die Anforderung gilt als teilweise erfüllt, wenn ausschließlich analytische oder simulationsbasierte Zielgrößen genutzt werden. Im Fall einer multikriteriellen Optimierung mit beiden Zielgrößentypen gilt diese Anforderung als vollständig erfüllt.

Anforderung 3: Realitätsnähe der Planungsergebnisse

Bei der Planung eines Fabriklayouts müssen diverse Nebenbedingungen berücksichtigt werden und in einen Planungsansatz einfließen, um ein möglichst realitätsnahes Planungsergebnis zu

generieren. Dazu gehört u. a., dass die Größe der Funktionseinheiten bei der Planung berücksichtigt wird, anstatt diese auf eine einheitliche Größe zusammenzufassen. Ein weiteres Beispiel stellt die Berücksichtigung von zulässigen Bodenlasten dar, die eine Platzierung einer Funktionseinheit vorgeben oder einschränken kann. Außerdem sollte der Freiheitsgrad während des Platzierungsvorgangs von Funktionseinheiten so groß wie möglich sein, sodass die Realitätsnähe der Planungsergebnisse nicht durch die Modellierung der Platzierungsmöglichkeiten beschränkt wird. Die bestehenden Ansätze erfüllen die 3. Anforderung, wenn folgende Kriterien berücksichtigt werden:

- ☐ Funktionseinheiten werden mit ihrer Breite und Länge betrachtet. Es erfolgt keine vereinfachte Darstellung mittels einer Einheitsgröße.
- ☐ Die Höhe der Funktionseinheiten und der Planungsumgebung werden berücksichtigt.
- ☐ Funktionseinheiten besitzen definierte Aufnahme- und Abgabepunkte für den Materialtransport.
- ☐ Nebenbedingungen wie die zulässigen Bodenlasten und Medienverfügbarkeiten werden berücksichtigt.
- ☐ Das Planungsproblem wird als Open Field Layout modelliert. Eine Platzierung der Funktionseinheiten ist somit im gesamten Bereich und nicht lediglich an definierten Positionen direkt an den Materialflusswegen möglich.

Anforderung 4: Skalierbarkeit hinsichtlich der Problemgröße

Ein Planungsvorgehen sollte die Generierung von hochqualitativen Layoutvarianten in einer kurzen Zeit ermöglichen, um personelle, zeitliche und Berechnungsressourcen zu reduzieren. Da das Layoutplanungsproblem ein NP-schweres Problem darstellt, ist es wichtig, dass auch für Problemstellungen mit steigender Größe geeignete Lösungen gefunden werden können. Darüber hinaus wird durch eine kurze Durchführungszeit eine kontinuierliche Umplanung als Folge einer Layoutevaluation möglich. Zur Bewertung dieser Anforderung wird geprüft, ob der Ansatz lediglich theoretisch beschrieben oder auch für die Lösung von Layoutplanungsproblemen mit variierender Größe eingesetzt wird. Falls Layoutplanungsprobleme gelöst werden, wird bewertet, wie stark die Lösungszeit bei steigender Problemgröße ansteigt. Da Layoutplanungsprobleme einen heterogenen Charakter besitzen, können die Absolutwerte nicht verglichen werden, weshalb dieses Kriterium qualitativ bewertet wird. Ferner wird bewertet, ob ein Planungsansatz sicher zur optimalen Lösung führt.

Anforderung 5: Nachvollziehbarkeit der Planungsergebnisse

Da die Ergebnisse der frühen Phase der Fabriklayoutplanung maßgeblich über die Leistungsfähigkeit des zukünftigen Produktionssystems entscheiden, ist es wichtig, dass der Entstehungsprozess zuverlässig und vertrauenswürdig ist. Diese Aspekte steigern die Akzeptanz einer Lösung und ermöglichen es, Rückschlüsse auf die zentralen Einflussfaktoren während der Lösungsfindung zu ziehen. Daher wird bewertet, ob bei einem Planungsansatz bei wiederholter Durchführung das gleiche Ergebnis erzielt wird, ob Entscheidungen stochastisch beeinflusst werden und ob ein Modell des Lösungsprozesses existiert, das eine weiterführende Analyse der zentralen Entscheidungen ermöglicht.

2.3.2 Zusammenfassende Bewertung

Die zuvor aufgeführten Anforderungen stellen den Ausgangspunkt für die nachfolgende Bewertung der bestehenden automatisierten Planungsansätze für die frühe Phase der Layoutplanung dar. Aktuelle Metaanalysen zeigen die hohe Anzahl an Publikationen im Bereich der automatisierten Fabriklayoutplanung auf [Burg21a, Pére21, Hoss18, Drir07]. Daher werden ähnliche Planungsansätze aggregiert und gemeinsam anhand ausgewählter repräsentativer Ansätze bewertet. Die zusammenfassende Bewertung ist in Abbildung 2-12 visualisiert.

Bewertung der exakten Planungsansätze

Zu den exakten Planungsansätzen gehören u. a. Branch-and-Bound- [Soli08], Dynamic-Programming- [Dunk05] oder Cutting-Plane-Algorithmen [Hoss18]. Bestehende Ansätze werden dabei entweder für die Neu- oder Umplanung verwendet. Integrierte Ansätze, die auf beide Planungsfälle angewendet werden, wurden nicht identifiziert. Daher ist die 1. Anforderung bei allen nachfolgenden Ansätzen nicht erfüllt. Zudem stellt die Skalierbarkeit der exakten Ansätze ein grundsätzliches Problem dar, da die kombinatorische Komplexität exponentiell mit der Anzahl der Funktionseinheiten steigt. Demnach ist die Lösung von Problemen mit industrieller Größe nicht praxistauglich, weshalb die 4. Anforderung bei allen exakten Planungsansätzen nicht erfüllt ist. Die Nachvollziehbarkeit der Ergebnisse ist hingegen bei allen exakten Ansätzen erfüllt, da immer eine optimale Lösung berechnet wird. Die Ergebnisse lassen sich auch bei wiederholter Durchführung reproduzieren, weshalb die 5. Anforderung als erfüllt angesehen wird.

Die *Ansätze zur Neuplanung* können in Fälle mit einer einzelnen und mehreren Zielgrößen unterteilt werden, wobei die Ansätze der einzelkriteriellen Optimierung überwiegen. Beispiele sind u. a. [Ahma16, Amar09, Soli08, Xie08, Dunk05]. XIE ET AL. stellen einen Branch-and-Bound-Algorithmus vor, der ein Layout mit minimalen Materialhandhabungskosten generiert. Das Problem wird als Open Field Layout modelliert, bei dem die 2D-Geometrie der Funktionseinheiten und Aufnahme- und Abgabepunkte für den Materialtransport berücksichtigt werden. Weitere Nebenbedingungen wie die Medienverfügbarkeit werden nicht berücksichtigt [Xie08]. Demnach wird die 3. Anforderung zu 60 % erfüllt.

Zu den multikriteriellen, exakten Planungsansätzen für die Neuplanung gehören u. a. [Ghos16, Hamm16, Hadi13, Hath13, Mani11]. HATHORN ET AL. nutzen ein Modell der nicht linearen Programmierung, um die Transport- und Baukosten des Layouts zu minimieren. Das Optimierungsproblem wird zwar als Open Field Layout unter Berücksichtigung der 2D-Geometrie der Funktionseinheiten modelliert. Aufnahme- und Abgabepunkte für den Materialtransport, Medienverfügbarkeiten und die Höhe der Funktionseinheiten werden nicht berücksichtigt. Darüber hinaus werden alle Zielgrößen analytisch und nicht mittels EDS bestimmt [Hath13].

Autor und Charakterisierung		Anforderungen				
		I: Anwendbarkeit für die Neu- und Umplanung	II: Multikriterielle Optimierung	III: Realitätsnähe der Planungs- ergebnisse	IV: Skalierbarkeit Problemgröße	V: Nachvollzieh- barkeit
Exakte Planungsverfahren	Xie et al. Neuplanung Einzelkriteriell Analytisch					
	Hathorn et al. Neuplanung Multikriteriell Analytisch					
	Ulutas et al. Umplanung Multikriteriell Analytisch					
Approximative Planungsverfahren	Budi et al. Neuplanung Einzelkriteriell Analytisch					
	Aurich et al. Neuplanung Multikriteriell Analytisch					
	Zhao et al. Neuplanung Multikriteriell Analytisch					
	Song et al. Umplanung Multikriteriell Analytisch					
	Pourhassan et al. Neu- und Umplanung Multikriteriell Analytisch – EDS-basiert					
	Seidelmann et al. Umplanung Multikriteriell Analytisch – EDS-basiert					
Legende		Erfüllt Teilweise erfüllt Nicht erfüllt				

Abbildung 2-12: Zusammenfassende Bewertung bestehender automatisierter Layoutplanungsansätze

Die *Ansätze zur Umplanung* sind im Vergleich zu Neuplanungsansätzen immer multikriteriell, da der Umplanungsaufwand (z. B. Umstrukturierungskosten) eine zusätzliche Zielgröße darstellt. Beispiele für exakte Umplanungsansätze sind u. a. [Ulut15, Ulut12, Lahm05, Chen98]. ULUTAS ET AL. optimieren ein Layout hinsichtlich der analytisch berechneten Transport- und Umstrukturierungskosten. Die Modellierung weist jedoch Defizite auf, da alle Funktionseinheiten als eine Elementarzelle approximiert werden und die tatsächliche Geometrie nicht berücksichtigt wird. Zudem wird das Problem nicht als Open Field Layout modelliert und Medienverfügbarkeiten bleiben unberücksichtigt [Ulut15].

EDS-basierte Ansätze werden nur vereinzelt mit einer exakten Planungsmethode kombiniert. Die Integration erfolgt dabei nicht während der Optimierung, vielmehr werden Layoutalternativen bspw. auf Basis der Transportintensitäten generiert und abschließend in eine EDS überführt. Das beste Layout auf Basis der Simulationsergebnisse wird ausgewählt und in die Detailplanung überführt [Vite21]. Alternative Ansätze nutzen die Simulation, um die Einhaltung von Rahmenbedingungen für ein Layout zu überprüfen. Eine direkte Integration der Simulationsergebnisse in die Optimierung erfolgt jedoch nicht [Brac18a].

Zusammenfassend stellen die fehlende Einbindung von EDS in die Optimierung und eine unzureichende Skalierbarkeit aufgrund der Steigerung der Berechnungszeit bei großen Problemstellungen die primären Defizite der exakten, automatisierten Planungsansätze dar.

Bewertung der approximativen Planungsansätze

Approximative Planungsansätze adressieren die Defizite der exakten Planungsansätze, da diese Lösungen generieren, die sich dem Optimum annähern, weshalb sich die Berechnungszeit erheblich reduziert. Es existieren verschiedene Metaheuristiken, die für den jeweiligen Planungsfall angepasst werden. Beispiele sind GA, TS, Schwarm-Algorithmen oder Nachbarschaftssuchen [Zhu18]. Diese werden sowohl für die Neuplanung als auch für die Umplanung eingesetzt, jedoch adressieren die meisten Ansätze nur einen der beiden Fälle. Die Neuplanung wird zudem deutlich häufiger als die Umplanung betrachtet.

Der überwiegende Teil der Publikationen modelliert die Layoutplanung als *analytisches, einzelkriterielles Problem*, das für die *Neuplanung* genutzt wird. Hierzu zählen u. a. [Budi22, Guan16, Izad16, Rubi16, Jeri13]. BUDI ET AL. nutzen einen GA, um die aus dem Layout resultierenden Transportkosten zu minimieren. Die Autoren betrachten die 2D-Geometrie der Funktionseinheiten und modellieren das Problem als Open Field Layout. Aufnahme- und Abgabepunkte sowie Medienverfügbarkeiten werden jedoch betrachtet [Budi22].

Zu den Ansätzen der *multikriteriellen analytischen Neuplanung* zählen u. a. [Hu22, Auri20, Azad16, Salm15, Aiel12]. Der Ansatz nach AURICH ET AL. basiert auf einer Mischung zwischen GA und TS und ermöglicht es, 2D-Layoutvarianten unter Berücksichtigung von verschiedenen Zielgrößen und Nebenbedingungen zu generieren [Auri20]. Damit gehört dieser Ansatz zu einem der umfangreichsten Ansätze in der Literatur. Es werden jedoch keine dynamischen Zielgrößen mittels EDS betrachtet. Daneben ist der Non-dominated Sorting GA (NSGA) ein Ansatz, der auf dem konventionellen GA basiert und ausschließlich zur multikriteriellen Optimierung eingesetzt wird. Beispielhafte Publikationen zur Nutzung des NSGA sind durch [Huo21, Zhao20, Emam13] gegeben. In der Publikation von ZHAO ET AL. wird das FLP als Open Field Layout modelliert. Hierbei werden die Transportintensität und der Flächenbedarf

optimiert. Es erfolgt jedoch keine Betrachtung weiterer Nebenbedingungen und es werden keine EDS-Ergebnisse in der Optimierung genutzt [Zhao20].

Rein *analytische Ansätze zur Umplanung* von bestehenden Layoutvarianten wurden z. B. von [Song14, Şahi10, Dong09, Kult07, McKe06b] entwickelt. SONG ET AL. optimieren hinsichtlich der Materialhandhabungskosten und Umstrukturierungskosten für mehrere Perioden und nutzen einen GA für die Optimierung. Die Planungsaufgabe wird ebenfalls als Open Field Problem modelliert, Randbedingungen wie die Medienverfügbarkeit und Aufnahme- und Abgabepunkte für den Materialfluss werden nicht beachtet [Song14].

Eine deutlich kleinere Anzahl an Publikationen integriert *EDS-Ergebnisse* in die Optimierung. Exemplarische Publikationen für die Neuplanung sind [Azad16, Lee12, Jith10] und für die Umplanung [Seid22, Azim13].

POURHASSAN ET AL. nutzen einen GA und berücksichtigen die Materialhandhabungskosten und die Störungen im Materialfluss, die ein Ergebnis der EDS sind. Zudem werden Umstrukturierungskosten in zukünftigen Perioden beachtet, weshalb das 1. Kriterium erfüllt ist. Weitere Nebenbedingungen wie die Medienverfügbarkeiten werden nicht berücksichtigt [Pour17].

SEIDELMANN ET AL. betrachten einen reinen Umplanungsfall, bei dem ein GA genutzt wird, um die Umstrukturierungskosten, die mittlere Durchlaufzeit und die Maschinenauslastung zu optimieren. Defizite bestehen jedoch hinsichtlich der 3. Anforderung, da alle Funktionseinheiten die Größe einer einzigen Elementarzelle besitzen und daher Aufnahme- und Abgabepunkte vernachlässigt werden [Seid22].

Zusammenfassend führen Metaheuristiken je nach Algorithmus zu guten Lösungen unter Betrachtung der quantitativen Randbedingungen und Zielgrößen. Grundsätzlich lassen sich sowohl einzelkriterielle als auch multikriterielle FLPs lösen. Der größte Teil der Metaheuristiken berücksichtigt nur eine einzelne Zielgröße. Hierbei werden zumeist einzelne Materialflusskenngrößen, wie die Materialhandhabungskosten, als maßgebliches Kriterium betrachtet, während andere Zielgrößen unberücksichtigt bleiben. Die hierfür notwendige Formulierung von expliziten analytischen Bewertungsfunktionen ist bei komplexen Systemen aufgrund der nicht linearen Eigenschaften stark erschwert [Law13] und wird für metaheuristische Verfahren in der Layoutplanung meist mittels vereinfachender Annahmen und Abstraktion des Produktionssystems durchgeführt [Pére21]. Dies erhöht den manuellen Aufwand, erfordert a priori Wissen und schränkt die Gültigkeit der Modelle hinsichtlich des realen Produktionssystembetriebs ein. Nur wenige Ansätze betrachten eine Optimierung hinsichtlich mehrerer Zielgrößen und berücksichtigen dabei die Ergebnisse einer EDS. Daneben stellt die Nachvollziehbarkeit der Ergebnisse ein Defizit von Metaheuristiken dar, da Lösungen stochastisch erzeugt werden, die optimale Lösung auch bei langen Rechenzeiten nicht sicher erreicht wird und Lösungen nur bedingt reproduziert werden können. Daneben besteht ein Zielkonflikt zwischen der Lösungsqualität und der Rechendauer. Bei längeren Rechendauern können bessere Lösungen erwartet werden. Die erwartete Lösungsqualität ist bei großen Problemstellungen und gleicher Rechendauer niedriger als bei kleinen Problemstellungen. Zudem können die bestehenden Metaheuristiken Wissen aus bereits gelösten Problemen nicht zur Lösung ähnlicher Probleme verwenden, da für jedes Problem eine

zusätzliche Berechnung initialisiert werden muss. Hieraus wird deutlich, dass keiner der Ansätze alle Anforderungen vollständig erfüllt.

Eine mögliche leistungsfähige Alternative, die zur Erfüllung dieser Anforderungen beitragen kann, ist das RL. RL wird bereits erfolgreich für komplexe Entscheidungsprozesse in Kombination mit Simulationen verwendet und ermöglicht es, Wissen über dynamisch und stochastisch geprägte Systeme ohne eine umfangreiche mathematische Modellierung zu erlernen und Probleme mit hoher kombinatorischer Komplexität zu lösen.

2.4 Reinforcement Learning

2.4.1 Grundlagen des maschinellen Lernens

RL ist neben dem Supervised Learning (SL) (dt. überwachtes Lernen) und dem Unsupervised Learning (USL) (dt. unüberwachtes Lernen) eine Klasse des ML. Methoden des ML können als Teilgebiet der KI angesehen werden und besitzen die Fähigkeit, eigenständig Muster aus einem Datensatz zu extrahieren und dessen Zusammenhänge zu erlernen [Good16].

Die ersten Grundlagen des ML wurden in den 1950er Jahren von Arthur Samuel geschaffen, der ein selbstlernendes Computersystem für das Brettspiel Dame vorstellt und damit die Begriffe der KI und des ML prägt [Samu59]. Ein Teil moderner Methoden des ML baut auf künstlichen neuronalen Netzen (KNN) auf [Rose58]. Diese basieren auf Perzeptronen (künstliche Neuronen), die anfänglich von Frank Rosenblatt entwickelt wurden und die Lernfähigkeit des menschlichen Gehirns simulieren sollen. Das Kombinieren von mehreren künstlichen Neuronen in einer Schicht und das sequentielle Verknüpfen von mehreren Schichten wird als mehrschichtiges Perzeptron (engl. Multilayer Perceptron (MLP)) bezeichnet, das für die Approximation von nicht linearen Zusammenhängen genutzt werden kann [Rose58]. Abbildung 2-13 stellt ein MLP dar. Dieses besteht aus einer Eingangs- und Ausgangsschicht (engl. Input- und Outputlayer) und mindestens einer verdeckten Schicht (engl. Hidden-Layer). Jede Schicht umfasst mehrere künstliche Neuronen. Diese sind im Fall eines vollständig verknüpften KNN mit allen künstlichen Neuronen der Vorgänger- und Nachfolgerschicht über Kanten mit einem zugehörigen Kantengewicht verbunden [LeCu15]. Der Wert eines künstlichen Neurons N_j kann mittels Gleichung 2-2 berechnet werden. Demnach werden die Werte aller k Eingangsneuronen N_i mit dem zugehörigen Kantengewicht w_i gewichtet und addiert. Die Summe und der Bias-Wert b_j werden an die Aktivierungsfunktion ϕ übergeben, wodurch der Wert des Neurons bestimmt wird [Good16].

$$N_j = \phi \left(\sum_{i=1}^k w_i N_i + b_j \right) \quad (2-2)$$

Zu Beginn des Entwicklungsprozesses von KNN werden die Kantengewichte häufig zufällig initialisiert. Das Ziel des anschließenden Trainingsvorgangs besteht in der Anpassung aller Kantengewichte, sodass der Zusammenhang zwischen Eingangs- und Ausgangswerten bestmöglich abgebildet wird. Zur Anpassung der Kantengewichte wird eine Fehlerrückführung (engl. Backpropagation) genutzt. Hierfür wird im Fall des SL der berechnete Ausgabewert des KNN mit dem zu prognostizierenden Wert verglichen. Die Differenz zwischen den Werten wird

als Fehlerwert (engl. Loss) in eine Fehlerfunktion überführt und zur Anpassung der Kantengewichte genutzt. Das Training wird folglich als Optimierungsproblem aufgefasst, bei dem die Kantengewichte so eingestellt werden müssen, dass der Loss minimiert wird. Zur Anpassung der Gewichte wird, basierend auf der Fehlerfunktion, ein Gradient berechnet. Das Kantengewicht wird entsprechend des Gradienten und der Lernrate (engl. Learning Rate) angepasst, die bestimmt, wie stark der Gradient berücksichtigt wird. Demnach ist der Trainingsprozess ein iterativer Prozess, bei dem die Kantengewichte schrittweise verbessert werden [Chau95].

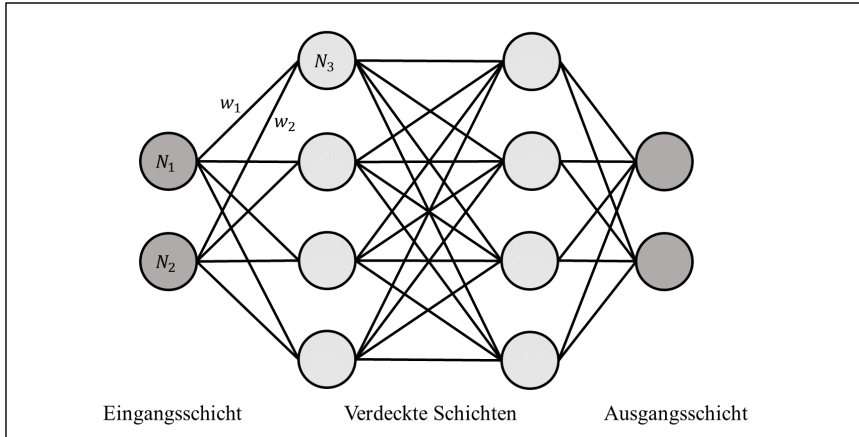


Abbildung 2-13: Darstellung eines MLP nach [Good16]

Das Erlernen von komplexen Wirkzusammenhängen ist häufig nur durch den Einsatz von KNNs mit vielen verdeckten Schichten und Neuronen pro Schicht möglich (Deep Learning). Die Anpassung der Gewichte auf Basis der berechneten Gradienten ist ein zeitaufwändiger und rechenintensiver Prozess [LeCu15]. Der Bedarf an leistungsstarken Computern war zu Beginn der Entwicklung von KNNs eine Einschränkung, weshalb tiefe KNN nicht zur Lösung industriell relevanter Problemstellungen genutzt wurden. Durch die Entwicklung von leistungsfähigen Graphikprozessoren (GPU) wurde diese Schwachstelle behoben, weshalb in den letzten Jahren immer leistungsfähigere KNN-Architekturen entwickelt wurden [Muth20].

Die für diese Arbeit relevanten KNN-Architekturen sind neben dem zuvor erklärten MLP (vgl. Abbildung 2-13) das Convolutional Neural Network (CNN) und das Graph Neural Network (GNN). Die KNN-Architekturen unterscheiden sich hinsichtlich der Eingangsdaten. MLPs verarbeiten mit jedem Neuron in der Eingangsschicht einen numerischen Wert. Eine Herausforderung stellen große Parameterräume dar. Für Bilddaten in Graustufen kann z. B. jedem Pixel ein Wert zwischen 0 und 1 zugeordnet werden, weshalb die ca. 2 Millionen Neuronen in der Eingangsschicht notwendig sind, um ein Bild der Auflösung 1920×1080 zu verarbeiten [O'Sh15]. CNNs adressieren dieses Problem, indem Convolutional-Layer und Pooling-Layer als Schichten hintereinander angeordnet werden. Im Convolutional-Layer wird ein Filter, der nur einen Teil des Bilds berücksichtigt (z. B. 3×3 Pixel) und die Identifikation von Nachbarschaftsbeziehungen ermöglicht, schrittweise über das Bild verschoben (siehe Abbildung 2-14). Dieser Bildausschnitt wird mit einer Kernelfunktion multipliziert und alle

Werte werden aufsummiert. Der Pooling-Layer wird genutzt, um die Dimension der Daten zu reduzieren, indem überflüssige Informationen verworfen werden. Ein Beispiel für eine Pooling-Funktion ist das Max-Pooling, bei dem z. B. aus einem 2×2 Bereich nur der maximale Wert weitergegeben wird. Hierdurch wird eine Datenreduktion von 75 % erzielt, die in höheren Berechnungsgeschwindigkeiten resultiert. Dieser Ablauf kann beliebig häufig wiederholt werden, bevor ein KNN mit vollständig verknüpften Schichten genutzt wird, um die gewünschten Ausgabewerte zu berechnen. Durch den Einsatz von CNNs können folglich Informationen und Nachbarschaftsbeziehungen in Bildern effizient extrahiert werden [Alba17].

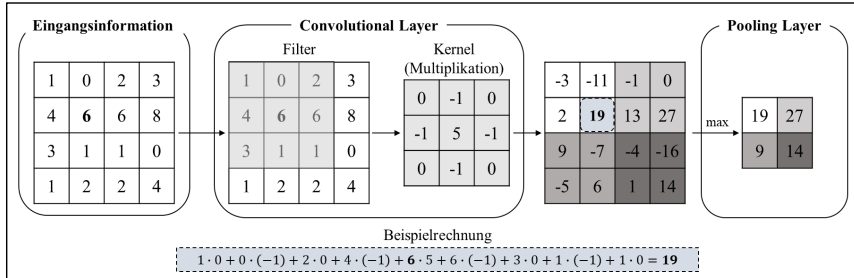


Abbildung 2-14: Beispielhafte Berechnung einer 2D-Convolution mit anschließendem Pooling

GNN eignen sich zur Verarbeitung von graphbasierten Daten, die im Kern Knoten und verbindende Kanten umfassen. Im Vergleich zu einem CNN hat ein Knoten nicht nur maximal 4 direkte Nachbarn (z. B. die Pixel rechts und links, oben und unten), sondern eine beliebige Anzahl an Nachbarn, deren individuelle Beziehung über die Kanteninformationen spezifiziert werden kann. Ein GNN kombiniert dabei die permutationsäquivalente Schicht (engl. Permutation-Equivariant-Layer), ein lokales Pooling und ein globales Pooling. In der ersten Schicht werden die Knoten im Graphen entsprechend ihrer Ähnlichkeitsbeziehung angeordnet. Im lokalen Pooling wird die Dimensionalität ähnlich zum CNN reduziert. Das globale Pooling stellt den gesamten Graphen in der finalen Repräsentation dar und wird als Übergabeschicht an ein KNN mit vollständig verknüpften Schichten genutzt. Die globale Repräsentation des Graphens ist dabei permutationsinvariant. Demnach verändert eine Permutation der Reihenfolge der Knoten und Kanten die globale Repräsentation nicht [Bron21, Scar09].

Es existieren zudem Weiterentwicklungen der Basis GNN-Architektur, wie Graph Convolutional Networks (GCN) [Kipf16] oder Graph Attention Neural Networks (GANN), die eine Erweiterung um den Attention Mechanismus darstellen, bei dem ein Fokus (engl. Attention) auf einen Teil der Daten gelegt wird, während die Relevanz eines anderen Teils abgeschwächt wird. Hierdurch wird die Leistungsfähigkeit der Architektur weiter gesteigert [Veli17].

Da das Training der unterschiedlichen KNN-Architekturen zeit- und datenaufwändig ist, können bereits trainierte KNN genutzt werden, um den Trainingsprozess zu beschleunigen. Dieser Vorgang wird als Transferlernen (engl. Transfer Learning) bezeichnet. Hierbei wird ein trainiertes KNN teilweise oder vollständig wiederverwendet und einzelne Schichten werden problemabhängig neu definiert. Durch die Wiederverwendung von vortrainierten KNN können erlernte Zusammenhänge bereits zu Beginn des Trainings genutzt werden. Folglich wird das Training eines transferierten Netzes als Fine-Tuning bezeichnet, da nur noch ein reduzierter

Anteil an Abhängigkeiten neu erlernt werden muss [Yang20, Weis16, Torr10]. Als Resultat können die Trainingsdauern reduziert und die Leistungsfähigkeit des trainierten KNN gesteigert werden [Pan10].

Die unterschiedlichen KNN-Architekturen werden sowohl im SL, USL und RL eingesetzt. Die Unterschiede zwischen den drei Klassen des maschinellen Lernens bestehen in der Datenbasis und dem Anwendungszweck. Bei Ansätzen des SL und USL liegt die Datenbasis bereits vor dem Training vor. Beim SL besteht der Datensatz aus Kombinationen von Eingangswerten und Zielwert, der prognostiziert werden soll. SL wird daher u. a. für Regressions- und Klassifikationsprobleme genutzt. Bei USL existiert kein zu prognostizierender Zielwert. Daher wird USL u. a. genutzt, um Datenpunkte entsprechend ihrer Ähnlichkeit in Gruppen zu aggregieren oder die Dimension eines Datensatzes zu reduzieren [Good16].

RL unterscheidet sich von USL und SL, da der Datensatz während des Trainings generiert wird. Das Ziel besteht darin, eine Strategie für sequentielle Entscheidungsprobleme zu erlernen [Kael96]. Die ersten Entwicklungen gehen auf MINSKY [Mins61] und WALTZ ET AL. [Walt65] zurück. Ein Meilenstein im Bereich der RL-Entwicklung wurde 2016 von Forschern der Google DeepMind Gruppe mit der Entwicklung von AlphaGo erzielt. AlphaGo wurde trainiert, um das Brettspiel Go zu erlernen, das aufgrund von 10^{172} verschiedenen Spielzuständen als unlösbares Problem für selbstlernende Systeme angesehen wurde [Silv16]. AlphaGo gewann 4-1 gegen den professionellen Go-Spieler Lee Sedol und nutzte dabei neu erlernte Strategien, die bis dato nicht eingesetzt wurden. RL eignet sich daher zur Lösung von hochkomplexen Entscheidungsproblemen, bei denen die Aufstellung von starren Entscheidungsregeln aufgrund der Problemgröße unzweckmäßig erscheint. Diese Probleme kann RL teilweise besser als ein Mensch und bestehende automatisierte Ansätze lösen und entwickelt hierbei unter Umständen neue Entscheidungsstrategien [Sutt18, Silv16]. Im Kontext der Produktion wird RL erfolgreich für die Routenplanung von Fördermitteln [Ye23], der Produktionsprogrammplanung [Cunh20] oder der Roboterkontrolle [Joha19] genutzt. Ein besonderes Potential liegt außerdem in der Verbindung von RL und Simulationsumgebungen. Hierbei ist es dem RL-Ansatz möglich, die häufig nicht linearen und analytisch schwer beschreibbaren Zusammenhänge zu erlernen und eigenständig neue Strategien zur Lösungsfindung zu erlernen [Li23].

2.4.2 Grundlagen des Reinforcement Learnings

Die allgemeine Struktur eines RL-Ansatzes ist in Abbildung 2-15 dargestellt. Die zwei zentralen Bestandteile sind der Agent und die Umgebung. Der Agent interagiert mit der Umgebung, indem dieser basierend auf dem aktuellen Zustand der Umgebung S_t eine Aktion A_t wählt und diese an die Umgebung übergibt. Die Umgebung setzt die gewählte Aktion um und wechselt zum Folgezustand S_{t+1} , der wiederum den Ausgangspunkt zur Wahl der nächsten Aktion A_{t+1} bildet. Außerdem berechnet die Umgebung einen Belohnungswert R_t der als Feedback zurück an den Agenten gegeben wird und beschreibt, wie gut oder schlecht eine Aktion war. Dieser Zyklus, bestehend aus dem Startzustand S_t , der gewählten Aktion A_t , dem resultierenden Folgezustand S_{t+1} und der Belohnung R_t wird als eine Transition (engl. Step) bezeichnet. Mehrere Transitionen werden zu einer Episode zusammengefasst. Am Ende einer Episode wird die Umgebung wieder in den initialen Startzustand zurückgesetzt [Sutt18].

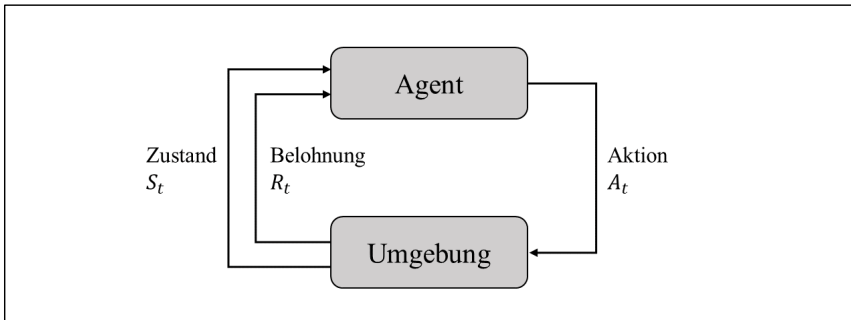


Abbildung 2-15: Struktur eines Reinforcement Learning Ansatzes nach [Sutt18]

Der Agent versucht eine Strategie (engl. Policy) zu erlernen, die genutzt wird, um eine Sequenz aus Aktionen zu wählen, welche die Summe der diskontierten Belohnungswerte über eine Episode maximiert. Demnach stellt die Belohnungsfunktion das zentrale Werkzeug dar, um das Verhalten des Agenten zu steuern. Daher sollte die Belohnungsfunktion so gewählt werden, dass ein gewolltes Verhalten in höheren Belohnungswerten als ein ungewolltes Verhalten resultiert. Der Definitionsvorgang der Belohnungsfunktion wird als Rewardshaping bezeichnet [Wier12]. Des Weiteren geht der Agent von Markov-Eigenschaften im Entscheidungsprozess aus. Das bedeutet, dass alle relevanten Informationen zur Wahl der Aktion im aktuellen Zustand enthalten sind. Informationen aus den vorangegangenen Transitionen werden bei der Entscheidungsfindung folglich nicht berücksichtigt [Sutt18, Papa87].

Im Anwendungsbeispiel eines Schachspiels erhält der Agent das Spielfeld als Zustandsinformation. Damit besitzt er alle relevanten Informationen, um eine Spielfigur und dessen neue Position auszuwählen. Es können verschiedene Belohnungsfunktionen gewählt werden. Im einfachsten Fall erhält der Agent immer eine Belohnung von 0. Lediglich am Ende des Spiels wird ein Belohnungswert von 1 generiert, wenn das Spiel gewonnen wurde und ein Wert von -1, falls das Spiel verloren wurde. Komplexere Belohnungsfunktionen können zusätzlich den Wert jeder Spielfigur berücksichtigen [Lai15].

Im Vergleich zu einer regelbasierten Lösungsstrategie besitzt der Agent zu Beginn des Trainings keine Vorkenntnisse über das Problem und kennt keine geeignete Lösungsstrategie. Lediglich das Feedback der Umgebung wird für das Erlernen der Strategie genutzt. Daher ist es notwendig, den Zusammenhang zwischen Zustand, Aktionen und resultierenden Belohnungen zu identifizieren. Die Explorationsstrategie gibt an, in welchem Verhältnis der Agent zufällige Aktionen wählt (engl. Exploration) und wie häufig die vielversprechendste Aktion gewählt wird (engl. Exploitation). Zufällige Aktionen sind notwendig, um neue Zusammenhänge zu entdecken, jedoch können vielversprechende Zustände, die zur Lösung des Problems beitragen, am besten gefunden werden, wenn auch vielversprechende Aktionen genutzt werden. Häufig werden daher zu Beginn des Trainings hohe Explorationsraten verwendet, sodass viele zufällige Aktionen ausgewählt werden, um die Eigenschaften des Problems zu identifizieren. Im Verlauf des Trainings wird diese schrittweise reduziert, wodurch zunehmend die vielversprechenden Aktionen ausgewählt werden [Sutt18].

Neben der Explorationsstrategie muss auch die Handhabung von ungültigen Aktionen definiert werden. Im Fall des Schachbeispiels würde eine ungültige Aktion immer dann auftauchen,

wenn der Agent eine Figur entgegen ihrer Bewegungsregeln ziehen möchte (z. B. diagonales Ziehen eines Turms). Es existieren zwei verbreitete Möglichkeiten, um ungültige Aktionen vorzubeugen. Im ersten Fall werden solche Aktionen bestraft. Hierdurch erlernt der Agent den Unterschied zwischen gültigen und ungültigen Aktionen. Eine Alternative stellt die Nutzung einer Maskierungsfunktion dar. Diese verhindert die Auswahl von ungültigen Aktionen und kann in einer Reduzierung des Trainingsaufwands resultieren [Huan20].

Falls RL für die Lösung multikriterieller Optimierungsprobleme eingesetzt werden soll, können verschiedene Grundstrukturen genutzt werden. In der Regel werden die einzelnen Optimierungsgrößen in Form einer Linearkombination aggregiert und in eine einzelne Belohnungsfunktion überführt. In diesem Fall ändert sich die Grundstruktur des RL-Ansatzes nicht. Der Ablauf entspricht daher dem zuvor geschilderten Schema. Der einzige Unterschied besteht in der Belohnungsfunktion, die sich aus mehreren Teilfunktionen zusammensetzt. Der Agent erhält jedoch weiterhin einen aggregierten Belohnungswert [Haye22]. Eine Alternative zur Nutzung einer Linearkombination wird in Abbildung 2-16 dargestellt. Hier wird für jede einzelne Optimierungsfunktion ein separater Agent trainiert. Jeder Agent bewertet den Aktionsspielraum entsprechend der individuellen Zielgröße. Erst im Anschluss an die individuelle Evaluation werden diese unter Nutzung von Gewichtungen aggregiert, um die vielversprechendste Aktion auszuwählen [Tajm18].

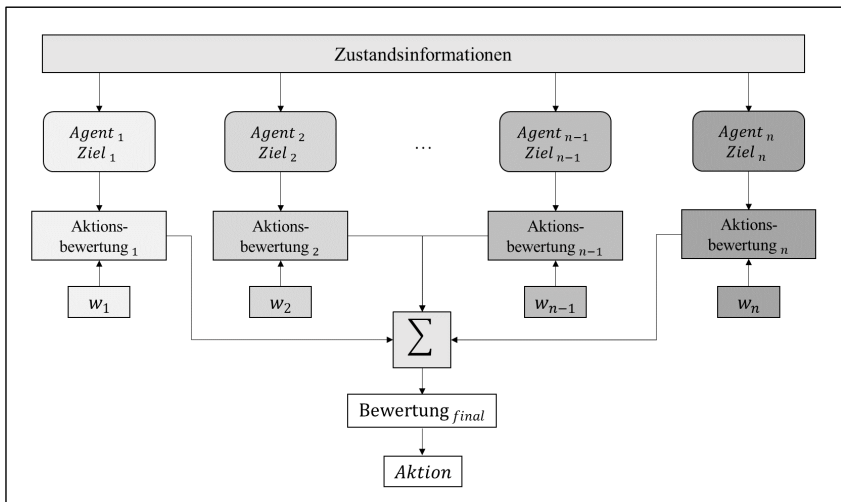


Abbildung 2-16: Struktur eines multikriteriellen RL-Ansatzes unter Nutzung eigener Agenten für jede Zielgröße nach [Tajm18]

A priori ist nicht ersichtlich, welche Struktur zu den besseren Ergebnissen führt, weshalb diese miteinander verglichen werden müssen, um sicherzustellen, dass die leistungsfähigste Struktur ausgewählt wird. Beide Strukturen können genutzt werden, um eine Pareto-Front zu generieren. Hierfür werden Experimente mit unterschiedlichen Zielgrößengewichtungen initialisiert und anschließend entsprechend der einzelnen Zielgrößen ausgewertet und in einem Graphen visualisiert [Wu01]. Alle Ergebnisse, die nicht dominiert werden, sind Teil der Pareto-Front. Eine Lösung wird dominiert, wenn es eine andere Lösung gibt, die in mindestens einer

Zieldimension eine bessere Bewertung erzielt, während alle verbleibenden Bewertungen gleich gut sind [Nara99]. Die Pareto-Front kann anschließend verwendet werden, um den Zielkonflikt zwischen den Optimierungsgrößen zu analysieren [Sale22]. Komplementäre Ziele resultieren dabei in einem linear ansteigenden Trend, während konkurrierende Ziele zu einer konvexen Kurve führen.

2.4.3 Reinforcement Learning Agenten

Die Auswahl einer Agenten-Architektur stellt eine zentrale Fragestellung bei der Entwicklung eines RL-Algorithmus dar. Es existiert eine Vielzahl verschiedener RL-Agenten, die durch ihr Trainingsverhalten, die Explorationsstrategie und die Modelkomplexität, gemessen an der Anzahl der zugehörigen Hyperparameter, charakterisiert werden [Sutt18].

RL-Agenten können anhand des Auswahlprozesses für Aktionen in Nutzen-basierte (engl. Value-based), Policy- basierte und hybride (Actor-Critic) Ansätze unterteilt werden. Ein Agent wird als Nutzen-basierter Ansatz eingeordnet, wenn eine Nutzenfunktion die Grundlage für den Auswahlprozess einer Aktion ist. Nutzenfunktionen können aktions- oder zustandsorientiert sein. Aktionsorientierte Nutzenfunktionen (engl. Action Value Function) approximieren den Nutzen der Aktionsausführung für einen definierten Zustand. Zustandsorientierte Nutzenfunktionen (engl. State Value Function) approximieren den Nutzen, der durch das Erreichen des Zustands selber entsteht. Der Agent schätzt folglich den Nutzen einer Aktion bzw. des Folgezustands ein und wählt die vielversprechendste Aktion aus [Li17].

Policy-basierte Ansätze verwenden bei der Aktionswahl keine Nutzenfunktion als Zwischenschritt. Stattdessen wird eine Policy (Strategie) erlernt, die jeder Aktion zustandsabhängig Auswahlwahrscheinlichkeiten zuordnet. Die Policy entspricht demnach einer Wahrscheinlichkeitsfunktion, die direkt zur Aktionsauswahl genutzt wird [Li17].

Actor-Critic Ansätze kombinieren die Grundprinzipien der Nutzen- und Policy-basierten Ansätze. Die Struktur des Ansatzes, der aus zwei Teilen, dem Actor und dem Critic besteht, ist in Abbildung 2-17 visualisiert. Der Policy-basierte Actor wählt die nächste Aktion basierend auf der Policy aus. Die Policy des Actors wird durch das Feedback des Nutzen-basierten Critics aktualisiert. Als Grundlage für die Verbesserung der Policy des Actors wird der zu minimierende Temporale-Differenzfehler (TD-Fehler) genutzt. Dieser gibt die Abweichung zwischen dem durch den Agenten erwarteten Nutzen und dem tatsächlich realisierten Nutzen an. Durch die Integration beider Grundprinzipien in einem hybriden Ansatz soll ein besseres Lernverhalten erreicht werden als bei rein Nutzen- oder Policy-basierten Ansätzen [Mnih16].

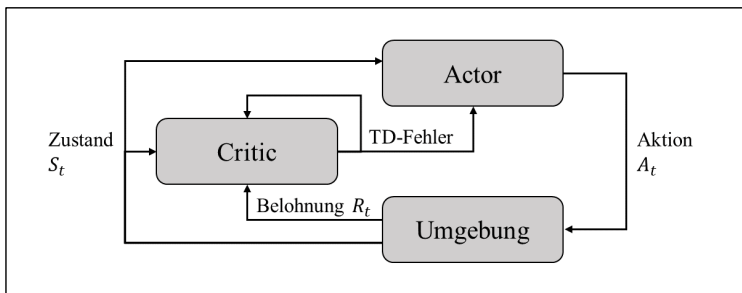


Abbildung 2-17: Struktur eines Actor-Critic Ansatzes nach [Szep10]

Darüber hinaus unterscheiden sich RL-Ansätze in ihrem Lernverfahren. Hierbei wird zwischen On-Policy- und Off-Policy-Ansätzen unterschieden (siehe Abbildung 2-18). Ein On-Policy-Agent verbessert seine Strategie anhand der gerade durchgeführten Transition. Off-Policy-basierte Agenten verbessern ihre Strategie auf der Grundlage früherer Transitionen. Hierfür wird ein sogenannter Replay-Buffer verwendet, in dem Transitionen vergangener Schritte gespeichert werden. Für das Training werden vielzählige Transitionen aus dem Replay-Buffer gezogen (engl. Sampling) und zur Strategieverbesserung durch den Agenten bewertet. Folglich ist bei On-Policy Ansätzen die Strategie, die zur Entscheidungsfindung der Aktion geführt hat, die selbe Strategie, die aktualisiert wird, während sie sich bei Off-Policy Ansätzen unterscheiden können [Dong20].

Es existiert eine Vielzahl von verschiedenen RL Agenten mit unterschiedlichen Lernansätzen und Trainingsverhalten. Im Vorfeld ist nicht klar einschätzbar, welcher Agent ein Problem am leistungsfähigsten löst. Demnach ist der Vergleich von unterschiedlichen Agenten bei der Entwicklung eines RL Ansatzes notwendig. Nachfolgend werden die für diese Dissertation genutzten Agenten und deren Charakteristika beschrieben. Die ausgewählten Agenten decken die verschiedenen Agentenkategorien und Lernverfahren ab und werden regelmäßig in der Forschung und industriellen Praxis eingesetzt (siehe Tabelle 2-3).

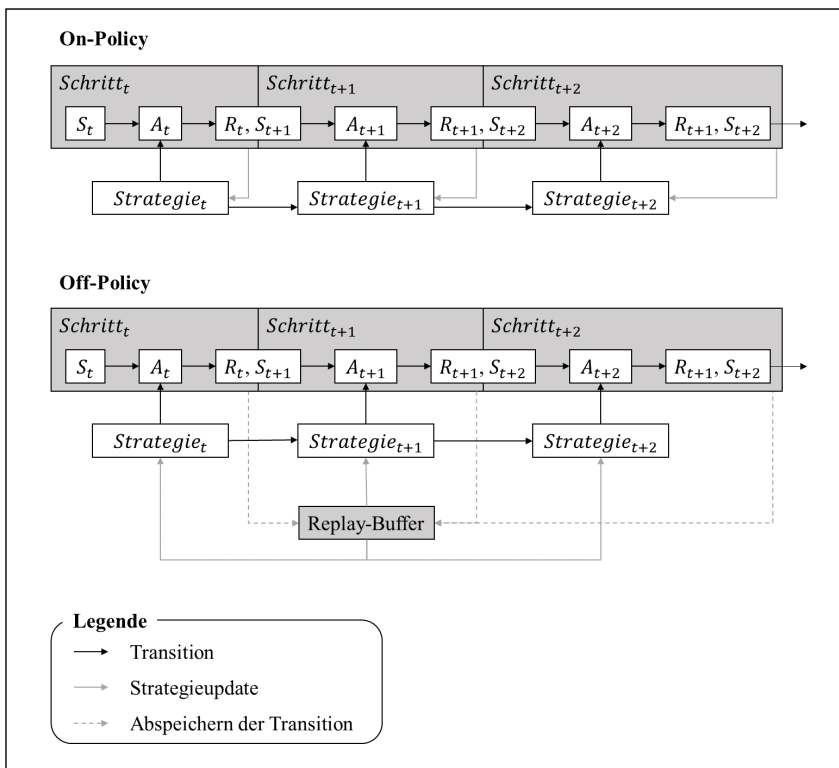


Abbildung 2-18: Unterschied zwischen On- und Off-Policy Ansätzen – eigene Darstellung

Tabelle 2-3: Überblick über Reinforcement Learning Agenten

Agent	Agentenkategorie	Lernverhalten
Vanilla Policy Gradient (VPG)	Policy-basiert	On-Policy
Rainbow Deep Q Learning (Rainbow DQN)	Nutzen-basiert	Off-Policy
Advantage Actor Critic (A2C)	Hybrid	On-Policy
Proximal Policy Optimization (PPO)	Hybrid	On-Policy
Soft Actor Critic (SAC)	Hybrid	Off-Policy

Der **Vanilla Policy Gradient (VPG)** Agent ist ein Policy-basierter Ansatz, der einen On-Policy Lernansatz nutzt, um seine Strategie zu verbessern. Die Strategie wird durch ein KNN abgebildet, dessen Parameter unter Nutzung des stochastischen Gradientenanstiegs (engl. Stochastic Gradient Ascent) angepasst werden. Die Grundlage für die Gradientenberechnung stellt der undiskontierte Belohnungswert dar, der unter Nutzung der aktuellen Strategie erwartet wird. Zudem wird die Vorteilsfunktion (engl. Advantage Function) berücksichtigt [Sutt00]. Diese beschreibt die Differenz zwischen der aktionsorientierten Nutzenfunktion und der zustandsorientierten Nutzenfunktion und kann durch den TD-Fehler approximiert werden [Brai93]. Details zur mathematischen Implementierung werden in [Sutt00] beschrieben.

Ein Vorteil rein Policy-basierter Ansätze wie dem VPG ist die ausgeprägte Konvergenzeigenschaft [Sutt18]. Nachteile bestehen in der hohen Varianz der geschätzten Gradienten [Gron12] und in der Empfindlichkeit der Leistungsfähigkeit des VPG in Bezug auf die Anfangswerte des KNNs. Außerdem kann ein konvergierendes Trainingsverhalten bei nicht-linearen Nutzenfunktionen nicht garantiert werden [Schu17].

Der **Rainbow Deep Q Learning (Rainbow DQN)** Agent ist die fortschrittlichste Variante der Q-Learning Agenten. Q-Learning Ansätze gehören zu den Nutzen-basierten Ansätzen, bei denen eine aktionsorientierte Nutzenfunktion (Q-Funktion) verwendet wird. Bei konventionellem Q-Learning wird eine Tabelle genutzt, die alle Aktion-Zustand-Kombinationen umfasst. Der erwartete aktionsorientierte Nutzen für jedes Aktion-Zustand-Paar (Q-Werte) wird in die Tabelle eingetragen und entsprechend der Bellmann-Gleichung aktualisiert [Watk92]. Eine Einschränkung des klassischen Q-Learnings ist die Notwendigkeit, alle Aktion-Zustand-Kombinationen in einer Tabelle zu erfassen, da viele Probleme nur durch große oder kontinuierliche Zustandsräume beschrieben werden können. Durch die Anzahl an möglichen Kombinationen ist die Anwendbarkeit beschränkt [Mnih13].

Der Deep Q Learning Ansatz von MNIH ET AL. adressiert diese Problematik, indem die Tabelle durch ein KNN ersetzt wird, das die Q-Werte approximiert. Hierdurch kann Q-Learning auch für größere Probleme genutzt werden. Der Deep Q Learning Agent neigt jedoch dazu, den Q-Wert zu optimistisch zu schätzen, da das gleiche KNN zur Wahl und Evaluation einer Aktion genutzt wird. Daraus kann ein instabiles Trainingsverhalten resultieren, bei dem sich die Leistungsfähigkeit des Agenten zeitweise verschlechtert, statt besser zu werden [Mnih13].

Der Double Deep Q Learning (DDQN) Agent ist hingegen durch ein stabiles und konvergierendes Trainingsverhalten charakterisiert. Statt einem KNN werden zwei KNN, das

Q-Netz und das Ziel-Netz, parallel genutzt. Das Q-Netz wird zur Aktionswahl verwendet. Die Evaluation einer Aktion, die zur Verbesserung des Q-Netzes genutzt wird, wird vom Ziel-Netz durchgeführt. Durch diese Entkopplung wird das Problem der überoptimistischen Einschätzung der Q-Werte behoben [Hass15].

Der Rainbow DQN kombiniert verschiedene Weiterentwicklungen in einem Ansatz und baut auf dem DDQN-Agenten auf [Hess17]. Es wird ein Off-Policy Lernverhalten unter Verwendung eines Replay-Buffers genutzt. Das bedeutet, dass für das Training Transitionen aus dem Replay-Buffer gezogen werden. Beim DDQN-Agenten wird jede Transition im Replay-Buffer mit der gleichen Wahrscheinlichkeit gezogen. Der Rainbow DQN nutzt stattdessen das Prioritized Experience Replay (PER). Anstatt jede Transition mit der gleichen Wahrscheinlichkeit für das Training zu nutzen, ordnet das PER den gespeicherten Transition Prioritäten auf Basis des TD-Fehlers und der Aktualität der Transition zu [Hess17, Scha15]. Zudem wird der Duellierungsmechanismus nach WANG ET AL. in den Rainbow DQN integriert. Der Duellierungsmechanismus teilt die KNNs in zwei Ströme auf, um den Nutzen des aktuellen Zustands und den Advantage-Wert separat voneinander zu schätzen. Anschließend werden die beiden Ströme wieder zusammengeführt, sodass gelernt werden kann, welche Zustände wertvoll sind, ohne dass der Einfluss jeder einzelnen Aktion für jeden Zustand explizit gelernt werden muss [Hess17, Wang15]. Darüber hinaus wird das einstufige Training des DDQN-Agenten auf ein mehrstufiges Training (engl. Multi-Step Learning) erweitert. In der Berechnung des Fehlers, der für das Training genutzt wird, werden demnach mehrere Vorwärtstransitionen anstelle einer einzelnen Vorwärtstransition verwendet [Hess17, Sutt88]. Ferner werden Noisy-KNNs integriert, um die Leistungsfähigkeit mit geringem zusätzlichem Rechenaufwand zu verbessern [Fort17, Hess17]. Verteilungsbasiertes RL (engl. Distributional RL) ist die letzte Erweiterung, die es dem Agenten erlaubt, die Verteilung der Belohnungen explizit zu lernen, anstatt nur den Mittelwert zu schätzen, was zu einem schnelleren und stabileren Lernverhalten führen kann [Bell17, Hess17]. Implementierungsdetails des Rainbow DQN können in der Originalveröffentlichung nachgeschlagen werden [Hess17].

Durch die Integration dieser Erweiterungen stellt der Rainbow DQN einen der leistungsfähigsten RL-Agenten dar. Er kann für Probleme mit einem hochdimensionalen Zustandsraum verwendet werden. Allerdings ist dieser Agententyp nur für einen diskreten Aktionsraum geeignet. Nachteilig ist zudem, dass jede Erweiterung individuelle Hyperparameter besitzt, weshalb die Komplexität der Hyperparameteroptimierung im Vergleich zum DDQN-Agenten steigt [Hess17].

Hybride Ansätze versuchen die Vorteile der Policy-basierten und Nutzen-basierten Ansätze zu kombinieren. Ein Basisagent ist der **Advantage Actor-Critic (A2C)**, der als On-Policy Lernansatz kategorisiert wird. Der A2C wird ähnlich wie der VPG auf Basis von Gradienten optimiert, die auf Basis der nächsten n_T Transitionen (engl. n-Step Bootstrapping) berechnet werden [Mnih16]. Im Vergleich zum VPG ist die Varianz des geschätzten Gradienten beim A2C geringer [Schu15]. Allerdings weist der Trainingsprozess Schwächen hinsichtlich der Stabilität auf [Schu17].

Weiterentwicklungen des Basis A2C-Agenten versuchen diese Defizite abzuwenden und die Leistungsfähigkeit der hybriden Agenten weiter zu steigern. Zu den am häufigsten eingesetzten Ansätzen zählen der On-Policy Proximal Policy Optimization (PPO) Agent [Schu17] und der Off-Policy Soft Actor-Critic (SAC) Agent [Haar18b].

Der **PPO-Agent** kompensiert das Problem des instabilen Trainingsverhaltens, das bei der Aktualisierung des KNNs entsteht und gilt als dateneffizienter als der A2C-Agent. Bei vielen Policy-basierten Agenten führen zu große Parameteranpassungen des KNNs zu einer Verschlechterung der Strategie, wodurch das Trainingsverhalten instabil wird. Zudem benötigen die KNN zum Erlernen eines Verhaltens einen umfangreichen Datensatz, was ein Nachteil für On-Policy Ansätze ist, da diese die aktuelle Erfahrung nach der Aktualisierung der KNN-Parameter verwerfen [Schu17]. Der PPO baut auf dem robusten Trainingsverhalten des Trust Region Policy Optimization (TRPO) Ansatzes auf und verbessert diesen durch eine einfachere Implementierung, eine allgemeine Anwendbarkeit und eine Reduzierung der benötigten Trainingsdaten [Hees17]. Die zentrale Idee des PPO ist, dass die Abweichung zwischen der alten und aktualisierten Strategie nur in einem definierten Intervall liegen darf. Hierdurch wird das Trainingsverhalten stabilisiert, weshalb der PPO häufig für Kontrollprobleme mit kontinuierlichem Zustands- und Aktionsraum genutzt wird. Jedoch ist die Leistungsfähigkeit stark von den Hyperparametern abhängig und es besteht keine Konvergenzgarantie [Schu17].

Der **SAC-Agent** wurde von HAARNOJA ET AL. entwickelt und reagiert weniger sensitiv auf Hyperparameteränderungen als der PPO-Agent [Haar18b]. Er basiert auf dem Maximum-Entropie RL Rahmenwerk und ist als Off-Policy Ansatz klassifiziert [Haar18a]. Der SAC nutzt je eine Approximationsfunktion für die zustandsorientierte Nutzenfunktion, die aktionsbasierte Nutzenfunktion und die Policy-basierte Aktionsstrategie. Zur Aktualisierung der Approximationsfunktionen werden, wie beim Rainbow DQN, vergangene Transitionen aus dem Replay-Buffer entnommen. Der SAC berücksichtigt die Entropie in der Aktionswahl, welche als Maß für die Zufälligkeit in der aktuellen Strategie angesehen werden kann [Haar18b]. Hierdurch wird die Strategie des Agenten zusätzlich verbessert [Todo08]. Dabei resultiert ein hohes Maß an Entropie in höheren Explorationsraten [Haar18b]. Der SAC wird als skalierbarer, dateneffizienter und stabiler Agent angesehen und eignet sich zum Lösen von komplexen, hochdimensionalen Aufgaben [Haar18b, Gu16].

Es existiert eine Vielzahl von Agenten, die Unterschiede in der Struktur, dem Lernverhalten, der Parameterkomplexität und der Leistungsfähigkeit aufweisen. Modernere Ansätze sind dabei häufig mit einer besseren Leistungsfähigkeit, aber komplexeren Strukturen verbunden, weshalb die Verbesserung der Hyperparameter erschwert ist und die Nachvollziehbarkeit des Lernprozesses verringert wird.

2.4.4 Erklärbares Reinforcement Learning

RL-Ansätze besitzen, im Gegensatz zu konventionellen Metaheuristiken, die Fähigkeit, Muster und Zusammenhänge eines Problems zu erlernen. Diese Information erlaubt es, Fragestellungen hinsichtlich der Feature-Relevanz und der zentralen Einflussfaktoren zur Lösung des Problems zu beantworten [Well21, Puiu20]. Daher besitzt RL das Potential, die bestehenden Nachteile konventioneller Metaheuristiken im Kontext der Nachvollziehbarkeit zu überwinden, wenn geeignete Analyseansätze eingesetzt werden [Well21]. Die Entwicklung solcher Ansätze ist der Fokus des Forschungsfelds der XAI [Puiu20]. Dabei besteht die Zielsetzung in der Reduzierung der Informationslücke zwischen dem Nutzer und dem Ansatz der KI. Um dieses Ziel zu erreichen, verarbeiten, aggregieren und bewerten XAI-Ansätze verschiedene Informationsquellen, z. B. die Sensitivität von Input-Output-Beziehungen

[Conf21, Gunn19, Adad18, Glas08]. Die Informationslücke variiert in Abhängigkeit vom genutzten Ansatz. So besitzen Entscheidungsbäume eine höhere Transparenz als tiefe KNN, die als Black-Box aufgefasst werden [Cast16]. Der Zusammenhang zwischen Modelltransparenz und -leistungsfähigkeit wird durch den Transparency-Accuracy-Trade-Off beschrieben, der aussagt, dass komplexe Modelle, die in einer ausgeprägten Intransparenz resultieren, leistungsfähiger sind als Modelle mit einer geringen Komplexität [Dosi18].

Die angestrebten Ziele bei der Entwicklung und dem Einsatz von XAI-Ansätzen variieren quellenabhängig. ADADI ET AL. beschreiben mögliche Potentiale, die durch den Einsatz von XAI-Ansätzen realisiert werden können. Hierzu zählen u. a. die Prävention vor Fehlverhalten durch bessere Erklärbarkeit, die Steigerung der Leistungsfähigkeit des KI-Ansatzes und die Identifikation von Zusammenhängen, die auf andere Anwendungsbereiche übertragen werden können [Adad18]. ARRIETA ET AL. definieren die folgenden neun Zielsetzungen: Vertrauenswürdigkeit, Kausalität, Übertragbarkeit, Information, Vertrauen, Fairness, Zugänglichkeit, Interaktivität und Datenschutzbewusstsein [Arri19]. Der Autor HAGRAS definiert die Ziele Transparenz, Kausalität, Fairness und Sicherheit [Hagr18]. Es ist hervorzuheben, dass kein Ansatz existiert, der alle Ziele gleichzeitig erfüllt. Daher ist die Festlegung des XAI-Ziels ein erster Schritt bei der Entwicklung eines XAI-Ansatzes.

Darüber hinaus werden in der Literatur unterschiedliche Definitionen für die Schlüsselbegriffe von XAI verwendet. Interpretierbarkeit und Erklärbarkeit werden in der Literatur oft synonym verwendet, obwohl verschiedene Aspekte damit verbunden sind [Dosi18]. Das Ziel der Interpretierbarkeit ist es, abstrakte Konzepte in einen für Menschen verständlichen Bereich zu übersetzen. Erklärbarkeit zielt hingegen darauf ab, die entscheidungsbeeinflussenden Merkmale in der interpretierbaren Domäne zu identifizieren [Mont18]. Nachvollziehbarkeit und Transparenz kann als Synonym für Interpretierbarkeit verstanden werden [Frei14], wobei die Transparenz Einblick in die Funktionslogik des Modells gewährt [Lipt16]. Keine der oben genannten Definitionen ist jedoch spezifisch oder restriktiv genug, um eine Formalisierung zu ermöglichen. Sie hängen von den Expertenkenntnissen des Nutzers, seinen Präferenzen und zusätzlichen Kontextvariablen ab [Dosi18].

Ein weiterer Einflussfaktor bei der Entwicklung eines XAI-Ansatzes ist der Nutzer, der als Zielgruppe definiert wird. Die Zielgruppe variiert hinsichtlich ihres Wissensstands und Anforderungen an den XAI-Ansatz in der Nutzungsphase. ARRIETA ET AL. unterscheiden zwischen fünf möglichen Zielgruppen. Die *Experten* stellen die erste Gruppe dar, wenden den KI-Ansatz aktiv in ihrem Alltag an und besitzen Expertise im Themenfeld KI. Neben der Verbesserung der Vertrauenswürdigkeit bei der Anwendung des KI-Ansatzes steht auch die Generierung von wissenschaftlichem Mehrwert im Fokus der Zielgruppe. Die Gruppe der *Nutzer ohne KI-Expertise* ist hingegen an fairen Entscheidungen interessiert, während die Gruppe der *Entwickler und Data Scientists* zusätzliches Wissen über die Funktionalität und die Zusammenhänge des Ansatzes erfassen möchte. Die vierte Gruppe sind *Führungskräfte*, die dazu verpflichtet sind, die Fähigkeit des KI-Ansatzes hinsichtlich der Einhaltung von Vorschriften zu bewerten. Die letzte Gruppe sind *Regulierungsbehörden*, welche die Konformität des KI-Ansatzes mit der Gesetzgebung bescheinigen wollen [Arri19].

Demnach müssen XAI-Ansätze auf den genutzten KI-Methode, die angestrebten XAI-Ziele, die Zielgruppe und das zugrunde liegende Problem, das durch den Einsatz von KI gelöst werden soll, abgestimmt sein. Daraus leiten sich folgende Fragen ab, die durch eine oder die

Kombination von XAI-Ansätzen beantwortet werden sollen: Was sind die Fähigkeiten und Kenntnisse des KI-Ansatzes? Was hat der KI-Ansatz getan? Was macht der KI-Ansatz aktuell? Was wird der KI-Ansatz tun? [Gunn19] Außerdem können Warum- und Warum-nicht-Fragen gestellt werden, um die Verständlichkeit von KI-Ansätzen weiter zu steigern [Lim09].

Das Forschungsfeld des erklärbaren RL (XRL) untersucht in Analogie die Potentiale und Herausforderungen im Kontext von RL [Sae23]. XRL-Ansätze werden in der Literatur anhand unterschiedlicher Kategorien charakterisiert. PUIUTTA ET AL. unterteilen diese abhängig vom Zeitpunkt der Informationsextraktion in intrinsische und post-hoc XRL-Ansätze und differenzieren Ansätze mit einem globalen oder lokalen Fokus [Puiu20]. MILANI ET AL. definieren drei Kategorien. Folglich können XRL-Ansätze die Feature-Relevanz, das Lernverhalten und die Markov Eigenschaften oder die erlernte Strategie (Policy) untersuchen [Mila22]. Im Rahmen dieser Dissertation wird die Unterteilung nach WELLS ET AL. genutzt, die fünf Zwecke (Zusammenarbeit mit Menschen, Visualisierung, Zusammenfassung der erlernten Strategie, abfragebasierte Erklärung und Verifikation) identifizieren, die ein XRL-Ansatz als Zielsetzung nutzen kann [Well21].

Die Zusammenarbeit mit Menschen (engl. Human Collaboration) steht dann im Fokus, wenn der RL-Ansatz direkt mit dem Menschen interagiert. Dabei bezieht sich die Interaktion darauf, dass der menschliche Nutzer den Agenten in der Entscheidungsfindung beeinflusst, indem menschliche Erfahrung in die Entscheidungsfindung des RL-Ansatzes mit einfließt. Dies ist insbesondere dann von Relevanz, wenn der Entscheidungsprozess nur teilweise beobachtbar ist [Well21]. Eine Anwendung im Produktionskontext wurde von HAYES ET AL. untersucht, die XRL-Ansätze zur Transparenzsteigerung während der Mensch-Roboter-Kollaboration nutzen [Haye17]. Für die Layoutplanung kann diese Kategorie jedoch vernachlässigt werden, da in der Optimierungsphase in der Regel keine Interaktion zwischen dem Planer und dem RL-Ansatz stattfindet.

Außerdem können XRL-Ansätze Informationen graphisch aggregieren und bereitstellen. Dabei ist die Saliency Map eine der am häufigsten genutzten Visualisierungsformen [Well21], da sie die Relevanz von Eingangsinformationen (engl. Feature) auf die Aktionswahl des RL-Ansatzes darstellt [Heui20]. GOLDMAN ET AL. verwenden den Ansatz der Saliency Map, um die Robustheit eines KNN-Modells zu untersuchen. Hierbei werden Verschiebungssignale im Schweißprozess, abgegebene Leistung und akustische Signale als Eingaben für eine Klassifizierungsaufgabe von Teilen mit guter und schlechter Qualität verwendet [Gold21]. Da geeignete Visualisierungstechniken die Zugänglichkeit zu komplexen Zusammenhängen verbessern können, werden sie im Kontext der Layoutplanung als wichtig angesehen, sodass auch Planende ohne tiefgreifendes RL-Wissen relevante Informationen erfassen können.

XRL-Ansätze, welche die erlernte Strategie zusammenfassen, werden als Policy Summarization bezeichnet. Diese Methoden können genutzt werden, um zu untersuchen, warum Aktionen in einer bestimmten Situation ausgewählt wurden [Well21]. Das so generierte Verständnis über das Verhalten des Agenten kann verwendet werden, um verschiedene Agententypen miteinander zu vergleichen oder den Grad der Entscheidungsautonomie festzulegen [Amir18]. Policy Summarization Ansätze besitzen somit das Potential, die Nachvollziehbarkeit von Entscheidungen im Kontext der Layoutplanung zu verbessern.

Daneben können Policy Summarization Ansätze eine Grundlage bieten, um abfragebasierte Erklärungen (engl. Query-based Explanation) zu ermöglichen [Haye17]. Diese nutzen Spracheingaben, um Erklärungen von Agenten zu erhalten [Well21] und werden u. a. im Bereich der Mensch-Roboter-Kollaboration genutzt [Srid19].

Verifikationsmethoden werden vorrangig in sicherheitsbezogenen RL-Anwendungen genutzt. Anwendungsfelder sind die Überprüfung der Einhaltung eines vorgegebenen Verhaltens, sodass einem Fehlverhalten in sicherheitskritischen Situationen vorgebeugt wird [Well21]. Da Arbeitssicherheitsaspekte in der frühen Phase der Fabriklayoutplanung eine untergeordnete Wichtigkeit besitzen, werden Verifikationsmethoden nachfolgend nicht weiter betrachtet [Wien14a].

XRL-Ansätze können zusammenfassend mit unterschiedlichen Aspekten die Erklärbarkeit und Transparenz von RL-Ansätzen verbessern. Daher müssen XRL-Ansätze auf die Problem- und Zielgruppeneigenschaften abgestimmt sein. Im Kontext der Fabriklayoutplanung sind insbesondere Visualisierungs- und Policy Summarization Ansätze von Interesse. Die Visualisierung trägt zur Verbesserung der Zugänglichkeit der Ergebnisse bei, da relevante Informationen aggregiert und auch für Planenden ohne umfassendes RL-Wissen anschaulich werden. Eine Steigerung der Transparenz und der Nachvollziehbarkeit von Entscheidungen kann durch Policy Summarization Ansätze erreicht werden. Im Folgenden werden diese Ansätze näher erläutert.

Eine **Policy Summarization** aggregiert die erlernte Strategie des RL-Ansatzes auf übergeordneter Ebene [Well21]. Das Ziel besteht nicht darin, einzelne Aktionen zu erklären, sondern in der Identifikation von Einflussfaktoren auf die Aktionswahl oder in der Zusammenfassung der Gesamtmenge der gewählten Aktionen [Haye17]. Es existieren verschiedene Ansätze der Policy Summarization. Hierzu zählt u. a. der Instruction-based Behavior Explanation Ansatz. Dieser ermöglicht es, das zukünftige Verhalten des Agenten mittels Code-Annotationen zu beschreiben, indem menschenlesbare Bezeichnungen für Funktionen und Variablen bereitgestellt und Kausaldiagramme erstellt werden [Fuku17]. Weitere Policy Summarization Ansätze sind das Symbolic Deep RL [Lyu19] und das Programmatically Interpretable RL [Verm18]. Beide Konzepte verwenden symbolische Darstellungen und Suchalgorithmen, um für den Menschen lesbare Richtlinien zu generieren. Darüber hinaus können Ansätze der Policy Summarization zur Erklärung einzelner Aktionen genutzt werden. CRUZ ET AL. verwenden einen solchen Ansatz, um den durch den RL-Ansatz approximierten Nutzen und die Erfolgswahrscheinlichkeiten für jede Aktion zu vergleichen. Der Ansatz ermöglicht es, die von einem Roboter getroffenen Entscheidungen einem nicht fachkundigen menschlichen Benutzer zu erklären [Cruz23]. Durch die Analyse von Stärken und Schwächen in der erlernten Strategie können die relevantesten Einflussfaktoren zur erfolgreichen Aktionsauswahl identifiziert werden. Diese Informationen werden zur Anpassung des Trainings verwendet und können die Leistungsfähigkeit des Ansatzes weiter verbessern. Darüber hinaus kann die Aggregation der erlernten Strategie in einem zugänglichen Format dazu beitragen, dass auch Planer oder Entscheidungsträger, die nicht über vertiefte Kenntnisse im Bereich des ML verfügen, die Ergebnisse besser verstehen und anwenden können. Folglich ist es notwendig, die Ergebnisse des Policy Summarization Ansatzes geeignet zu visualisieren [Haye17].

Eine Darstellung von Informationen als **Saliency Map** kann im Kontext der Fabriklayoutplanung vorteilhaft sein. Saliency Maps visualisieren den Einfluss einer Zustandsinformation (Input-Feature) auf den Entscheidungsprozess. Im Kontext der Fabrikplanung sind sog. Perturbation-Ansätze von Relevanz, da sie das Verhalten des RL-Ansatzes analysieren, wenn kleine Änderungen in den Zustandsinformationen vorliegen. Somit kann z. B. analysiert werden, wie sich der Entscheidungsprozess verändert, wenn sich die Transportintensität zwischen zwei Funktionseinheiten ändert. Der Vorteil von Perturbation-Ansätzen ist die Modellunabhängigkeit, da sie lediglich die Eingangsinformationen und die gewählten Aktionen des Agenten analysieren. Folglich können sie für alle Agententypen eingesetzt werden [Hube22].

Perturbation-Ansätze können mit unterschiedlichem Fokus genutzt und in Single-Feature-Sensitivitätsansätze und Occlusion-Sensitivitätsansätze unterteilt werden. GEVREY ET AL. [Gevr03] haben den Single-Feature-Sensitivitätsansatz basierend auf den Ergebnissen von SCARDI ET AL. [Scar99] entwickelt. Die Methode zielt darauf ab, den Einfluss der Änderung eines einzelnen Input-Features zu quantifizieren, während alle anderen Input-Feature konstant gehalten werden. Dabei wird das Feature, das zu der größten Änderung der Ausgabeschicht führt, als am einflussreichsten für den Entscheidungsprozess eingestuft [Gevr03, Scar99]. Die Veränderung eines Input-Features wird anhand von Gleichung 2-3 bestimmt. Dabei entspricht z einem Vektor, der alle Feature enthält. Der Wert des variierten Features z_i wird um den Faktor e sequentiell reduziert oder erhöht [Gevr03].

$$z_{i,neu} = z_i \cdot (1 \pm e) \quad (2-3)$$

Occlusion-Sensitivitätsansätze generieren eine Saliency Map, indem eine Maskierungsfunktion auf eine Gruppe an Input-Features angewendet wird [Ivan21]. Hierfür wird ein Bereich der Größe $h \times h$ definiert und schrittweise über alle Input-Feature geschoben. Die Werte der Input-Feature in diesem Bereich werden durch einen anderen Wert ersetzt (siehe Abbildung 2-19). Im Fall eines Bildes, das als Eingangsinformation genutzt wird, wird die Farbe jedes Pixels als Input-Feature angesehen. Die Occlusion-Sensitivität kann berechnet werden, indem das unmaskierte Bild durch das KNN evaluiert wird und anschließend das Bild unter Anwendung der Maskierungsfunktion bewertet wird. Die Maskierungsfunktion könnte z. B. die Farbe aller Pixel im Bereich $h \times h$ auf Weiß ändern. Die zusammenfassende Saliency Map wird erzeugt, indem der Bereich der Maskierungsfunktion über das gesamte Bild verschoben wird und die gemessenen Abweichungen normalisiert werden [Zeil13]. Eine Herausforderung bei der Anwendung von Occlusion-Sensitivitätsansätzen ist die Festlegung einer geeigneten Bereichsgröße zur Anwendung der Maskierungsfunktion, da dies die Interpretation der Ergebnisse beeinflusst [Ivan21].

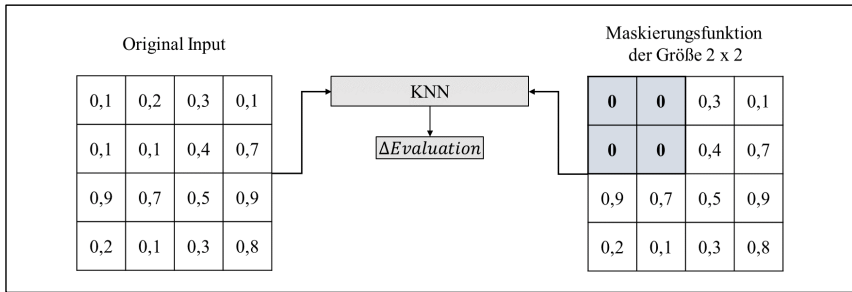


Abbildung 2-19: Anwendung einer Maskierungsfunktion zur Berechnung der Occlusion-Sensitivität

2.4.5 Zusammenfassung der Leistungspotentiale von Reinforcement Learning Ansätzen

Für einen RL-basierten Lösungsansatz können die nachfolgenden Leistungspotentiale identifiziert werden.

RL-Ansätze sind selbstlernende Systeme, die sich durch eine hervorragende Leistungsfähigkeit auszeichnen. Dabei können RL-basierte Ansätze die Leistungsfähigkeit von bestehenden Methoden und von Menschen getroffenen Entscheidungen übertreffen, wie am Beispiel des Spiels Go beobachtet werden konnte [Silv16].

Eine besondere Herausforderung für konventionelle Lösungsansätze stellen dynamisch geprägte Probleme dar, die durch eine hohe kombinatorische Komplexität charakterisiert sind. Deren Lösung benötigt entweder definierte Verhaltensregeln oder eine analytische Problemformulierung durch Zielfunktionen und Nebenbedingungen. Ein regelbasierter Lösungsansatz erscheint nur für eine begrenzte Problemkomplexität zweckmäßig, da das Einbeziehen aller Eventualitäten in einem erhöhten Definitionsaufwand resultiert. Analytische Problemformulierungen für hoch nichtlineare Probleme stellen zudem eine Herausforderung dar, weshalb Annahmen und Vereinfachungen für die Modellierung genutzt werden, aber die Lösungsqualität einer automatisierten Lösungsmethode reduzieren [Mazy21, Hoss18, Bian09].

RL wird bereits erfolgreich bei Problemen mit einer hohen kombinatorischen Komplexität genutzt, die durch dynamische Einflüsse charakterisiert sind. Ein RL-Ansatz benötigt lediglich ein Feedback in Form einer Belohnung, die z. B. das Ergebnis einer Simulation sein kann. Anschließend erlernt der RL-Ansatz eigenständig die linearen und nichtlinearen Zusammenhänge des Problems und kann daraus eine zustandsorientierte Strategie ableiten, die auch auf große Zustandsräume angewendet werden kann. RL ist ein skalierbarer Ansatz, der für einzel- und multikriterielle Probleme genutzt wird [Sutt18, Moss16].

Eine weitere Fähigkeit von RL besteht darin, dass der RL-Agent lediglich einen Belohnungswert als Feedback für seine Aktionswahl erhält und die erlernte Strategie nicht durch etablierte Lösungsstrategien (z. B. Regeln) beeinflusst wird. Hierdurch können neuartige Lösungen entstehen, die zu einer verbesserten Problemlösungsfähigkeit führen [Sutt18, Silv16]. Ferner können Aufgaben so modelliert werden, dass der RL-Ansatz eine allgemeingültige Lösungsstrategie erlernt, die auf unterschiedliche Problemvarianten

übertragbar ist [Wier12]. Im Vergleich zu konventionellen Metaheuristiken muss nicht jedes Mal eine neue Berechnung durchgeführt werden, da der RL-Ansatz keine individuelle Lösung, sondern die grundlegenden Zusammenhänge zur Lösung des Problems erlernt hat.

XRL-Methoden ermöglichen, die Transparenz und Nachvollziehbarkeit des Trainingsprozesses zu steigern und beheben somit Defizite konventioneller Metaheuristiken.

Die aufgezeigten Potentiale wurden in ersten Beiträgen für das Fabriklayoutplanungsproblem genutzt. UNGER ET AL. und SCHNEIDEWIND ET AL. beschreiben in ihrem Beitrag eine theoretische Modellierungsmöglichkeit, um RL für die Fabriklayoutplanung nutzbar zu machen. Die Autoren wenden den Modellierungsansatz jedoch nicht an, nennen keine zu berücksichtigenden Optimierungszielgrößen und verbleiben daher bei einer rein theoretischen Beschreibung des Lösungsansatzes [Schn23, Unge21]. HEINACH ET AL. und IDEKA ET AL. nutzen bereits einen RL-Ansatz zur Lösung eines Fabriklayoutplanungsproblems. Die gewählte Modellierung weist allerdings Defizite auf, da die Problemstellungen nicht als Open Field Layoutplanungsproblem modelliert wurden und nur ein einzelkriterielles Optimierungsproblem mit wenigen Funktionseinheiten ohne Nutzung einer Simulation gelöst wurde [Hein23, Iked22]. Demnach werden durch die Ansätze zentrale Forschungsaspekte, die durch eine multikriterielle und simulationsbasierte Modellierung und Optimierung sowie die Skalierbarkeit des Ansatzes auf eine industriell relevante Problemgröße entstehen, nicht beantwortet, noch werden Aspekte des XRL einbezogen.

Zur Überwindung der bestehenden Defizite und zur Nutzung des Potentials einer RL-basierten Fabriklayoutplanung müssen die nachfolgenden Entwicklungsschritte durchgeführt werden:

- ☐ Modellierung der Fabriklayoutplanungsaufgabe als sequentielles Entscheidungsproblem mit gleichbleibendem Aktionsspielraum für die gesamte Episode
- ☐ Modellierung der Umgebung
- ☐ Definition der Zustandsinformationen
- ☐ Definition der multikriteriellen Belohnungsfunktion
- ☐ Definition einer Handhabungsstrategie für ungültige Aktionen
- ☐ Auswahl einer Agentenarchitektur, der zugehörigen Explorationsstrategie sowie der Hyperparameter
- ☐ Sicherstellung der Skalierbarkeit des Ansatzes
- ☐ Entwicklung geeigneter XRL-Methoden zur Steigerung der Nachvollziehbarkeit der Lösung
- ☐ Validierung der Leistungspotentiale anhand industrieller Anwendungsszenarien

3 Zielsetzung und Aufgabenstellung

In diesem Kapitel werden die zentralen Ausführungen und Schlussfolgerungen aus der Betrachtung des Standes der Wissenschaft und Technik zusammengefasst (Kapitel 3.1) und darauf aufbauend die übergeordnete Zielsetzung, die Forschungsfrage und die Entwicklungsschritte abgeleitet (Kapitel 3.2).

3.1 Erkenntnisse aus dem Stand von Wissenschaft und Technik

- ☐ Die Qualität der Fabriklayoutplanung beeinflusst maßgeblich die Produktivität der zukünftigen Fabrik. Dabei ist die frühe Phase der Layoutplanung von besonderer Relevanz, da in dieser die Freiheitsgrade in der Planung am höchsten sind, fehlerhafte Planungsergebnisse nachträglich nur zu deutlich erhöhten Kosten angepasst werden können und auf unsicherem Wissen aufgebaut werden muss.
- ☐ In der Fabriklayoutplanung werden meist mehrere Zielgrößen parallel verfolgt, wie z. B. die Minimierung der Durchlaufzeit bei einer bestmöglichen Medienver- und -entsorgung. Zielgrößen können in teilweise konfliktären Zusammenhängen zu einander stehen. Die Lösung solcher Zielkonflikte stellt eine zentrale Herausforderung der Fabriklayoutplanung dar.
- ☐ Die Eigenschaften eines Produktionssystems und die zugehörigen Materialflusseigenschaften können nur basierend auf Annahmen und Vereinfachungen durch rein analytische Gleichungen beschrieben werden. Als Resultat sind bei einer analytischen Bewertung und Optimierung des Fabriklayouts Unterschiede zwischen der erwarteten und tatsächlichen Leistungsfähigkeit des Produktionssystems wahrscheinlich. Durch die Nutzung von Materialflusssimulationen können diese Differenzen reduziert und weitere Zielgrößen berücksichtigt werden, die bei einer rein analytischen Beschreibung nicht abgebildet werden können.
- ☐ Die manuelle Fabriklayoutplanung ist meist zeitintensiv und beruht auf dem Expertenwissen der Planenden. Automatisierte Planungsansätze (vorrangig Metaheuristiken) können als Unterstützungswerkzeug in der frühen Phase der Fabriklayoutplanung eingesetzt werden, um den zeitlichen Aufwand zur Erstellung von Layoutvarianten zu reduzieren. Hierfür wird das Fabrikplanungsproblem als mathematisches Optimierungsproblem modelliert, das eine hohe kombinatorische Komplexität aufweist. Bestehende Ansätze zur Lösung dieses Problems weisen Defizite auf, da überwiegend analytische Problemmodellierungsansätze genutzt werden, das Planungsproblem nicht realitätsnah modelliert wird oder ausschließlich die Neu- oder Umplanung bestehender Fabriken betrachtet wird. Konventionelle Metaheuristiken können zwar zur Lösung industrieller Problemstellungen genutzt werden, der Lösungsprozess ist jedoch nicht nachvollziehbar oder reproduzierbar.
- ☐ RL zielt auf die Lösung von sequentiellen Entscheidungsproblemen ab und hat in anderen Anwendungsbereichen bereits bewiesen, eine leistungsfähige Alternative zu bestehenden Problemlösungstechniken zu sein. RL wurde erfolgreich für Problemstellungen mit einer ausgeprägten dynamischen und kombinatorischen Komplexität eingesetzt. Hierbei können Simulationen in die Umgebung des Ansatzes integriert werden. Ein Vorteil von RL gegenüber konventionellen Lösungsverfahren besteht darin, dass das Systemverhalten nicht vollständig analytisch beschrieben werden muss, sondern dass eine Strategie ausschließlich auf der Basis eines Belohnungsfeedbacks erlernt werden kann.

- RL-Ansätze erlernen die Zusammenhänge, die zum Lösen einer Aufgabe notwendig sind. Hierdurch können neue Problemlösungsstrategien entwickelt werden, ohne dass Expertenwissen eingebracht werden muss. Die erlernte Strategie ist im Vergleich zu konventionell eingesetzten Metaheuristiken als Teil des Agenten gespeichert und kann erneut abgerufen werden. So kann nicht nur erlerntes Wissen auf andere Problemstellungen übertragen, sondern kann auch die Nachvollziehbarkeit der generierten Lösung gesteigert werden, wenn Ansätze des XRL zum Einsatz kommen.
- Bestehende Ansätze, die RL zur Generierung von Fabriklayoutvarianten nutzen, liegen überwiegend in Form von theoretischen Überlegungen vor, ohne dass sie für die Lösung eines Fabriklayoutplanungsproblems genutzt wurden. Die vereinzelt funktionsfähigen Ansätze beziehen sich lediglich auf einzelkriterielle, analytisch formulierte Problemstellungen, deren Komplexität und Realitätsnähe signifikante Schwächen aufweisen.

Die Fabriklayoutplanung besitzt Bedarfe, die in bestehenden Ansätzen keine oder unzureichende Berücksichtigung finden. Demgegenüber stehen die Leistungspotentiale von RL, die mit den Bedarfen in vielen Bereichen übereinstimmen, weshalb der erfolgreiche Einsatz von RL bestehende Schwächen in der automatisierten Layoutplanung überwinden kann.

3.2 Forschungsziele und Aufbau der Arbeit

Die Zielstellung dieser Dissertation besteht in einer Qualitätserhöhung der frühen Fabriklayoutplanungsphase durch die kombinierte Nutzung von RL, EDS-Ergebnissen und XRL-Ansätzen. Hierfür wird folgende Forschungsfrage untersucht: Wie muss ein RL-Ansatz, der als Unterstützungswerkzeug eingesetzt werden kann und eine multikriterielle Neu- und Umanplanung von Fabriklayouts unter Berücksichtigung von dynamischen Eigenschaften des Materialflusses ermöglicht, gestaltet sein und eignet sich dieser zur Lösung industrieller Planungsprobleme? Die Erreichung der Zielsetzung und die Beantwortung der Forschungsfrage wird durch sequentielle durchgeführte Entwicklungsschritte sichergestellt, die visuell in Abbildung 3-1 zusammengefasst sind.

Den Ausgangspunkt für den ersten Entwicklungsschritt, der Modellierung des Fabriklayoutplanungsproblems als RL-kompatibles sequentielles Entscheidungsproblem, stellen die in Kapitel 4.1 definierten Systemgrenzen und Annahmen in Form von notwendigen Eingangsinformationen und erwarteten Ergebnissen dar. Darauf aufbauend werden in Kapitel 4.2 der Aktionsspielraum und in Kapitel 4.3 die Zustandsinformationen des RL-Agenten modelliert. Anschließend erfolgt eine Verknüpfung der zuvor modellierten Elemente in Kapitel 4.4, in dem der RL-Agent finalisiert wird. Hierbei werden u. a. die zur Informationsverarbeitung genutzten KNN-Architekturen beschrieben.

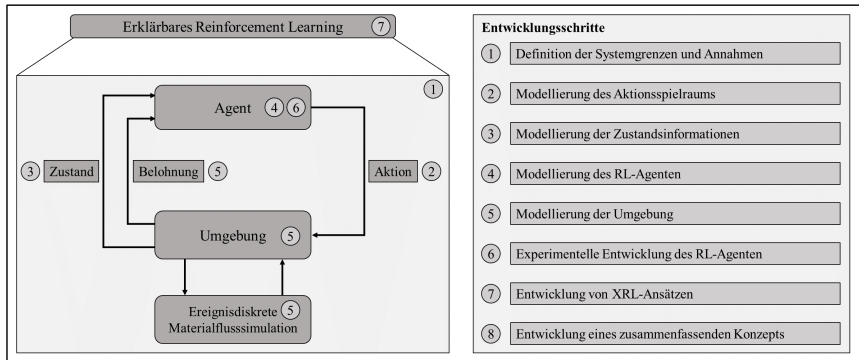


Abbildung 3-1: Überblick über die Entwicklungsschritte zur Reinforcement Learning basierten Fabriklayoutplanung

Für die Bearbeitung des nächsten Entwicklungsschritts wird die Modellierung der Umgebung durch die Definition der Belohnungsfunktion vervollständigt. Der RL-Ansatz soll konventionelle analytische Zielgrößen und Ergebnisse einer EDS umfassen. Daher werden in Kapitel 4.5.1 analytische Zielgrößen zur Optimierung definiert und in den Ansatz integriert. Außerdem wird in Kapitel 4.5.2 ein EDS-Modul entwickelt, das die Umgebung des RL-Ansatzes erweitert. Dieses umfasst die zentralen Fertigungs- und Materialflusselemente und ermöglicht die Simulation unterschiedlicher Produktionssysteme. Außerdem werden die zugehörigen Schnittstellen entwickelt, sodass die EDS eigenständig im Trainingsprozess des RL-Agenten initialisiert und ein Simulationslauf durchgeführt werden kann. Die Ergebnisse sind unterschiedliche Zielgrößen des Materialflusses, die zusammen mit den analytisch formulierten Zielgrößen in eine Belohnungsfunktion überführt werden.

In einem weiteren Schritt wird die Skalierbarkeit des entwickelten Ansatzes hinsichtlich der Problemgröße und -komplexität sichergestellt (Kapitel 5). Hierfür werden Layoutplanungsprobleme mit zunehmender Größe definiert, sodass die Lösungsfähigkeit anhand repräsentativer Problemstellungen untersucht werden kann. Da die Leistungsfähigkeit verschiedener Agenten im Vorfeld nicht eingeschätzt werden kann, werden unterschiedliche Agententypen miteinander verglichen und der geeignetste Ansatz ausgewählt. In Analogie hierzu wird die Leistungsfähigkeit der zwei vorgestellten RL-Strukturen zur multikriteriellen Optimierung untersucht (siehe Kapitel 2.4.2). Somit existiert nach erfolgreicher Bearbeitung des 6. Entwicklungsschritts ein funktionsfähiger RL-Ansatz zur multikriteriellen und simulationsbasierten Fabriklayoutneu- und -umplanung.

Während des 7. Entwicklungsschritts wird die Erklärbarkeit des entwickelten Ansatzes untersucht. Hierfür werden basierend auf den zu definierenden Erklärbarkeitszielen zugehörige Methoden entwickelt und anhand eines simplifizierten Anwendungsbeispiels erklärt (Kapitel 6). Abschließend wird ein ganzheitliches Konzept entwickelt, das alle Informationsflüsse, Initialisierungsschritte und Abhängigkeiten darstellt. Zur Steigerung der Zugänglichkeit des Ansatzes wird ein Softwaredemonstrator entwickelt, der das Gesamtkonzept digital abbildet und einen potentiellen Nutzer durch die einzelnen Phasen des Ansatzes führt (Kapitel 7).

Abbildung 3-2 stellt die Kapitel der Dissertation zusammenfassend dar.

Kapitel 4: Modellierung des Fabriklayoutplanungsproblems	
4.1: Definition der Systemgrenzen und zugehöriger Annahmen	
4.2: Modellierung des Aktionsspielraums	
4.3: Modellierung der Zustandsinformationen	
4.4: Modellierung des Reinforcement Learning Agenten	
4.5: Integration der Optimierungszielgrößen und Modellierung der Belohnungsfunktion	
4.5.1: Integration analytisch formulierter Zielgrößen	4.5.2: Integration von EDS-Ergebnissen
Kapitel 5: Experimentelle Entwicklung des Reinforcement Learning Agent	
5.1: Umgang mit ungültigen Aktionen	
5.2: Festlegung des Reinforcement Learning Agent Typs	
5.3: Festlegung der Agentenstruktur zur multikriteriellen Optimierung	
Kapitel 6: Erklärbares Reinforcement Learning für die Fabriklayoutplanung	
6.1: Definition der Zielgrößen	
6.2: Entwicklung der XRL-Ansätze	
6.2.1: Policy Summarization	6.2.2: Occlusion-Sensitivitätsansätze
Kapitel 7: Zusammenfassendes Konzept zur Reinforcement Learning basierten Fabriklayoutplanung	
Kapitel 8: Validierung	
8.1: Umplanungsszenario	
5.2: Neuplanungsszenario	
5.3: Diskussion und Schlussbetrachtung	
Kapitel 9: Zusammenfassung und Ausblick	

Abbildung 3-2: Struktur der verbleibenden Kapitel dieser Dissertation

4 Modellierung des Fabriklayoutplanungsproblems

Der Ausgangspunkt für die Lösung eines Optimierungsproblems mittels RL stellt eine Modellierung als sequentielles Entscheidungsproblem dar. Zudem wird bei der Entwicklung von RL-Ansätzen angenommen, dass Markov-Eigenschaften vorliegen. Daher sollten zu jedem Zeitpunkt alle notwendigen Zustandsinformationen zur Entscheidungsfindung verfügbar sein, ohne dass vorangegangene Zustände und Entscheidungen berücksichtigt werden müssen. Weiterhin müssen Optimierungsziele definiert und quantifiziert werden, sodass diese in eine Belohnungsfunktion überführt werden können.¹

4.1 Definition der Systemgrenzen und zugehöriger Annahmen

Der Ausgangspunkt zur Modellierung des Fabriklayoutplanungsproblems als sequentielles Entscheidungsproblem sind die Systemgrenzen und zugehörigen Annahmen. Da die Fabriklayoutplanung ein zentrales Element der Fabrikplanung darstellt und in einem Abhängigkeitsverhältnis zu anderen Planungsdisziplinen steht, muss zu Beginn festgelegt werden, an welchem Punkt der Fabrikplanung der zu entwickelnde Ansatz beginnen und enden soll. Der Ausgangspunkt wird durch die abgeschlossene Betriebsmittelplanung und Dimensionierung der Funktionseinheiten definiert. Somit ist aus der Betriebsmittelplanung bekannt, wie viele und welche Funktionseinheiten im zu planenden Fabriklayout benötigt werden. Zur Bestimmung der Funktionseinheiten verwendet die Betriebsmittelplanung Informationen über die Prozesskette und die prognostizierte Menge der zu produzierenden Produkte. Folglich wird davon ausgegangen, dass alle Prozessketten und die zu produzierende Menge für jedes Produkt vorliegen. Zudem wird angenommen, dass alle funktionsbezogenen Eigenschaften der Funktionseinheiten (z. B. Prozess- und Rüstzeiten) bekannt sind. Im Rahmen der Dimensionierungsplanung wird die Dimension der zu platzierenden Funktionseinheit unter Berücksichtigung von Reserveflächen bestimmt.

Außerdem wird angenommen, dass die Eigenschaften der Fabrikhalle im Umplanungsfall bekannt sind. Hierzu zählen neben der Dimension der Fabrikhalle auch Informationen zur Verfügbarkeit von Medien und Bodentraglasten sowie belegte Flächen, die nicht durch andere Funktionseinheiten belegt werden dürfen.

Daneben wird davon ausgegangen, dass bereits ein Materialflusskonzept vorliegt, das die angestrebte Materialflusssteuerungsstrategie, die zur Verfügung stehenden Fördermittel und deren Eigenschaften umfasst. Abschließend sollte die Reihenfolgeplanungsstrategie festgelegt sein, die bestimmt, in welcher Reihenfolge die Fertigungsaufträge auf welchen Maschinen gefertigt werden.

Der RL-Ansatz wird in der frühen Phase der Layoutplanung eingesetzt. Dieser soll die Generierungs- und Optimierungsaufgabe von Blocklayoutvarianten automatisiert mit einer hohen Planungsqualität durchführen. Demnach ist die abschließende Systemgrenze durch den manuellen Detailplanungsschritt definiert, in der die bewerteten Blocklayoutvarianten aufgegriffen und in ein Feinlayout überführt werden.

¹ Teile der nachfolgenden Ergebnisse wurden bereits in Journal- [Klar24b, Klar24c, Klar23a, Klar24d, Klar21] und Konferenzbeiträgen [Klar24a, Klar23c, Klar23b, Klar22a, Klar22b] publiziert.

4.2 Modellierung des Aktionsspielraums

Die Entscheidungsgröße im zu lösenden Fabriklayoutplanungsproblem entspricht der Positionierung einer Funktionseinheit im Layout. Da RL zur Lösung sequentieller Entscheidungsprobleme genutzt wird, werden die Funktionseinheiten schrittweise platziert. Hierfür wird das zu planende Layout als ein Raster mit $n = n_x \cdot n_y$ Positionen unterteilt (siehe Abbildung 4-1). Jede Position im Layout wird als Elementarzelle aufgefasst und kann zusätzliche Eigenschaften wie die Höhe, Medienverfügbarkeit und Bodentraglasten besitzen. Der Status einer Elementarzelle beschreibt, ob diese frei, durch eine Funktionseinheit oder Wand belegt oder für den Materialtransport reserviert ist. Diese Modellierungsform erlaubt es, Neu- und Umanplanungsprobleme zu modellieren. Im Fall einer reinen Neuplanung kann das Raster als vollständig leer, ohne Berücksichtigung von Höhen, Medienverfügbarkeiten und Bodentraglasten angenommen werden. Im Umanplanungsfall müssen hingegen bestehende Restriktionen berücksichtigt werden. Durch die Festlegung der Elementarzelleneigenschaften kann das bestehende Layout und zugehörige Restriktionen berücksichtigt werden. Die Funktionseinheiten werden ebenfalls mittels Elementarzellen beschrieben und können durch die Kombination mehrerer Elementarzellen unterschiedliche Größen annehmen.

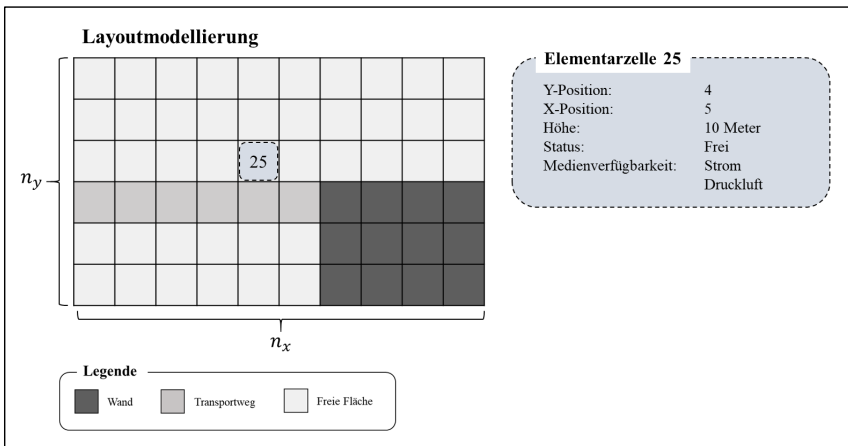


Abbildung 4-1: Modellierung des Layouts mittels Elementarzellen

Die Aktionswahl des Agenten bestimmt die Position der linken unteren Ecke einer Funktionseinheit und deren Rotation. Die linke untere Ecke der Funktionseinheit dient folglich als Referenzpunkt für die Aktion und stellt ebenfalls den Materialaufnahme- und -abgabepunkt dar. Da die Funktionseinheiten sequentiell platziert werden, wird die Platzierungsreihenfolge im Vorfeld festgelegt. Diese kann analog zum Dreieckprobiervorgehen entsprechend der Transportintensitäten oder anhand der tatsächlichen Produktionsprozessreihenfolge bestimmt werden.

Der daraus resultierende Aktionsspielraum kann als einzeldiskret, multidiskret oder multikontinuierlich modelliert werden. Die Struktur der Ausgangsschicht eines KNNs ist für alle drei Modellierungsformen in Abbildung 4-2 visualisiert. Unabhängig von der Modellierungsform wird eine Aktion A_t durch ein Tupel, bestehend aus der x-Position $A_{t,x}$, y-Position $A_{t,y}$ und Rotation $A_{t,\varphi}$ der Funktionseinheit, beschrieben. Insgesamt stehen vier

Rotationsmöglichkeiten mit je zusätzlichem 90 ° Rotationswinkel zur Verfügung. Die Art der genutzten Modellierung muss auf den jeweiligen Agententyp abgestimmt sein (siehe Kapitel 2.4.3), da nicht jeder Agent diskrete und kontinuierliche Aktionen auswählen kann.

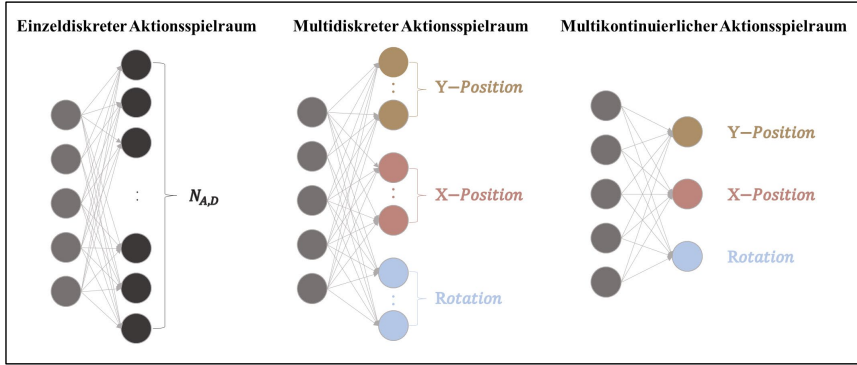


Abbildung 4-2: Modellierung des Aktionsspielraums durch drei unterschiedliche KNN-Strukturen

Die *einzeldiskrete Aktionsmodellierung* $A_{t,D}$ umfasst $N_{A,D} = n_x \cdot n_y \cdot 4$ Aktionsmöglichkeiten und wird durch Gleichung 4-1 dargestellt.

$$A_{t,D} = \begin{pmatrix} \text{Position 1; Rotation } 0^\circ \\ \text{Position 1; Rotation } 90^\circ \\ \vdots \\ \text{Position } n; \text{Rotation } 180^\circ \\ \text{Position } n; \text{Rotation } 270^\circ \end{pmatrix} \quad (4-1)$$

Im Fall einer *multidiskreten Aktionsmodellierung* $A_{t,MD}$ werden die x-Position, y-Position und Rotation entsprechend Gleichung 4-2 bis 4-4 separiert voneinander ausgewählt und anschließend aggregiert. Hierdurch wird der Aktionsspielraum auf $N_{A,MD} = n_x + n_y + 4$ reduziert.

$$A_{t,x} = \begin{pmatrix} \text{Position 1} \\ \vdots \\ \text{Position } n_x \end{pmatrix} \quad (4-2)$$

$$A_{t,y} = \begin{pmatrix} \text{Position 1} \\ \vdots \\ \text{Position } n_y \end{pmatrix} \quad (4-3)$$

$$A_{t,\varphi} = \begin{pmatrix} 0^\circ \\ 90^\circ \\ 180^\circ \\ 270^\circ \end{pmatrix} \quad (4-4)$$

Die *multikontinuierliche Aktionsmodellierung* ähnelt dem multidiskreten Fall, da die x-Position, y-Position und Rotation separiert voneinander ausgewählt werden. Allerdings wird der Aktionsspielraum als kontinuierlich zwischen -1 und 1 angenommen (siehe Gleichung 4-5). Die ausgewählten Werte für jede Dimension werden anschließend dekodiert und einer diskreten Position im Layout und Rotation der Funktionseinheit zugeordnet.

$$A_{t,x}; A_{t,y}; A_{t,\varphi} \in [-1, 1] \quad (4-5)$$

4.3 Modellierung der Zustandsinformationen

Der Agent wählt eine Aktion aus, die anschließend von der Umgebung umgesetzt wird und zu einer Änderung der aktuellen Zustandsinformationen führt. Diese stellen den neuen Ausgangspunkt für die Aktionswahl dar. Demnach muss der Zustand alle notwendigen Informationen beinhalten, um eine belohnungsmaximierende Entscheidung zu ermöglichen. Für die Fabriklayoutplanung benötigt der RL-Agent Informationen über die Prozesskette und die zu produzierenden Mengen, über den aktuellen Zustand des Layouts und über die Funktionseinheit, die durch die gewählte Aktion platziert wird (siehe Abbildung 4-3).

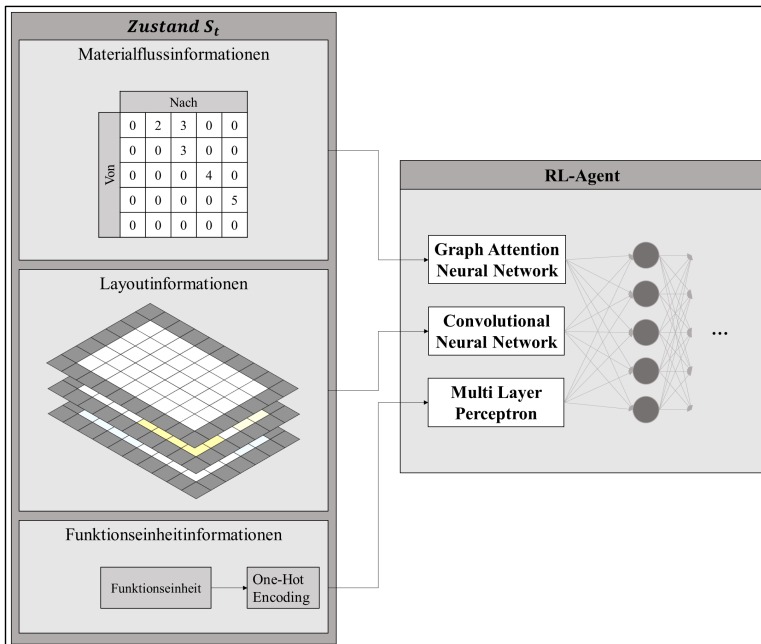


Abbildung 4-3: Zustandsinformationen und deren Verarbeitung durch den Agenten

Die Informationen über die Prozesskette und die zu produzierende Menge werden in Form einer Transportmatrix bereitgestellt. Diese fasst die Produktmengen, die zwischen allen Funktionseinheiten transportiert werden müssen, zusammen und bildet die Prozessketten und die Materialflussbeziehungen ab. Der aktuelle Zustand des Layouts wird durch den Zustand der

einzelnen Elementarzellen definiert. Die Platzierung einer Funktionseinheit kann den Zustand mehrerer Elementarzellen von „frei“ auf „belegt“ ändern. Daher wird das Layout als eine Matrix aufgefasst, in der jedes Matricelement einer Elementarzelle entspricht und deren Zustandsinformationen bündelt. Der Zustand des Layouts ist jedoch nicht ausschließlich durch freie oder belegte Elementarzellen beschreibbar, da zusätzliche Attribute wie die verfügbaren Medien, einzuhaltende Bodenlasten oder Höheninformationen berücksichtigt werden können. Daher besitzt jedes Matricelement mehrere Dimensionen, wobei jede Dimension einem Attribut der Elementarzelle entspricht. Abschließend werden die Informationen zur platzierenden Funktionseinheit durch einen One-Hot enkodierten Tensor bereitgestellt.

4.4 Modellierung des Reinforcement Learning Agenten

Die Modellierung des Agenten baut auf den Zustandsinformationen und dem Aktionsspielraum auf und zielt darauf ab, die Zustandsinformationen zielgerichtet zu verarbeiten und in den gewünschten Aktionsspielraum zu überführen. Daher werden drei verschiedene Architekturen verwendet, um je einen Bestandteil der Zustandsinformationen zu verarbeiten (siehe Abbildung 4-3). Für die Berücksichtigung der Materialflussbeziehungen wird ein GANN verwendet, da der Materialfluss durch die Transportmatrix implizit als Graph abgebildet wird. Die Knoten entsprechen dabei den Funktionseinheiten und die Transportbeziehungen definieren die Kanten und die zugehörige Gewichtung. Zur Nutzung des GANNs muss die Verankerung (engl. Embedding) definiert werden. Die initiale Verankerung wird durch eine einzelne Schicht an Neuronen generiert (Dense-Layer). Alle Funktionseinheiten mit einer Materialflussverbindung werden durch Kanten verbunden. Aufbauend auf dem initialen Dense-Layer können paarweise Aufmerksamkeitsbewertungen berechnet werden. Diese bilden den Einfluss der verbundenen Knoten ab und spiegeln daher die Abhängigkeit zwischen den Funktionseinheiten wider. Abschließend wird das Ergebnis dieser Aggregation zur ursprünglichen Knotendarstellung hinzugefügt und durch einen weiteren Dense-Layer in die endgültige Verankerung überführt. Die layoutbezogenen Informationen werden mit einem CNN verarbeitet, da die Nachbarschaftsbeziehungen im Layout und der Zustand einzelner Bereiche von besonderem Interesse sind. Bei der Betrachtung von Farbinformationen in Bildern wird bei einem CNN häufig ein separater Channel für jede Farbinformation (Rot, Grün und Blau) genutzt [O'Sh15]. Dieses Prinzip kann für die Eigenschaften der Elementarzellen adaptiert werden. Daher wird jede Informationsdimension (Informationen über freie und belegte Flächen, Informationen über verfügbare Medien, Höheninformationen und Informationen über Bodenlasten) durch einen eigenen Channel berücksichtigt. Die Zustandsinformationen über die zu platzierende Funktionseinheit werden durch ein MLP verarbeitet.

Anschließend werden alle Architekturen zusammengeführt und durch weitere versteckte Schichten bis zur Ausgabeschicht überführt. Die Hyperparameter dieser Schichten werden in Kapitel 5 definiert, da diese einen entscheidenden Einfluss auf die Leistungsfähigkeit des Ansatzes haben.

Abschließend muss festgelegt werden, mit welcher Strategie die Auswahl ungültiger Aktionen verhindert wird. Wie in Kapitel 2.4.2 beschrieben, können ungültige Aktionen bestraft werden oder Maskierungsfunktionen zur Prävention ungültiger Aktionen genutzt werden. Die Maskierungsfunktion prüft, ob eine gewählte Aktion zu einem ungültigen Layout führt. Ein ungültiges Layout entsteht, wenn Funktionseinheiten auf belegten Flächen platziert werden

oder wenn eine Funktionseinheit durch die gewählte Aktion nicht mehr durch Transportmittel erreicht werden kann (siehe Abbildung 4-4). Im Fall einer ungültigen Aktionswahl wird diese an die Umgebung transferiert und umgesetzt. Andernfalls wird die gewählte Aktion abgelehnt und die nächstbessere Aktion wird durch die Maskierungsfunktion geprüft. Entscheidend für die Wahl einer Präventionsform ist die daraus resultierende Leistungsfähigkeit des Ansatzes. Daher werden die verschiedenen Ansätze zur Prävention ungültiger Aktionen in Kapitel 5.1 miteinander verglichen.

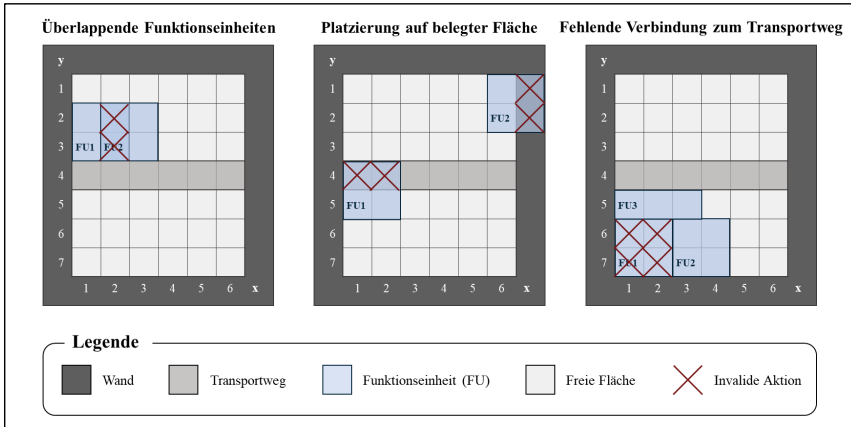


Abbildung 4-4: Beispielhafte Darstellung ungültiger Aktionen

4.5 Integration der Optimierungszielgrößen und Modellierung der Belohnungsfunktion

Im nachfolgenden Kapitel werden die Optimierungszielgrößen definiert und in eine Belohnungsfunktion überführt. Tabelle 4-1 fasst die berücksichtigten Optimierungszielgrößen zusammen. Diese können anhand des Anwendungsfalls (Neu- oder Umplanung), des Ziels (Minimierung oder Maximierung), anhand der Art der Zielgröße (analytisch oder dynamisch) und des Bewertungstyps (funktionseinheitbezogen oder layoutbezogen) charakterisiert werden.

Analytisch formulierte Zielgrößen können ohne die Verwendung einer EDS bestimmt werden (Kapitel 4.5.1), während dynamisch geprägte Zielgrößen das Ergebnis der EDS darstellen (Kapitel 4.5.2). Bei einer layoutbezogenen Bewertung wird das gesamte Layout bewertet. Folglich ist der Belohnungswert, den der Agent erhält, für alle platzierten Funktionseinheiten derselbe. Bei einer funktionseinheitbezogenen Bewertung wird für jede Transition ein eigener Belohnungswert errechnet, der sich nur auf den gerade ausgeführten Schritt bezieht. Um sowohl eine schritt- als auch layoutbezogene Bewertung zu ermöglichen, wird die Bewertung des Layouts erst nach dessen Fertigstellung durchgeführt. Anschließend werden die ermittelten Zielgrößen wieder in eine funktionseinheitbezogene Bewertung überführt und für das Training des Agenten genutzt (siehe Abbildung 4-5). Hierfür werden alle Zielgrößen mittels Linearkombination in eine funktionseinheitbezogene Belohnungsfunktion überführt (siehe Gleichung 4-6 und 4-7). Die zugehörigen Gewichtungen w_i dienen dazu, einzelne Zielgrößen zu priorisieren oder nicht in die Optimierung einzubeziehen ($w_i = 0$). Die einzelnen Zielgrößen

werden so modelliert, dass sie einen Wertebereich von -1 bis 0 annehmen. Dabei entspricht 0 dem bestmöglichen Wert und -1 dem schlechtesten Ergebnis.

$$R(t) = w_1 \cdot R_{DZ}(t) + w_2 \cdot R_{AF}(t) + w_3 \cdot R_{AT} + w_4 \cdot R_S + w_5 \cdot R_{TI}(t) + w_6 \cdot R_M(t) + w_7 \cdot R_B(t) + w_8 \cdot R_{UM}(t) + w_9 \cdot R_A(t) \quad (4-6)$$

$$\sum_{i=1}^9 w_i = 1 \quad (4-7)$$

Tabelle 4-1: Klassifizierung der Optimierungskenngrößen

Zielgröße	Art der Zielgröße		Ziel		Planungsfall		Bewertungstyp	
	Analytisch	Dynamisch	Min	Max	Neu-planung	Um-planung	Funktions-einheit	Layout
Durchlaufzeit (R_{DZ})		X	X		X	X	X	X
Auslastung der Funktionseinheiten (R_{AF})		X		X	X	X	X	
Auslastung der Transportmittel (R_{AT})		X		X	X	X		X
Stau im Materialfluss (R_S)		X	X		X	X		X
Transportintensität (R_{TI})	X		X		X	X	X	
Medienverfügbarkeit (R_M)	X		-	-	X	X	X	
Bodentraglast (R_B)	X		-	-	X	X	X	
Übersichtlichkeit des Materialflusses (R_{UM})	X			X	X	X	X	
Ähnlichkeit (R_A)	X			X		X	X	

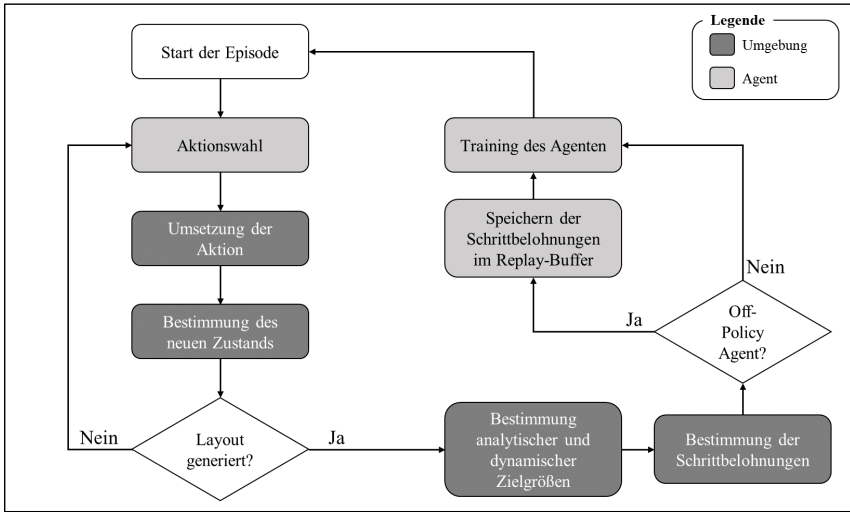


Abbildung 4-5: Episodenablauf zur Integration unterschiedlicher Bewertungstypen in die Belohnungsfunktion

4.5.1 Integration analytisch formulierter Zielgrößen

Die **Transportintensität** (R_{TI}) ist eine der am häufigsten verwendeten Zielgrößen in der Fabriklayoutplanung und stellt einen Spezialfall der Transportkosten mit konstantem Kostensatz pro Transport dar. Die Transportintensität kann mit Hilfe der Gleichungen 4-8 und 4-9 berechnet werden und ist als funktionseinheitbezogene Zielgröße kategorisiert. Dementsprechend ändert sich der Belohnungswert R_{TI} in Abhängigkeit der im Schritt t platzierten Funktionseinheit m_t . Zur Berechnung von R_{TI} wird die aggregierte Transportintensität d_{m_t} durch eine optimistisch geschätzte maximale Transportintensität $d_{m_t, \max}$ dividiert. Zur Bestimmung von d_{m_t} werden alle eingehenden und ausgehenden Transportmengen $a_{m_t, j}$ mit den zugehörigen minimalen Distanzen $d_{m_t, j}^*$, die mit einem Dijkstra-Algorithmus berechnet werden, multipliziert und aufsummiert. Dabei gibt N_{FU} die Anzahl der Funktionseinheiten an.

$$R_{TI}(t) = - \frac{d_{m_t}}{d_{m_t, \max}} \quad (4-8)$$

$$d_{m_t} = \sum_{j=1}^{N_{FU}} d_{m_t, j}^* \cdot a_{m_t, j} \quad (4-9)$$

Abbildung 4-6 zeigt die Berechnung von R_{TI} für $t = 2$ (Funktionseinheit 2) anhand eines Beispiels mit drei Funktionseinheiten ($N_{FU} = 3$). Zu Beginn wird der kürzeste Weg zwischen den Aufnahme- und Abgabepunkten der Funktionseinheiten bestimmt. Als Messgröße werden

die Elementarzellen verwendet. So beträgt die Entfernung zwischen den Funktionseinheiten 1 und 2 eine Elementarzelle und die Entfernung zwischen den Funktionseinheit 2 und 3 vier Elementarzellen. Unter Berücksichtigung der Anzahl der Materialtransporte, die der Transportmatrix entnommen werden können, ergibt sich eine absolute Transportintensität von $1 \cdot 20 + 4 \cdot 10 = 60$. Die maximale Transportintensität kann ermittelt werden, indem die x- und y-Dimension des Layouts addiert wird ($5 + 10$) und mit der Gesamtzahl an Transporten multipliziert wird. Das Ziel der Optimierung der Transportintensität ist es also, Funktionseinheiten, die stark miteinander vernetzt sind, möglichst nahe beieinander zu positionieren. Im Falle eines Positionierungskonfliktes zwischen zwei Funktionseinheiten ist daher die stärker vernetzte Funktionseinheit zu bevorzugen.

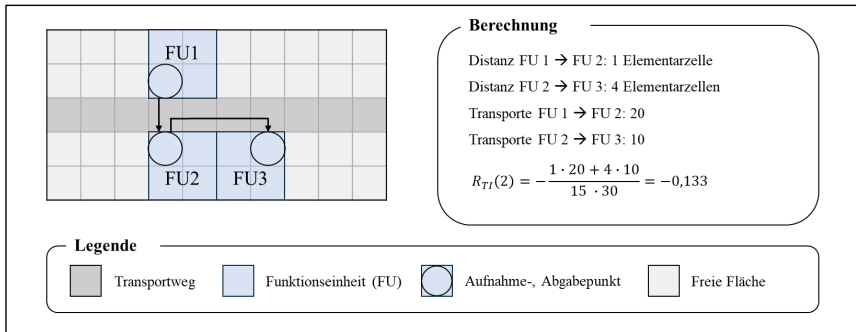


Abbildung 4-6: Berechnung von R_{Tl} für Funktionseinheit 2

Die **Übersichtlichkeit des Materialflusses** ($R_{ÜM}$) kann mit Hilfe der Gleichungen 4-10 bis 4-13 berechnet werden. Der optimale Wert wird erreicht, wenn alle durch den Materialfluss verbundenen Funktionseinheiten in einer Linie angeordnet sind, sodass das Material geordnet in eine Richtung fließt. Zur Berechnung wird nach dem Platzieren der Funktionseinheit m_t die Differenz zwischen den Annahme- und Abgabepunkten in x- und y-Richtung (Δx_{m_t} bzw. Δy_{m_t}) zu den bereits platzierten Funktionseinheiten ($N_{FU_{\text{platziert}}}$) berechnet. Dabei gibt die binäre Größe $z_{m_t,j}$ an, ob die Funktionseinheit m_t und j in einer Materialflussbeziehung stehen. Falls die Funktionseinheit j kein Vorgänger oder Nachfolger der Funktionseinheit m_t ist, gilt $z_{m_t,j} = 0$. Anschließend wird der kleinere Wert von Δx_{m_t} oder Δy_{m_t} durch einen geschätzten maximalen Differenzwert $\Delta_{\max,m_t}$ dividiert. Abbildung 4-7 zeigt ein Beispiel für die Berechnung. Im Fall von Funktionseinheit 2 wird nur die bereits platzierte Funktionseinheit 1 berücksichtigt. Die Berechnung der Differenzen in x- und y-Richtung ergibt Werte von 0 bzw. 1. Die maximale Differenz $\Delta_{\max,m_t}$ entspricht der y-Dimension des Layouts. Es ergibt sich ein optimaler Belohnungswert von Null, da alle Funktionseinheiten in einer Reihe angeordnet wurden. Für die 3. Funktionseinheit verschlechtert sich der Belohnungswert auf $-0,1$. Für die Berechnung werden alle bereits platzierten Vorgänger von Funktionseinheit 3 betrachtet und die Differenzen in x- und y-Dimension berechnet. Die maximale Differenz wird auf 10 ($2 \cdot 5$) erhöht.

$$R_{ÜM}(t) = -\frac{\min(\Delta x_{m_t}, \Delta y_{m_t})}{\Delta_{\max,m_t}} \quad (4-10)$$

$$\Delta x_{m_t} = \sum_{j=1}^{N_{FU_{\text{platziert}}}} |(\text{pos}_{x,m_t} - \text{pos}_{x,j})| \cdot z_{m_t,j} \quad (4-11)$$

$$\Delta y_{m_t} = \sum_{j=1}^{N_{FU_{\text{platziert}}}} |(\text{pos}_{y,m_t} - \text{pos}_{y,j})| \cdot z_{m_t,j} \quad (4-12)$$

$$z_{m_t,j} = \begin{cases} 1; & \text{falls Funktionseinheit } j \text{ und } m_t \text{ eine Verbindung aufweisen} \\ 0; & \text{sonst} \end{cases} \quad (4-13)$$

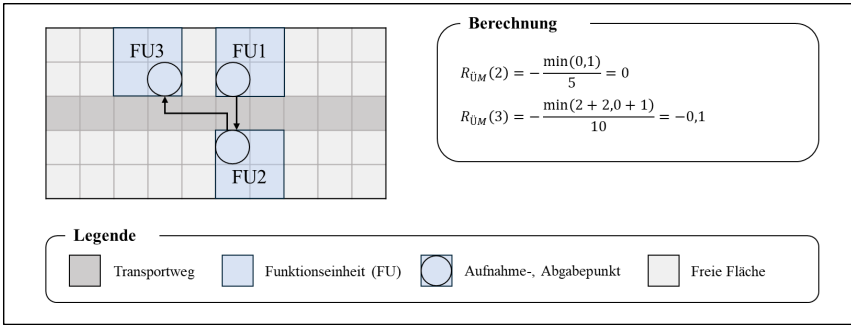


Abbildung 4-7: Berechnung von R_{UM} für Funktionseinheiten 2 und 3

Die Belohnungswerte für die Zielgrößen **Medienverfügbarkeit** (R_M) und **Bodentraglast** (R_B) werden in vergleichbarer Weise berechnet. Ziel ist es, dass vorhandene Medienanschlüsse wie Strom oder Druckluft genutzt werden, wenn sie von einer Funktionseinheit benötigt werden. Dadurch sollen zusätzliche Kosten für ein Nachrüsten vermieden werden.

Für die Berechnung des Belohnungswerts wird die Größe der platzierten Funktionseinheit $N_{Größe,m_t}$ berücksichtigt. Für jede Position wird ein zugehöriger Belohnungswert $r_{pos,i}$ berechnet. Für die Medienverfügbarkeit wird für jede Position geprüft, ob die Elementarzelle das geforderte Medium bereitstellen kann. Wenn die Anforderung erfüllt ist, wird eine Belohnung von 0 zurückgegeben. Wenn die Anforderung nicht erfüllt ist, wird der Nachrüstaufwand berücksichtigt. Ist der Aufwand gering (z. B. Anschluss an den Wasserkreislauf durch Verlegung einer neuen Leitung) ergibt sich eine Belohnung von $-0,5$, andernfalls (z. B. Beschaffung und Installation einer neuen Anlage zur Druckluftversorgung) eine Belohnung von -1 . Es können Strom, Druckluft, Öl und Kühlwasser berücksichtigt werden und die Integration weiterer Medien erfolgt nach derselben Vorgehensweise. Analog werden die Belohnungen für die Bodentraglast berechnet. Für den Fall, dass keine Nachrüstung möglich ist, wird die Aktion als ungültig gekennzeichnet und entsprechend der Vorgehensweise zur Platzierung von zwei Funktionseinheiten auf derselben Position gehandhabt (siehe Kapitel 5.1).

$$R_B(t) = \frac{\sum_{i=1}^{N_{Größe, m_t}} r_{pos,i}}{N_{Größe, m_t}} \quad (4-14)$$

$$R_M(t) = \frac{\sum_{i=1}^{N_{Größe, m_t}} r_{pos,i}}{N_{Größe, m_t}} \quad (4-15)$$

$$r_{pos,i} = \begin{cases} 0; & \text{Anforderungen erfüllt} \\ -0,5; & \text{Anforderungen nicht erfüllt; Nachrüstaufwand gering} \\ -1; & \text{Anforderungen nicht erfüllt; Nachrüstaufwand hoch} \end{cases} \quad (4-16)$$

Ein Beispiel für die Berechnung der Medienverfügbarkeit ist in Abbildung 4-8 visualisiert. Das Layout ist durch drei Bereiche charakterisiert. Im ersten Bereich (links) steht das betrachtete Medium (z. B. Druckluft) zur Verfügung. Im mittleren Bereich ist das Medium nicht verfügbar, kann aber bei Bedarf unaufwändig nachgerüstet werden, während im rechten Bereich des Layouts ein hoher Nachrüstaufwand besteht. Es wird angenommen, dass alle Funktionseinheiten das Medium benötigen. Daher erreicht Funktionseinheit 1 einen Belohnungswert von 0, Funktionseinheit 2 von $-0,25$ und Funktionseinheit 3 von -1 .

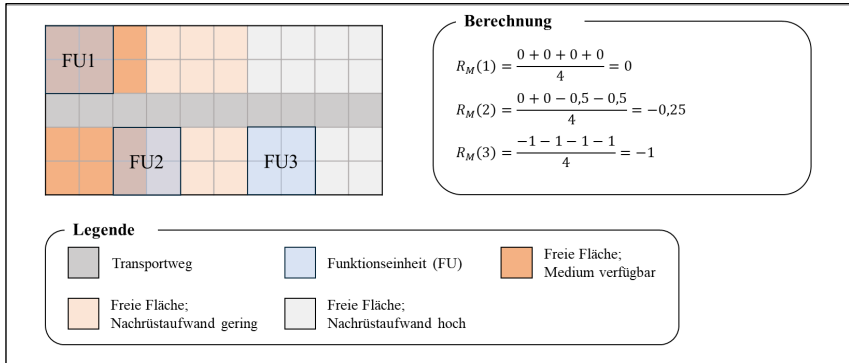


Abbildung 4-8: Berechnung von R_M für alle Funktionseinheiten

Die **Ähnlichkeit** (R_A) ist eine umplanungsbezogene Zielgröße und erfordert ein bestehendes Layout, das als Vergleichsgröße dient. Das Ziel der Ähnlichkeit ist es, dass das neue Layout dem ursprünglichen Layout entspricht. Verschiebungen oder Rotationen einer Funktionseinheit werden daher bestraft (siehe Gleichungen 4-17). Hierdurch wird der Umstrukturierungsaufwand auf ein Minimum reduziert. Um dennoch eine Umplanung durchzuführen, muss die Ähnlichkeit mit einer weiteren Zielgröße kombiniert werden. In diesem Fall sollte der RL-Agent Funktionseinheiten mit signifikantem Verbesserungspotential neu positionieren, während die restlichen Funktionseinheiten in ihrer Position verbleiben. Weiterhin wird zwischen fixierten und nicht fixierten Funktionseinheiten unterschieden. Fixierte Funktionseinheiten werden auf der Ausgangsposition platziert und generieren eine

Belohnung von 0. Nicht fixierte Funktionseinheiten können neu positioniert werden. Ein Beispiel mit drei Funktionseinheiten ist in Abbildung 4-9 dargestellt. Funktionseinheit 1 wird auf der gleichen Position platziert. Daher wird eine Belohnung von 0 realisiert. Für Funktionseinheit 2 wird gedreht, was in einer Belohnung von $-0,2$ resultiert. Da Funktionseinheit 3 verschoben wird, wird die Summe der x- und y-Differenz zwischen alter und neuer Position berechnet ($|x_{m_t \text{old}} - x_{m_t \text{new}}|$ bzw. $|y_{m_t \text{old}} - y_{m_t \text{new}}|$) und durch den geschätzten maximalen Distanzwert $\Delta_{\max}$ geteilt. Dieser Wert entspricht der Summe der x- und y-Dimension des Layouts.

$$R_A(t) = \begin{cases} 0; & \text{bei gleicher Positionierung} \\ -0,2; & \text{bei Rotation} \\ -\frac{|x_{m_t \text{old}} - x_{m_t \text{new}}| + |y_{m_t \text{old}} - y_{m_t \text{new}}|}{\Delta_{\max}}; & \text{bei einer Verschiebung} \end{cases} \quad (4-17)$$

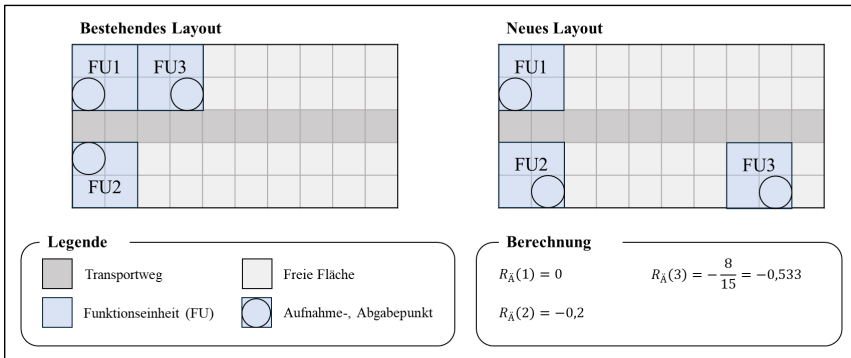


Abbildung 4-9: Berechnung von R_A für alle Funktionseinheiten

4.5.2 Integration von EDS-Ergebnissen

Für die Integration der dynamischen Zielgrößen ist es notwendig, ein Simulationsmodul zu entwickeln, das von der Umgebung initialisiert, ausgeführt und ausgewertet werden kann. Die hierfür entwickelte EDS wurde Python-basiert auf Basis des Open-Source-Pakets *SimPy* entwickelt. Der Ausgangspunkt für die Anwendung des Simulationsmodells ist ein fertig generiertes Fabriklayout, in dem alle Funktionseinheiten platziert wurden. Das Simulationsmodul muss mit unterschiedlichen Layoutkonfigurationen und Eigenschaften von Produktionssystemen kompatibel sein. Es wird daher modular entwickelt. Im Folgenden werden die notwendigen Eingangsinformationen sowie die zentralen Funktionalitäten der EDS erklärt. Die zentralen Funktionalitäten werden als Klassen implementiert und durch die Eingangsinformationen parametrisiert. Dazu gehören die Funktionseinheiten, Transportmittel, Produktionsplanung und -steuerung, Materialflusssteuerung und die externe Logistik (siehe Anhang A, Abbildung 11-1).

Die Klasse der **Funktionseinheiten** hat die zentrale Funktion, Produkte herzustellen. Dazu werden eingehende (Zwischen-) Produkte, die sich im Eingangsbestand der Funktionseinheit

befinden, für eine definierte Prozesszeit bearbeitet und zu neuen (Zwischen-) Produkten kombiniert. Funktionseinheiten sind durch folgende Eingangsdaten charakterisiert:

- ☐ Identifikationsnummer oder eindeutiger Name der Funktionseinheit
- ☐ Typ der Funktionseinheit: Maschine oder Lager
- ☐ Erforderliche 2D- oder optionale 3D-Geometrie
- ☐ Maximalbestand der Funktionseinheit
- ☐ Anfangsbestand an Produkten, der zu Beginn der Simulation im Eingangs- und Ausgangspuffer vorhanden ist
- ☐ Informationen pro Rüstzustand:
 - Prozesszeiten
 - Rüstzeiten
 - Eingangs-Ausgangsbeziehung

Eine Funktionseinheit des Typs Maschine besitzt die Funktion „Produzieren“, die entsprechend dem aktuellen Rüstzustand Produkte herstellt, wenn genügend Eingangsprodukte vorliegen. Die fertigen Produkte werden anschließend in das Ausgangslager überführt. Bei einer Funktionseinheit des Typs Lager existieren keine Prozesszeiten und die eingehenden Produkte entsprechen den ausgehenden Produkten.

Die **Transportmittel** haben die zentrale Funktion, Produkte zwischen verschiedenen Orten im Produktionssystem zu transportieren. Dabei werden aufzuladende Produkte aus dem Ausgangspuffer einer Funktionseinheit entnommen und dem Eingangspuffer einer anderen Funktionseinheit hinzugefügt. Folgende Eingangsdaten charakterisieren die Klasse der Transportmittel:

- ☐ Identifikationsnummer oder eindeutiger Name des Transportmittels
- ☐ Ladekapazität
- ☐ Lade- und Entladezeiten
- ☐ Transportgeschwindigkeit

Die Klasse der **Produktionsplanung und -steuerung** führt die Reihenfolgeplanung durch und legt die Rüstzustände aller Funktionseinheiten fest. Als Eingangsdaten dient eine Liste mit allen Aufträgen, den Auftragsmengen und Produktionsschritten sowie einer Zuordnung, welche Funktionseinheit welchen Produktionsschritt durchführen kann. Auf Basis dieser Informationen und der Logik der Reihenfolgeplanung (z. B. First-In-First-Out (FiFo)) wird für jede Funktionseinheit festgelegt, in welcher Reihenfolge die Produktionsschritte durchgeführt werden. Wenn die entsprechende Menge eines Produktionsschrittes gefertigt wurde, wird der Rüstzustand der Funktionseinheiten entsprechend der Auftragsreihenfolge angepasst. Die Produktionsplanungs- und -steuerungsklasse bestimmt außerdem das Simulationsende, das dann erreicht ist, wenn alle Produkte hergestellt wurden, und berechnet anschließend die Leistungsindikatoren.

Die Klasse der **Materialflusssteuerung** steuert die Transportmittel entsprechend dem gewählten Steuerungsprinzip (z. B. Pull-Prinzip). Dazu verfügt die Materialflusssteuerung über ein Observationsmodul, das die Bestände in den Eingangs- und Ausgangspuffern aller Funktionseinheiten sowie die Menge der aktuell transportierten Produkte überwacht und diese mit den Informationen der Reihenfolgeplanung harmonisiert. Mit einer Prioritätsliste wird festgelegt, welche Transporte als nächstes durchgeführt werden müssen. Ein Eintrag der

Prioritätsliste gibt die zu transportierende Menge an und legt fest, an welcher Funktionseinheit die Produkte auf- bzw. abgeladen werden sollen. Die Mengenzuordnung wird über eine weitere Liste definiert, die angibt, wie viele Produkte zu einem Bündel zusammengefasst werden können. Dieses Bündel entspricht der Menge an Produkten, die z. B. in einer Transportbox gleichzeitig transportiert werden können. Ein Transportmittel mit freien Kapazitäten wird anschließend dem Eintrag der Prioritätsliste zugeordnet und führt diesen aus. Für die Wegfindung wird ein Dijkstra-Algorithmus genutzt. Dazu werden die Elementarzellen des Layouts in einen Graphen überführt. Benachbarte Elementarzellen, die den Zustand „frei“ oder „Transportweg“ besitzen, werden als Knoten aufgefasst und durch eine Kante der Länge 1 verbunden. Zusätzlich werden die zugehörigen Aufnahme- und Abgabepunkte als Knoten in den Graphen integriert. Anschließend kann der kürzeste Weg zwischen der aktuellen Position des Transportmittels im Layout und den Aufnahme- und Abgabepunkten berechnet und dem Transportmittel zugeteilt werden.

Die Klasse der **externen Logistik** liefert Produkte an ausgewählte Funktionseinheiten, z. B. das Wareneingangslager. Dazu wird für jede zu beliefernde Funktionseinheit die Frequenz, die Art und Menge einer Lieferung definiert. Ferner wird die maximal zu liefernde Gesamtmenge festgelegt. Die Produkte werden in der festgelegten Frequenz an den Eingangspuffer der Funktionseinheit geliefert, sofern ausreichend Platz vorhanden ist.

Zur Ausführung der EDS sind **Schnittstellen** zwischen der Umgebung und dem Simulationsmodul erforderlich. Als Eingangsinformationen für das Simulationsmodul dienen die neuen Positionen der Funktionseinheiten. Eine Aktualisierungsfunktion passt die Position entsprechend an und startet den nächsten Simulationslauf. Somit werden alle Elemente des Simulationsmoduls zu Beginn des Trainings einmalig initialisiert und danach nur noch aktualisiert. Im Anschluss an den Simulationslauf wertet eine weitere Funktion die resultierenden Leistungsindikatoren aus und gibt diese in Form eines Arrays an die Umgebung zurück. Diese können dann zur Berechnung der Belohnungswerte verwendet werden. Folgende Zielgrößen können für die Optimierung verwendet werden:

Durch die Minimierung der **Durchlaufzeit** (R_{DZ}) kann die Zeit bis zur Fertigstellung eines Auftrags verkürzt werden, sodass mehr Aufträge im gleichen Zeitintervall bearbeitet werden können. Der Belohnungswert wird durch eine Kombination der funktionseinheitbezogenen Durchlaufzeit DZ_{mt} und der absoluten Durchlaufzeit des gesamten Produktionssystems DZ_{abs} , die für alle Funktionseinheiten gleich ist, berechnet (siehe Gleichung 4-18). Um einen Belohnungswert zwischen -1 und 0 zu erhalten, werden die Komponenten durch eine geschätzte maximale Durchlaufzeit DZ_{max} dividiert, gewichtet und addiert. Hierdurch wird sichergestellt, dass die absolute Durchlaufzeit des Produktionssystems minimiert wird und gleichzeitig jede Aktion zu einem anderen Belohnungswert führt. Alternativ kann auch nur eine Komponente berücksichtigt werden, dies führt jedoch zu einer längeren Trainingszeit [Klar22a].

$$R_{DZ}(t) = -0.5 \cdot \frac{DZ_{mt}}{DZ_{max}} - 0.5 \cdot \frac{DZ_{abs}}{DZ_{max}} \quad (4-18)$$

Der Belohnungswert zur **Auslastung der Funktionseinheiten** (R_{AF}) wird anhand von Gleichung 4-19 berechnet. Für jede Funktionseinheit wird die zur Wertschöpfung genutzte Zeit T_{ws,m_t} durch die absolute Durchlaufzeit des Produktionssystems dividiert, bevor ein Wert von 1 subtrahiert wird. Hierdurch wird der Belohnungswert auf einen Wertebereich zwischen -1 und 0 transformiert. Das Ziel besteht darin, die Auslastung der Funktionseinheiten zu maximieren, sodass eine höhere Auslastung zu einem Wert nahe 0 führt.

$$R_{AF}(t) = \frac{T_{ws,m_t}}{DZ_{abs}} - 1 \quad (4-19)$$

Die Berechnung des Belohnungswertes für die **Auslastung der Transportmittel** (R_{AT}) erfolgt für das gesamte Produktionssystem und kann nicht der Position einer einzelnen Funktionseinheit zugeordnet werden. Daher wird jeder Aktion einer Episode der gleiche Belohnungswert zugeordnet (siehe Gleichung 4-20). Zur Berechnung der Wertschöpfungszeit wird die Wartezeit $T_{warte,i}$ aller N_T Transportmittel von der absoluten Durchlaufzeit des Produktionssystems subtrahiert und normiert. Anschließend wird der Mittelwert aller Quotienten gebildet und auf einen Wertebereich von -1 bis 0 transformiert. Auf diese Weise werden Materialflusssysteme, die zu einer hohen Transportmittelauslastung führen, belohnt, während ungenutzte Transportmittel zu schlechteren Belohnungswerten führen.

$$R_{AT} = \frac{\sum_{i=1}^{N_T} \frac{DZ_{abs} - T_{warte,i}}{DZ_{abs}}}{N_T} - 1 \quad (4-20)$$

Schließlich kann der Belohnungswert für den **Stau im Materialfluss** (R_S) mittels Gleichung 4-21 berechnet werden. Auch dieser Belohnungswert ist für alle Aktionen einer Episode gleich, da der Stau im Materialfluss hauptsächlich durch die gesamte Layoutkonfiguration verursacht wird. Zur Berechnung werden alle Wartezeiten der Transportmittel addiert und durch eine geschätzte maximale Wartezeit $T_{warte,max}$ dividiert. Demnach besteht die Zielsetzung in einer Reduzierung des Staus im Materialfluss.

$$R_S = - \frac{\sum_{i=1}^{N_T} T_{warte,i}}{T_{warte,max}} \quad (4-21)$$

Abschließend ist zu betonen, dass die geschätzten Maximalwerte für alle Zielgrößen einmalig bei der Initialisierung des Trainings festgelegt werden und während des gesamten Trainings konstant bleiben.

5 Experimentelle Entwicklung des Reinforcement Learning Agent

Dieses Kapitel behandelt die Entwicklung und Auswahl des RL-Agenten. Hierbei wird eine Strategie zur Prävention ungültiger Aktionen ausgewählt, der genutzte Agent festgelegt und eine Struktur zur multikriteriellen Optimierung definiert. Da diese Elemente nicht auf Basis der Literatur festgelegt werden können, da sie problemspezifisch auszuwählen sind, werden für jedes Element experimentelle Untersuchungen durchgeführt. Die Leistungsfähigkeit der verschiedenen Elemente wird miteinander verglichen und das leistungsfähigste Element wird für die Folgeversuche übernommen. Dabei werden alle Experimente mit dem gleichen Setup durchgeführt (72 Stunden Trainingsdauer auf einem Intel Xeon E5-2670 Prozessor unter Nutzung von 30 GB RAM), sodass die Vergleichbarkeit der Ergebnisse gewährleistet ist.

5.1 Umgang mit ungültigen Aktionen

Die erste experimentelle Untersuchung adressiert den Umgang mit ungültigen Aktionen. Eine ungültige Lösung entsteht immer, wenn die Aktion zu mindestens einem der folgenden Zustände führt:

- ☐ Eine Elementarzelle ist durch mindestens zwei Funktionseinheiten belegt.
- ☐ Die Funktionseinheit wird auf mindestens einer Elementarzelle mit dem Zustand „Wand“ platziert.
- ☐ Die Funktionseinheit wird auf mindestens einer Elementarzelle mit dem Zustand „Transportweg“ platziert.
- ☐ Die Funktionseinheit wird so platziert, dass sie oder bereits platzierte Funktionseinheiten nicht mehr durch die interne Logistik erreicht werden können.
- ☐ Die Funktionseinheit wird auf mindestens einer Elementarzelle platziert, die die Anforderungen an die Medienverfügbarkeit nicht erfüllt und nicht nachgerüstet werden kann.
- ☐ Die Funktionseinheit wird auf mindestens einer Elementarzelle platziert, die die Anforderungen zur Bodentraglast nicht erfüllt und nicht nachgerüstet werden kann.

Beispiele für ungültige Fabriklayoutzustände wurden bereits in Abbildung 4-4 dargestellt. Das Ziel des Trainings ist es, zulässige und optimierte Fabriklayoutvarianten zu generieren. Zur Prävention ungültiger Zustände kann entweder eine Maskierungsfunktion genutzt werden oder Aktionen, die zu ungültigen Fabriklayouts führen, werden bestraft. Die Funktionsweise der Maskierungsfunktion ist in Abbildung 5-1 visualisiert. Demnach prüft die Maskierungsfunktion vor der Aktionsumsetzung, ob eine zulässige oder unzulässige Aktion vorliegt. Im Fall einer zulässigen Aktion wird eine alternative Aktion ausgewählt. Dieser Vorgang wird solange wiederholt, bis eine zulässige Aktion ausgewählt wurde oder keine Aktion mehr zur Verfügung steht. Letzteres führt zum Abbruch der Episode. Bei einem Nutzen-basierten Agenten wird der approximierte Nutzenwert verwendet, um die Aktionen beginnend mit dem höchsten approximierten Aktionsnutzen absteigend zu ordnen. Bei Verwendung eines Policy-basierten Agenten werden die Aktionen nacheinander entsprechend der approximierten Wahrscheinlichkeitsverteilung gezogen, um eine Reihenfolge zu definieren.

Alternativ können ungültige Aktionen durch die Belohnungsfunktion bestraft werden. In diesem Fall wird einer ungültigen Aktion der geringsten Belohnungswert (-1) zugeordnet. Hieraus ergibt sich die durch Gleichung 5-1 definierte Belohnungsfunktion.

$$R(t)_{\text{new}} = \begin{cases} R(t); & \text{bei valider Aktion} \\ -1; & \text{sonst} \end{cases} \quad (5-1)$$

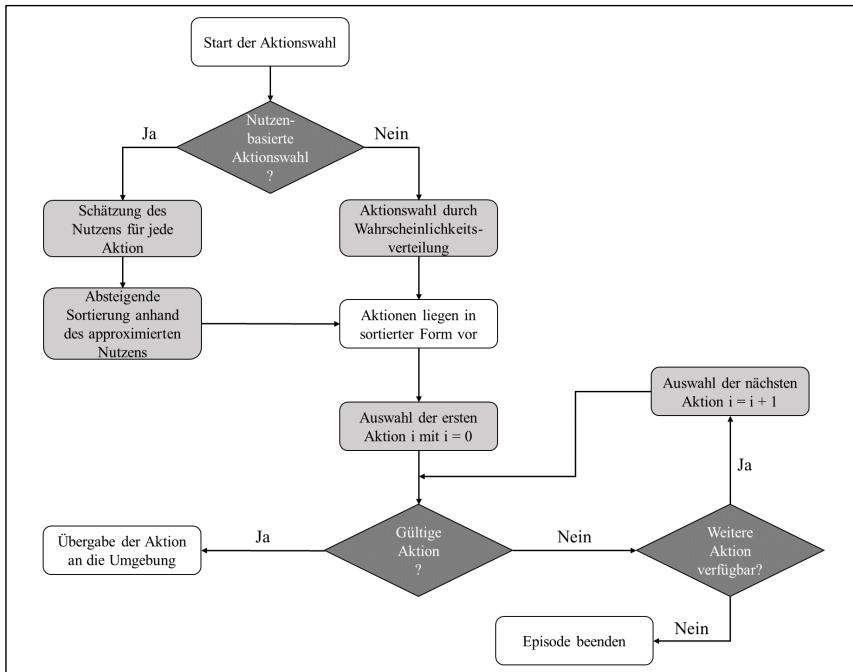


Abbildung 5-1: Funktionsweise der Maskierungsfunktion

Um die leistungsfähigste Alternative zu identifizieren, werden die verschiedenen Ansätze zur Prävention ungültiger Aktionen experimentell miteinander verglichen. Dazu wird ein *DDQN*-Agent mit einem multidiskreten Aktionsspielraum als Referenzagent verwendet und auf ein Fabriklayoutplanungsproblem mit 20 zu platzierenden Funktionseinheiten und einem Layoutraster mit den Dimensionen 30 x 7 (30 Zellen in der x-Dimension und 7 Zellen in der y-Dimension) angewendet. Als Zielgröße wird lediglich die Transportintensität betrachtet, daher gilt $w_5 = 1$. Zum Vergleich der Ansätze wird zu jedem Zeitpunkt das aktuell beste Layout gespeichert und anhand der realisierten Episodenbelohnungen bewertet. Dabei werden Fabriklayoutvarianten bevorzugt, die eine höhere Episodenbelohnung realisieren. Ein weiteres Vergleichskriterium ist das Konvergenzverhalten. Im Idealfall steigt die durchschnittliche Belohnungskurve zu Beginn des Trainings an und konvergiert dann zu einem möglichst hohen Wert. Hierbei ist ein Trainingsverhalten zu bevorzugen, das zu Beginn möglichst steil ansteigt und zu einem möglichst hohen durchschnittlichen Belohnungswert konvergiert.

Die Ergebnisse der experimentellen Untersuchung sind in Abbildung 5-2 visualisiert. Es sind deutliche Unterschiede zwischen den Belohnungskurven der beiden Ansätze zu erkennen. Die Belohnungskurve des Agenten, der die Maskierungsfunktion nutzt, liegt dauerhaft über der Belohnungskurve, die ohne die Maskierungsfunktion realisiert wurde. Die Unterschiede können durch das zu erlernende Verhalten begründet werden. Der Agent mit Maskierungsfunktion erzeugt bereits in der ersten Episode ein zulässiges Layout. Daher liegt der Fokus des Trainings ausschließlich auf der Optimierung des Layouts. Im alternativen Fall muss der Agent zunächst den Unterschied zwischen zulässigen und unzulässigen Lösungen lernen, bevor eine Optimierung angestrebt werden kann. Aufgrund der Explorationsstrategie werden daher vorrangig unzulässige Lösungen erzeugt, weshalb geringere Episodenbelohnungswerte realisiert werden. Es ist jedoch zu beobachten, dass der gleitende Durchschnitt auch für den Agenten ohne Maskierungsfunktion ansteigt und gegen ein konstantes Niveau konvergiert, sodass von einem Trainingserfolg ausgegangen werden kann. Da das Niveau jedoch unter dem Niveau des Agenten mit Maskierungsfunktion liegt, wird nachfolgend die Maskierungsfunktion für die Prävention ungültiger Aktionen verwendet.

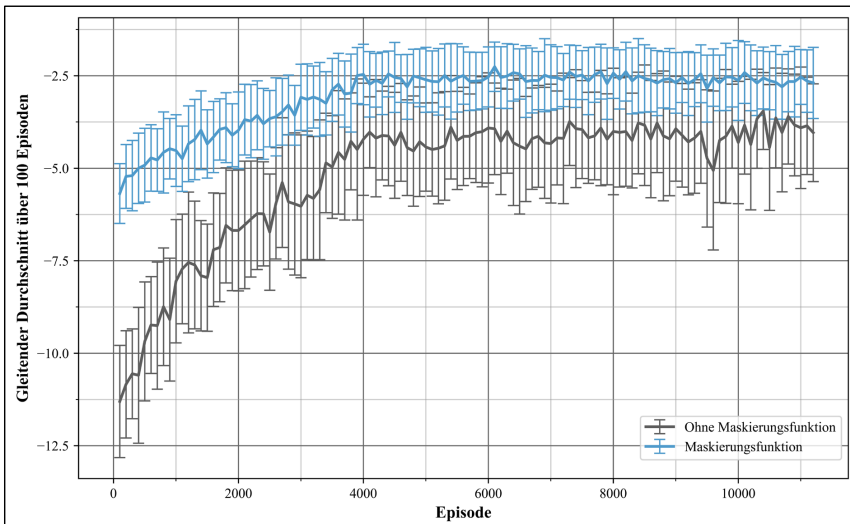


Abbildung 5-2: Vergleich des Lernverhaltens mit und ohne Nutzung der Maskierungsfunktion

5.2 Festlegung des Reinforcement Learning Agent Typs

Neben dem Umgang mit ungültigen Aktionen muss die genaue Konfiguration des Agenten festgelegt werden. Ziel ist es auch hier, die Leistungsfähigkeit des Ansatzes zu steigern. Die in Kapitel 2.4.3 charakterisierten Agenten werden in verschiedenen Varianten bezüglich des Aktionsspielraums implementiert (siehe Tabelle 5-1). Dabei wird ein diskreter Aktionsspielraum mit einem d gekennzeichnet (z. B. VPG_d). Analog dazu werden Agenten mit einem multidiskreten Aktionsspielraum mit md (z. B. VPG_{md}) und einem multikontinuierlichen mit mk (z. B. PPO_{mk}) ausgewiesen. Da die Leistung von jedem der 13 Agenten von den gewählten Hyperparametern abhängt, müssen diese optimiert werden, bevor

die Leistungsfähigkeit verglichen werden kann. Aufgrund der Vielzahl an Hyperparameterkombinationen wird für die Hyperparameteroptimierung ein vierschrittiges Vorgehen gewählt:

1. Festlegung der Standardparameter
2. Evaluation der KNN-Konfiguration
3. Evaluation des Optimizers und der Learning Rate
4. Evaluation der agentenspezifischen Hyperparameter

Tabelle 5-1: Übersicht über implementierte Agenten und dem zugehörigen Aktionsspielraum

		Aktionsspielraum		
		Diskret (d)	Multidiskret (md)	Multikontinuierlich (mk)
Agent	VPG	X	X	
	Rainbow DQN	X	X	
	A2C	X	X	X
	PPO	X	X	X
	SAC	X	X	X

Für die Hyperparameteroptimierung aller Agenten wird das gleiche Anwendungsbeispiel wie zuvor (Kapitel 5.1) verwendet. Da der Ansatz jedoch auf Probleme unterschiedlicher Größe anwendbar sein soll, erfolgt anschließend eine Übertragung auf ein kleineres und ein größeres Fabriklayoutplanungsproblem, sodass die Leistungsfähigkeit umfassend verglichen werden kann. Die Eigenschaften der Fabriklayoutplanungsprobleme werden in Abbildung 5-3 skizzenhaft zusammengefasst.

Das Fabriklayoutplanungsproblem mit reduzierter Komplexität besitzt 7×7 Elementarzellen und es sollen 4 Funktionseinheiten platziert werden. Es existiert kein bereits definierter Transportweg, sondern nur zwei Bereiche, die den Wareneingang und -ausgang kennzeichnen. Das 3. Fabriklayoutplanungsproblem ist durch eine höhere Komplexität gekennzeichnet. Hierbei werden 50 Funktionseinheiten positioniert. Des Weiteren ist der Anteil vordefinierter Strukturen im Verhältnis zu verfügbaren Freiflächen höher als in den vorangegangenen Fabriklayoutplanungsproblemen. Dieses 3. Anwendungsszenario ist somit der aus Anwendungssicht repräsentativste Fall und wird für die Auswahl des Agenten als priorisiert angesehen.

Die Hyperparameteroptimierung des *Rainbow DQN_{md}*-Agenten wird exemplarisch im Detail vorgestellt, da ein analoges Vorgehen für alle Agenten verwendet wurde. Als Referenzproblem wird das 2. Fabriklayoutplanungsproblem mit 20 Funktionseinheiten betrachtet.

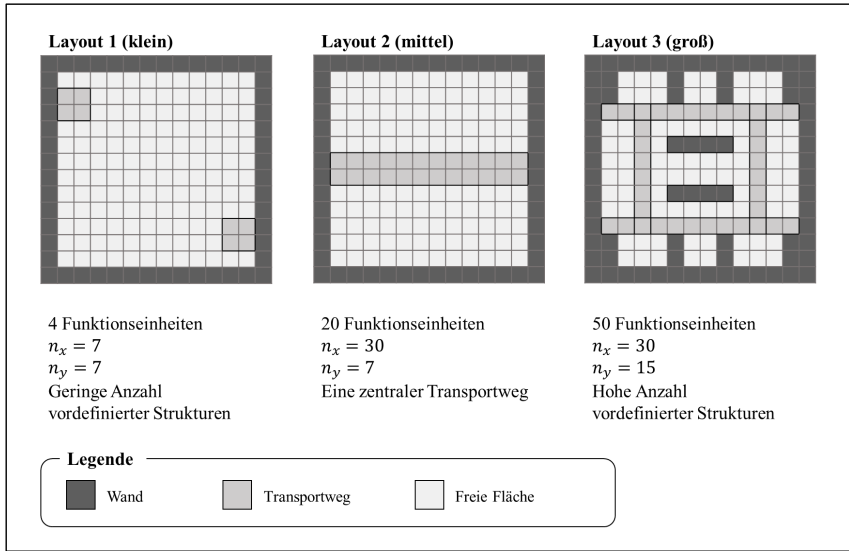


Abbildung 5-3: Fabriklayoutplanungsprobleme zur Evaluation der Leistungsfähigkeit der RL-Agenten

Im **1. Optimierungsschritt** werden die Standardparameter des *Rainbow DQN_{md}*-Agenten festgelegt. Diese bilden den Ausgangspunkt für die nachfolgenden Untersuchungen. Nach Abschluss eines Optimierungsschritts, werden die Standardparameter aktualisiert und als Startparameter für die weiteren Optimierungsschritte genutzt. Die Standardparameter des *Rainbow DQN_{md}*-Agenten sind in Tabelle 5-2 definiert. Die Learning Rate wird auf 0,01 gesetzt und der Adam Optimizer wird verwendet. Der Parameter γ ist ein Diskontierungsfaktor, der angibt, wie stark die Belohnungswerte zukünftiger Aktionen im Vergleich zur aktuellen Aktion gewichtet werden. Die Parameter ε_{start} , ε_{Min} und ε_N beschreiben die Explorationsstrategie (ε -Greedy). Der Agent wählt mit der Wahrscheinlichkeit ε eine zufällige Strategie. Diese Wahrscheinlichkeit beträgt zu Beginn des Trainings ε_{start} und wird linear über ε_N Episoden auf ε_{Min} reduziert. N_{Batch} wird für das Training des Agenten genutzt und gibt an, wie viele Transitionen aus dem Replay-Buffer zwischen den Episoden gezogen werden, um den Agenten zu trainieren. Höhere N_{Batch} Werte führen bei gleichbleibender Episodenzahl zu einem Anstieg der Berechnungszeit. Die Parameter α und β beschreiben die Eigenschaften des PER des *Rainbow DQN_{md}*-Agenten und werden als $\alpha = 0,2$ und $\beta = 0,5$ definiert. Diese Hyperparameterwerte basieren auf vorangegangenen Untersuchungen und haben zu guten Ergebnissen geführt. Es ist jedoch davon auszugehen, dass das volle Leistungspotential noch nicht ausgeschöpft ist.

Tabelle 5-2: Standardparameter des *Rainbow DQN_{md}*-Agenten

Parameter	Wert
Learning Rate	0,01
Optimizer	Adam
γ	0,6
ε_{Start}	0,4
ε_{Min}	0,1
ε_N	4000 Episoden
N_{Batch}	20
α	0,2
β	0,5

Der **2. Optimierungsschritt** zielt auf die Verbesserung der KNN-Konfiguration und der Aktivierungsfunktion ab. Die Tabellen in Abbildung 5-4 fassen die KNN-Konfigurationen und die berücksichtigten Aktivierungsfunktionen zusammen. Die KNN-Konfigurationen werden mit steigender Anzahl versteckter Schichten und unterschiedlicher Anzahl von Neuronen pro Schicht definiert, um möglichst heterogene Konfigurationen zu berücksichtigen. So besteht die 1. KNN-Konfiguration aus einer versteckten Schicht mit 10 Neuronen, während die 5. KNN-Konfiguration 12 versteckte Schichten umfasst. Zusätzlich werden 4 Aktivierungsfunktionen berücksichtigt. Zur Bestimmung der Hyperparameterkonfiguration mit der höchsten Leistungsfähigkeit wird ein vollfaktorieller Versuchsplan erstellt und ausgeführt. Dieser umfasst 20 verschiedene Parameterkombinationen pro Agent. Die Leistung wird anhand der realisierten Episodenbelohnung für das beste generierte Layout im Trainingsdurchlauf bewertet. Die Ergebnisse aller Trainingsdurchläufe sind in Abbildung 5-4 zusammengefasst. Das beste Ergebnis wird mit *Relu* als Aktivierungsfunktion und KNN-Konfiguration 5 erzielt. Diese Parametereinstellung wird für alle folgenden Optimierungsschritte verwendet.

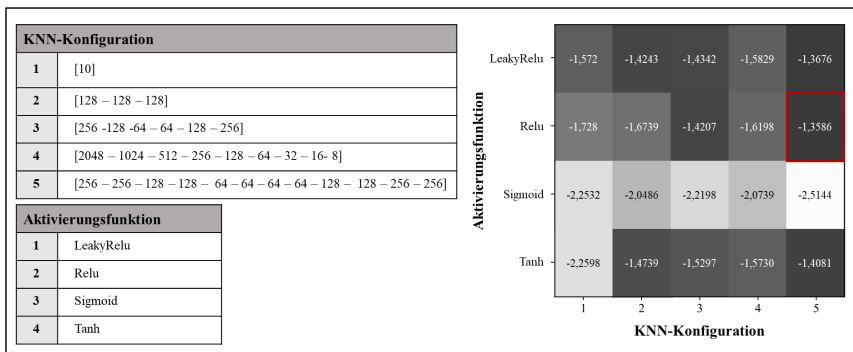


Abbildung 5-4: Ergebnisse des 2. Optimierungsschritts mit dem Fokus auf die KNN-Konfiguration und das Aktivierungsfunktion für den *Rainbow DQN_{md}*-Agent

Der **3. Optimierungsschritt** adressiert die Auswahl der Learning Rate und des Optimizers. Die berücksichtigten Parameter werden in Abbildung 5-5 zusammengefasst. Die Learning Rate wird in Zehnerpotenzen von 0,000001 auf 0,1 erhöht, während der Adam und der Stochastic Gradient Decent (SGD) Optimizer berücksichtigt werden. Somit werden für den 3. Optimierungsschritt 12 Trainingsdurchläufe pro Agent durchgeführt. Für den

Rainbow DQN_{md}-Agenten wird das beste Ergebnis mit einer Learning Rate von 0,0001 unter Verwendung des Adam Optimizer erzielt. Ferner ist zu beobachten, dass kleine bis mittlere Learning Rates beim Adam Optimizer zu besseren Ergebnissen führen als eine hohe Learning Rate. Bei Verwendung des SGD Optimizers führen hingegen hohe Learning Rates zu besseren Ergebnissen.

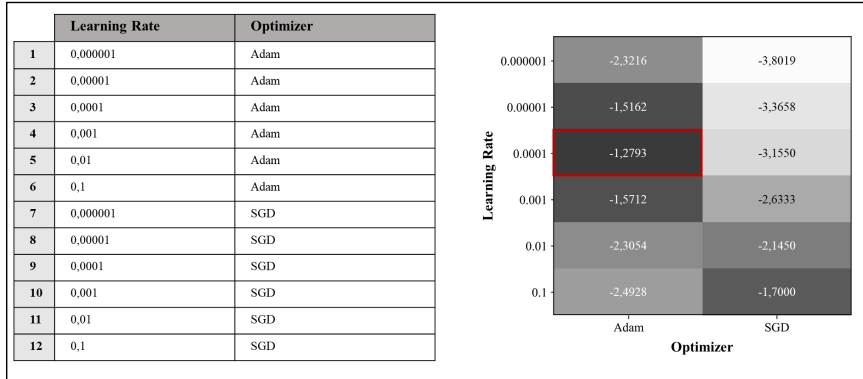


Abbildung 5-5: Ergebnisse des 3. Optimierungsschritts mit dem Fokus auf die Learning Rate und den Optimizer für den *Rainbow DQN_{md}*-Agent

Im **4. Optimierungsschritt** werden die agentenspezifischen Hyperparameter untersucht. Die berücksichtigten Parameter und die Versuchsergebnisse der 36 Parameterkombinationen werden in Abbildung 5-6 zusammengefasst. Die Ergebnisse der Trainingsdurchläufe zeigen, dass die besten Ergebnisse mit einer Hyperparameterkombination von $\varepsilon_{Start} = 0,8$, $\varepsilon_N = 12000$, $\gamma = 0,6$ und $N_{Batch} = 20$ erzielt werden. Zusammenfassend wird die Leistungsfähigkeit des *Rainbow DQN_{md}*-Agenten durch die Hyperparameteroptimierung von $-1,356$ (bestes Ergebnis nach dem 2. Optimierungsdurchlauf) auf $-1,236$ verbessert. Dies entspricht einer Leistungssteigerung von 9,11 %.

Die Optimierung der übrigen Agenten erfolgt analog. Anschließend werden die Agenten entsprechend der zuvor identifizierten Hyperparameter initialisiert und auf die beiden Referenzszenarien (Layoutplanungsprobleme 1 und 3) angewendet. Der Vergleich der Ergebnisse wird anhand von zwei Kriterien durchgeführt. Das zentrale Kriterium ist die maximale Episodenbelohnung für den Trainingsdurchlauf, die angibt, wie gut das beste Layout ist. Für den Fall, dass zwei Agenten den gleichen maximalen Episodenbelohnungswert erzielen, ist die Trainingsdauer bis zur Generierung des Layouts relevant.

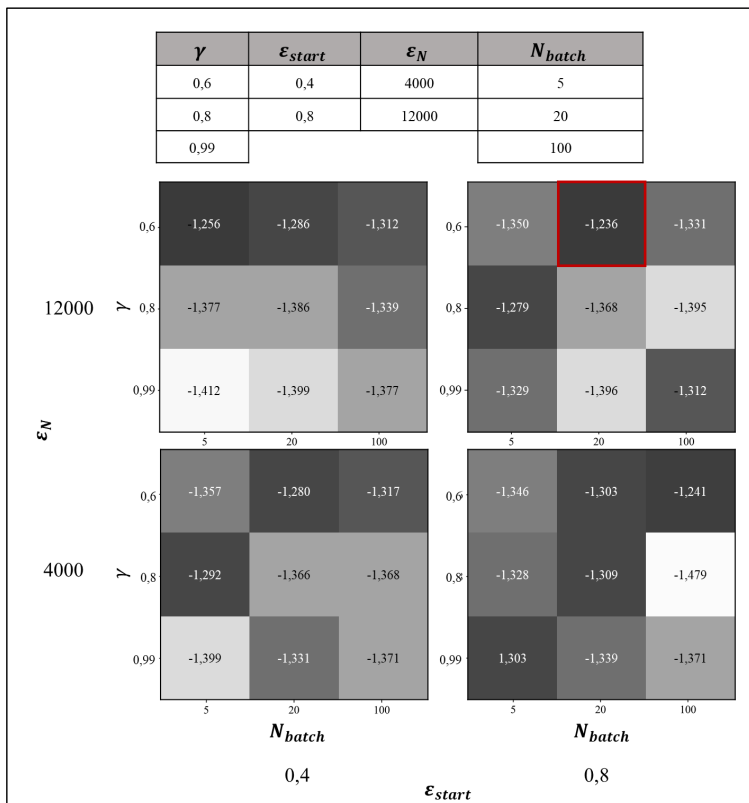


Abbildung 5-6: Ergebnisse des 4. Optimierungsschritts mit dem Fokus auf agenten-spezifische Hyperparameter für den *Rainbow DQN_{md}*-Agent

Abbildung 5-7 fasst den Trainingsprozess aller RL-Agenten für die drei betrachteten Anwendungsszenarien zusammen. Es werden der Zeitpunkt der Verbesserung und der zugehörige Episodenbelohnungswert abgebildet. Weiterhin ist hervorzuheben, dass alle RL-Agenten für die gleiche Zeit (72 Stunden) trainiert wurden. Ein fehlender Punkt am Ende der Trainingszeit bedeutet daher nicht, dass der Agent nicht die gesamten 72 Stunden trainiert wurde, sondern dass am Ende des Trainings keine weitere Verbesserung der Lösung erreicht wurde.

Die Trainingsergebnisse zeigen, dass die Auswahl des Agenten einen signifikanten Einfluss auf die Leistung und das Trainingsverhalten des RL-Ansatzes hat. Im Allgemeinen ist zu beobachten, dass die größten Verbesserungen im ersten Viertel der Trainingsdauer realisiert werden. Im anschließenden Trainingsverlauf können zwar noch weitere Verbesserungen erzielt werden, diese fallen jedoch geringer aus und die zeitlichen Abstände zwischen den Verbesserungen nehmen zu.

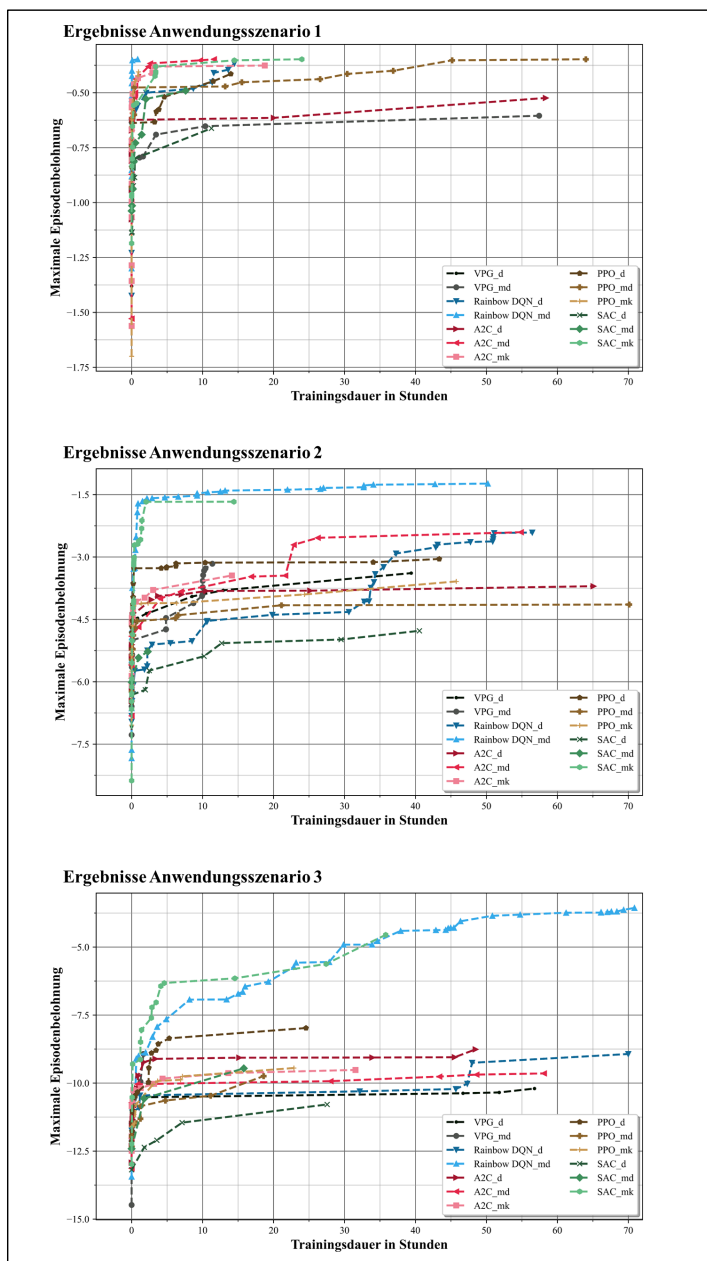


Abbildung 5-7: Visualisierung der Trainingsergebnisse für alle betrachteten Anwendungsszenarien zur Auswahl des leistungsfähigsten RL-Agenten

Die Ergebnisse zeigen, dass die Unterschiede zwischen den Belohnungswerten mit der Problemgröße zunimmt, und dass der *Rainbow DQN_{md}*-Agent und der *SAC_{mk}*-Agent in allen Anwendungsszenarien die höchsten Episodenbelohnungen und damit die besten Layouts erzeugen. Darüber hinaus wird deutlich, dass die multidiskreten Agenten in der Regel bessere Ergebnisse generieren als die diskreten Agenten desselben Agententyps. Eine mögliche Erklärung für diese Beobachtung kann die Anzahl der verfügbaren Aktionen sein, die im multidiskreten Fall kleiner als im diskreten Fall ist. Jedoch ist keine eindeutige Tendenz zwischen multidiskreten und multikontinuierlichen Agenten erkennbar. Das 3. Anwendungsszenario ist von besonderer Bedeutung, da es die höchste Problemkomplexität aufweist. Auch hier erzielen der *Rainbow DQN_{md}*-Agent und der *SAC_{mk}*-Agent die besten Resultate. Bis zu einer Trainingsdauer von 50 % sind die Ergebnisse der Agenten vergleichbar. Anschließend kann der *SAC_{mk}* Agent jedoch keine Verbesserung mehr erzeugen, während der *Rainbow DQN_{md}*-Agent das Layout weiter optimiert. Daraus kann geschlussfolgert werden, dass der *Rainbow DQN_{md}*-Agent bessere Fähigkeiten zur Überwindung lokaler Minima besitzt. Für die weiteren Entwicklungen wird daher ab sofort der *Rainbow DQN_{md}*-Agent verwendet.

5.3 Festlegung der Agentenstruktur zur multikriteriellen Optimierung

Im nächsten Entwicklungsschritt wird die Agentenstruktur für die multikriterielle Optimierung festgelegt. Wie in Kapitel 2.4.2 beschrieben, werden vorrangig zwei Grundstrukturen für die multikriterielle Optimierung verwendet. In der ersten Struktur wird ein einzelner Agent trainiert, der die Linearkombination der Belohnungsfunktion lernt. Im zweiten Fall wird ein Agent für jede Zielgröße der Belohnungsfunktion trainiert. Die Aktionsauswahl erfolgt durch Aggregation der Ergebnisse der einzelnen Agenten unter Berücksichtigung der gewählten Priorisierung in Form einer Linearkombination. Nachfolgend werden die beiden Strukturen anhand der Leistungsfähigkeit verglichen.

Das für den Vergleich verwendete Anwendungsszenario ist in Abbildung 5-8 abgebildet. Es umfasst 8 Funktionseinheiten, die in einem Layout der Größe 18 x 7 positioniert werden sollen. Für den Materialtransport wird ein Gabelstapler genutzt, der den Transportweg und freie Flächen zum Transport verwenden kann. Für die Wegfindung wird ein Dijkstra-Algorithmus genutzt und die Reihenfolgeplanung wird mittels einer FiFo-Logik unter Berücksichtigung einer Push-Steuerung durchgeführt. Die weiteren Prozesszeiten und Abhängigkeiten der Funktionseinheiten sind in der Tabelle in Abbildung 5-8 zusammengefasst. Der Produktionsprozess endet nach der Herstellung von 30 Produkten. Die benötigten Zukaufteile werden durch einen externen Lieferanten alle 10 Minuten an die erste Funktionseinheit geliefert, sofern diese freie Lagerkapazitäten besitzt. Es wird davon ausgegangen, dass alle Funktionseinheiten Druckluft als Medium benötigen. Diese ist nur in den Randbereichen des Layouts verfügbar, kann aber im mittleren Bereich nachgerüstet werden. Daher ist die erste Zielgröße die Optimierung hinsichtlich der Medienverfügbarkeit. Außerdem werden die Durchlaufzeit und die Übersichtlichkeit des Materialflusses als Zielgrößen betrachtet. Die Zielgrößen weisen konfliktäre Eigenschaften auf, da die Medienverfügbarkeit optimiert wird, wenn die Funktionseinheiten an den Rändern des Layouts angeordnet werden. Im Falle der Durchlaufzeit ist zu erwarten, dass die Funktionseinheiten in unmittelbarer Nähe positioniert werden, während die Übersichtlichkeit des Materialflusses optimiert wird, wenn die Funktionseinheiten in einer Linie platziert sind. Das Anwendungsszenario erlaubt somit die

Untersuchung der Leistungsfähigkeit der Agentenstrukturen im Fall einer multikriteriellen Optimierung mit konfliktären Zielgrößen.

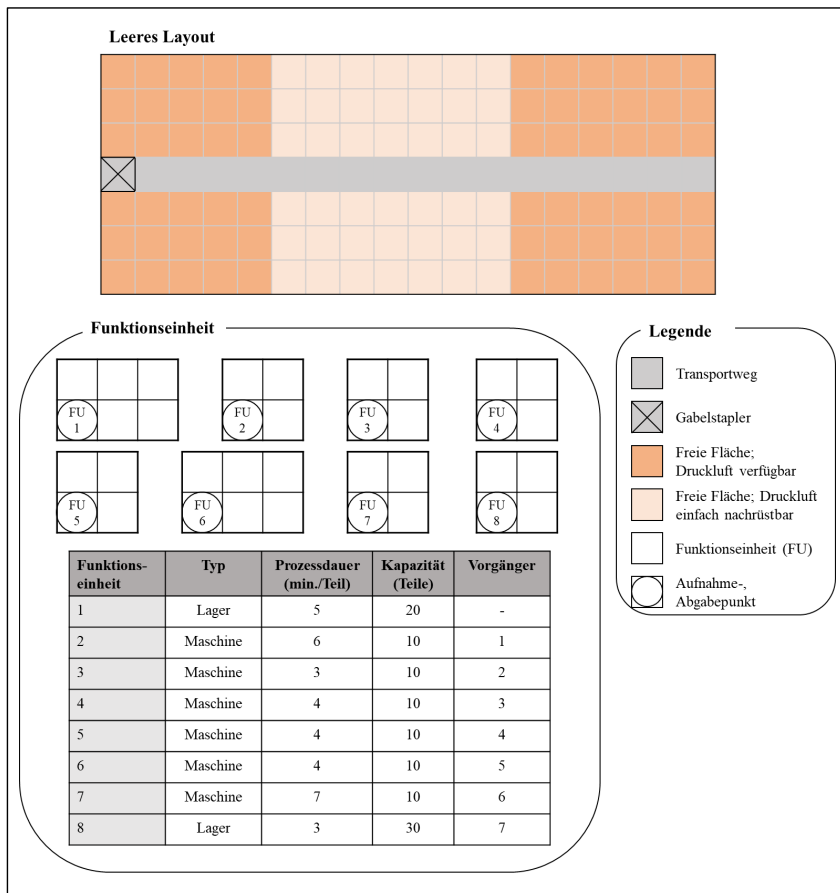


Abbildung 5-8: Anwendungsszenario zur Festlegung der Agentenstruktur im Fall einer multikriteriellen Optimierung

Die Leistungsfähigkeit wird anhand von drei Schritten untersucht (siehe Tabelle 5-3). Zu Beginn werden drei Versuche durchgeführt, die lediglich eine der drei Zielgrößen berücksichtigen. Hierdurch kann eine Vergleichslösung für jede Zielgröße generiert werden. Diese kann anschließend genutzt werden, um die Qualität der generierten Lösungen im multikriteriellen Fall zu evaluieren. Im zweiten Schritt werden die Durchlaufzeit und die Medienverfügbarkeit parallel betrachtet. Die Priorisierung der Zielgrößen wird variiert, sodass unterschiedliche Lösungen je nach Priorisierung zu erwarten sind. Im abschließenden Schritt werden alle drei Zielgrößen berücksichtigt. Es muss hervorgehoben werden, dass die 1. Funktionseinheit bei allen Lösungen in der oberen linken Ecke des Layouts platziert wird, da

Unterschiede in den Lösungen hierdurch einfacher analysiert werden können, da der Startpunkt der Prozesskette eindeutig definiert ist.

Tabelle 5-3: Priorisierung der drei Zielgrößen zur Untersuchung der Leistungsfähigkeit in der multikriteriellen Optimierung

	Versuchs Nr.	Durchlaufzeit (w_1)	Medienverfügbarkeit (w_6)	Übersichtlichkeit (w_8)
Schritt 1	1	1,00	0,00	0,00
	2	0,00	1,00	0,00
	3	0,00	0,00	1,00
Schritt 2	4	0,99	0,01	0,00
	5	0,80	0,20	
	6	0,60	0,40	
	7	0,50	0,50	
	8	0,40	0,60	
	9	0,20	0,80	
	10	0,01	0,99	
Schritt 3	11	0,20	0,60	0,20
	12	0,20	0,40	0,40
	13	0,40	0,40	0,20
	14	0,60	0,20	0,20
	15	0,20	0,20	0,60
	16	0,40	0,20	0,40
	17	0,33	0,33	0,33
	18	0,19	0,80	0,01
	19	0,01	0,80	0,19
	20	0,80	0,01	0,19
	21	0,80	0,19	0,01
	22	0,19	0,01	0,80
	23	0,01	0,19	0,80

Die Referenzergebnisse des 1. Schritts sind in Abbildung 5-9 dargestellt und entsprechen den zuvor definierten Erwartungen an die Lösungen. Im ersten Versuch wird nur die Durchlaufzeit als Zielgröße betrachtet. Die beste Lösung führt zu einer Episodenbelohnung von $-3,416$ und die Funktionseinheiten werden gruppiert angeordnet. Im zweiten Versuch wird die Medienverfügbarkeit priorisiert. Die beste Lösung wird mit einer Episodenbelohnung von $0,00$ bewertet. Daher wurde eine optimale Lösung generiert und die Funktionseinheiten wurden ausschließlich in Bereichen mit verfügbarer Druckluft platziert. Da die Durchlaufzeit nicht berücksichtigt wurde, sind die Funktionseinheiten nicht materialflussorientiert angeordnet und auch die Aufnahme- und Abgabepunkte werden nicht optimal ausgerichtet. Die Lösung, die auf die Übersichtlichkeit des Materialflusses abzielt, unterscheidet sich von den bisherigen Lösungen. Die Funktionseinheiten bilden eine Linie, wodurch die optimale Episodenbelohnung von $0,00$ erreicht wird. Wie in Abbildung 5-9 zu erkennen ist, werden in Abhängigkeit der Zielgrößenpriorisierung unterschiedliche Lösungen generiert, weshalb eine parallele Optimierung aller Zielgrößen nicht möglich ist. Die Ergebnisse erlauben somit Rückschlüsse auf das Verhalten der Agentenstrukturen bei der Lösung von Planungsproblemen mit Zielkonflikten zu ziehen.

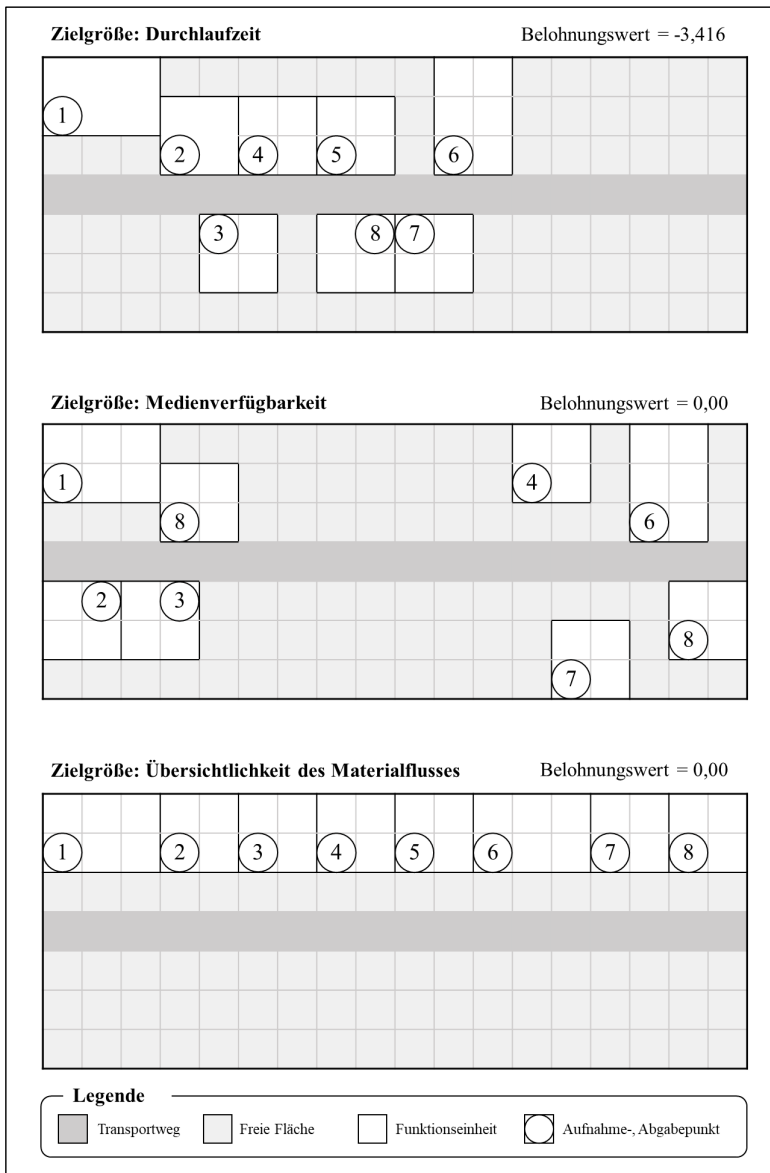


Abbildung 5-9: Referenzlösungen für alle Zielgrößen

Im zweiten Untersuchungsschritt werden die Durchlaufzeit und die Medienverfügbarkeit gleichzeitig als Zielgrößen betrachtet. Es werden sieben Versuche mit abnehmender Priorisierung der Durchlaufzeit und zunehmenden Priorisierung der Medienverfügbarkeit durchgeführt. Die Ergebnisse der Experimente werden in Abbildung 5-10 hinsichtlich der

erzielten Belohnung für beide Zielgrößen analysiert. Es kann beobachtet werden, dass unterschiedliche Priorisierungen zu unterschiedlichen Fabriklayouts führen. Die Lösungsvarianz bei Verwendung der ersten Agentenstruktur ist höher als bei der zweiten Agentenstruktur, was als Indiz für die Fähigkeit zur Überwindung lokaler Minima gewertet wird. Die zweite Agentenstruktur generiert hingegen teilweise identische Lösungen, obwohl unterschiedliche Priorisierungen genutzt wurden. Ferner sind die von der ersten Agentenstruktur generierten Lösungen mit einer Ausnahme dominant gegenüber den Lösungen der zweiten Agentenstruktur. Dominante Lösungen sind solche, die in mindestens einer Zielgröße bessere Ergebnisse erzielen, während in allen anderen Zielgrößen gleiche Ergebnisse erreicht werden. Darüber hinaus kann anhand von Abbildung 5-10 der Zielkonflikt zwischen den Zielgrößen analysiert werden. Die erste Agentenstruktur kann optimale Lösungen für die Medienverfügbarkeit generieren, wenn diese Zielgröße mit mindestens 50 % priorisiert wird. Eine höhere Priorisierung der Durchlaufzeit reduziert diese weiter, während bei der Medienverfügbarkeit ein schlechteres Ergebnis erreicht wird. Allerdings wird auch bei der höchsten Priorisierung der Durchlaufzeit die Referenzlösung nicht erreicht. Eine mögliche Erklärung hierfür ist, dass die Medienverfügbarkeit immer mit mindestens 1 % berücksichtigt wird. Folglich kann der Agent die Medienverfügbarkeit nicht vollständig ignorieren.

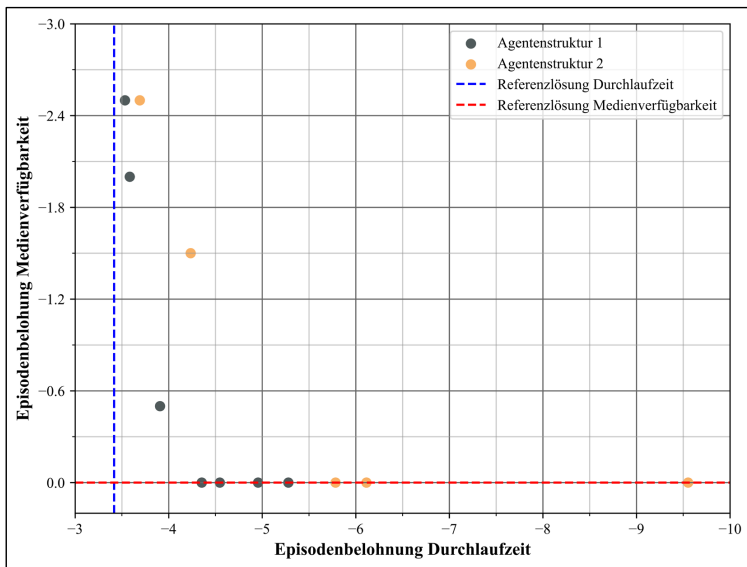


Abbildung 5-10: Vergleich der Agentenstrukturen für die multikriterielle Optimierung von zwei Zielgrößen

Der dritte Untersuchungsschritt berücksichtigt alle Zielgrößen und dient der Verifizierung der vorangegangenen Ergebnisse. Da die Analyse einer 3D-Pareto-Front komplexer ist als im zweidimensionalen Fall, wird eine Distanzmetrik als Quantifizierungswerkzeug verwendet. Diese misst die Differenz zu einem definierten Referenzpunkt, der durch den Schnittpunkt der drei Referenzlösungen aus Schritt 1 definiert ist. Die Ergebnisse sind in Tabelle 5-4 zusammengefasst. Kleinere Differenzen entsprechen besseren Lösungen, da diese näher an der

realisierten Referenzlösung liegen. Die erste Agentenstruktur generiert für die ersten beiden Zielgrößen erneut bessere Layouts als die zweite Agentenstruktur. Für die dritte Zielgröße kann beobachtet werden, dass die zweite Agentenstruktur bessere Lösungen generiert. Allerdings ist die Differenz der erzielten Mittelwerte für die Zielgrößen 1 und 2 größer als für die letzte Zielgröße. Analoge Ergebnisse können für die maximale Differenz festgestellt werden. Zusätzlich kann anhand der minimalen Differenz festgestellt werden, inwiefern die Referenzlösung reproduziert werden kann. Auch hier erzielt die erste Agentenstruktur die besseren Ergebnisse, da diese mindestens einmal in der Dimension der Medienverfügbarkeit und Übersichtlichkeit die optimale Lösung generiert.

Zusammenfassend erreicht Agentenstruktur 1 die besseren Ergebnisse im bi- und multikriteriellen Fall und wird daher in den folgenden Kapiteln zur multikriteriellen Optimierung verwendet.

Tabelle 5-4: Vergleich der Agentenstrukturen unter Berücksichtigung von drei Zielgrößen

Referenzlösung	Durchlaufzeit (R_{DZ})	Medienverfügbarkeit (R_M)	Übersichtlichkeit (R_U)
	-3,416	-0,00	-0,00
Durchschnittliche Differenz und Standardabweichung aller Lösungen			
Agentenstruktur 1	$1,27 \pm 0,65$	$0,71 \pm 0,75$	$0,85 \pm 0,50$
Agentenstruktur 2	$2,64 \pm 3,06$	$2,11 \pm 0,50$	$0,71 \pm 0,19$
Minimale Differenz			
Agentenstruktur 1	0,15	0,00	0,00
Agentenstruktur 2	0,27	0,20	0,25
Maximale Differenz			
Agentenstruktur 1	2,34	2,00	1,63
Agentenstruktur 2	6,14	2,30	0,84

6 Erklärbares Reinforcement Learning für die Fabriklayoutplanung

6.1 Definition der Zielgrößen

Wie in Kapitel 2.4.5 beschrieben, weisen konventionelle, automatisierte Planungsansätze Schwächen hinsichtlich der Transparenz im Optimierungsprozess auf. Aus Sicht der Planenden wäre es wünschenswert, zusätzliche Einblicke in die wesentlichen Einflussfaktoren des Optimierungsprozesses zu erhalten. Hierzu zählen folgende Fragestellungen:

- ☐ Ist die Belohnungsfunktion adäquat gewählt?
- ☐ Ist die generierte Lösung das Ergebnis der erlernten Strategie oder eines rein explorativen Verhaltens?
- ☐ Kann der RL-Ansatz lokale Minima überwinden?
- ☐ Wie groß ist der Einfluss der Transportintensität? Was verändert sich, wenn die Transportintensität zwischen zwei funktionalen Einheiten zu- oder abnimmt?
- ☐ Wie verändert sich die Lösung, wenn Freiflächen in Sperrflächen umgewandelt werden und nicht mehr für die Positionierung genutzt werden können, oder umgekehrt?
- ☐ Wurden bei der Generierung des Layouts suboptimale Entscheidungen getroffen, weil z. B. Positionierungskonflikte berücksichtigt werden mussten?

Diese Informationen können durch einen geeigneten XRL-Ansatz bereitgestellt werden. Ausgehend von diesen Fragestellungen werden ein primäres und ein sekundäres Untersuchungsziel verfolgt. Das primäre Untersuchungsziel adressiert die Qualität der generierten Lösung. Es wird untersucht, ob die generierte Lösung ein Ergebnis der erlernten Strategie des RL-Ansatzes ist oder ob rein zufällige Entscheidungen zur Lösungsgenerierung genutzt wurden. Außerdem soll die Frage beantwortet werden, ob die Belohnungsfunktion adäquat gewählt wurde. Der XRL-Ansatz dient zur Steigerung der Vertrauenswürdigkeit in die generierte Lösung und soll für Layoutplanungsprobleme mit unterschiedlichen Eigenschaften anwendbar sein.

Da es jedoch kein gemeinsames Verständnis von Vertrauenswürdigkeit und Interpretierbarkeit gibt und XRL bisher nicht in der RL-basierten Layoutplanung angewendet wurden, wird eine Definition dieser Begriffe für den betrachteten Kontext vorgeschlagen. Es wird davon ausgegangen, dass die Anwender des entwickelten XRL-Ansatzes über Erfahrungen im Bereich der Fabrikplanung verfügen. Infolgedessen können sie beurteilen, ob eine Lösung von hoher oder niedriger Qualität ist. Das Hauptaugenmerk des XRL-Ansatzes liegt darauf, ob eine Lösung willkürlich oder nach einer präzisen Strategie gefunden wurde. Die Vertrauenswürdigkeit wird also dadurch gewährleistet, dass willkürlich generierte Lösungen identifiziert und nicht in einer manuellen Feinplanungsphase berücksichtigt werden. Die vorgestellten XRL-Ansätze können folglich nicht direkt zur Messung der Vertrauenswürdigkeit verwendet werden, aber Einblicke in den Generierungsprozess der Layouts geben und zur Interpretierbarkeit des Lernprozesses beitragen. So können zufällig generierte Lösungen identifiziert und vermieden werden, was, je nach Ergebnis, die Vertrauenswürdigkeit eines betrachteten Layouts erhöht.

Im Fokus des sekundären Untersuchungsziels steht die Bedeutung der Input-Feature im Entscheidungsprozess. Die Interpretation dieser Ergebnisse ist jedoch problemspezifisch und kann sich in Abhängigkeit von den Problemcharakteristika ändern. Daher werden Methoden und Interpretationsvorschläge vorgestellt, die vorrangig für eine Zielgruppe mit einem tiefen Layoutplanungs- und RL-Verständnis von Interesse sind. Es muss jedoch darauf hingewiesen werden, dass die Ergebnisse nicht allgemein übertragbar sind.

6.2 Entwicklung der XRL-Ansätze

Nachfolgend wird die Funktionsweise und Ergebnisinterpretation der entwickelten XRL-Ansätze anhand eines exemplarischen Anwendungsszenarios (siehe Abbildung 6-1) erklärt. Dieses umfasst ein Layout der Größe 7 x 5, in dem 4 Funktionseinheiten platziert werden sollen. Die zugehörige Prozesskette und Transportmatrix ist in Abbildung 6-1 visualisiert und die Transportintensität wird als Zielgröße zur Optimierung verwendet.

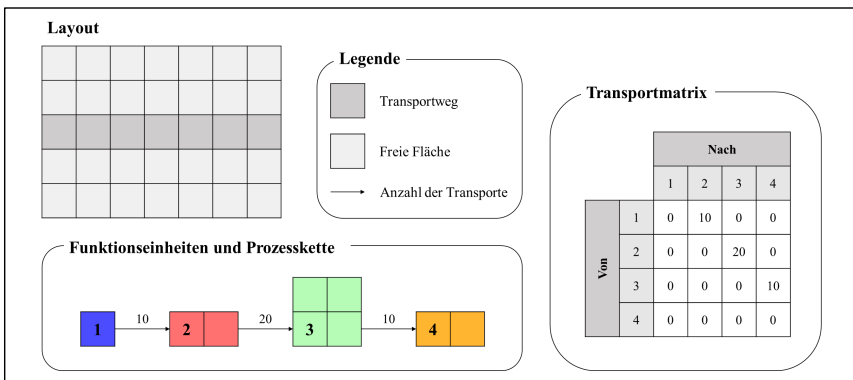


Abbildung 6-1: Exemplarisches Anwendungsszenario zur Erklärung des XRL-Ansatzes

6.2.1 Policy Summarization

Der Ausgangspunkt für die Analyse des Trainingsverhaltens mit einem Policy Summarization Ansatz sind Transitionen. Eine Transition besteht aus dem aktuellen Zustand, der gewählten Aktion, dem Folgezustand und der zugehörigen Belohnung. Die Ergebnisse des Policy Summarization Ansatzes ermöglichen eine Ergebnisanalyse im Kontext des primären Untersuchungsziels.

Es wird zwischen zwei unterschiedlichen Policy Summarization Ansätzen unterschieden, da diese sowohl funktionseinheitenbezogen als auch layoutbezogen ausgewertet werden können. Die funktionseinheitenbezogene Policy Summarization (PSFU) analysiert die gewählten Aktionen für jede Funktionseinheit separat. Das Ziel ist es, alle Aktionen zu identifizieren, die in einem maximalen Belohnungswert der Transition resultieren. Demnach aggregiert die PSFU die kurzfristig besten Aktionen im Trainingsprozess für jede Funktionseinheit. Im Gegensatz dazu analysiert die layoutbezogene Policy Summarization (PSL) das gesamte Layout auf Basis des übergeordneten Optimierungsziels, das in diesem Fall die Transportintensität ist. Das Ergebnis der PSL ist eine Zusammenfassung der k besten Layouts, die im Trainingsprozess generiert wurden. Falls mehr als k Layouts mit der gleichen Bewertung existieren, kann die Anzahl der aggregierten Layouts den Wert k übersteigen.

Die Ergebnisse der beiden Policy Summarization Ansätze können in eine Verteilungs- und eine Übereinstimmungsanalyse überführt werden. Die **Verteilungsanalyse** wird erstellt, indem die k besten Layouts der PSL zerlegt und auf Funktionseinheitsebene aggregiert werden. Zwei Beispiele, die für die Funktionseinheiten 1 und 4 generiert werden, sind in Abbildung 6-2 dargestellt. Beispielhaft wird $k = 8$ gewählt. Folglich werden die besten 8 Layouts extrahiert und auf Funktionseinheitsebene aggregiert. Daher wird für jede Funktionseinheit ein Diagramm erstellt, indem diese entsprechend ihrer Position in den $k = 8$ besten Layouts platziert wird. Wenn die betrachteten Layouts ähnlich sind, wird eine Funktionseinheit auf denselben oder teilweise überlappenden Positionen platziert werden. Die Methode zählt für jedes Element innerhalb des Layouts, wie oft es von einer Funktionseinheit belegt wurde, was zu den Verteilungen der Funktionseinheiten 1 und 4 führt, die in Abbildung 6-2 visualisiert sind. Für Funktionseinheit 1 ist zu erkennen, dass sie in allen 8 Layouts an derselben Stelle positioniert wird. Die Positionen von Funktionseinheit 4 variieren dagegen stärker. Weiterhin ist zu erkennen, dass es aufgrund der Größe von 1×2 teilweise zu Überschneidungen kommen kann. Im betrachteten Fall wurde das Element ($X = 3, Y = 4$) als Mittelpunkt der Positionierung verwendet. Allerdings wird die Funktionseinheit in verschiedene Richtungen gedreht. Die Ergebnisse der Verteilungsanalyse können als erster Indikator zur Beurteilung der Positionierungssicherheit herangezogen werden. Eine breit gestreute Verteilung wird mit einer erhöhten Unsicherheit assoziiert, während kleinere Variationen in der Positionierung die Vertrauenswürdigkeit in die generierten Lösungen erhöhen.

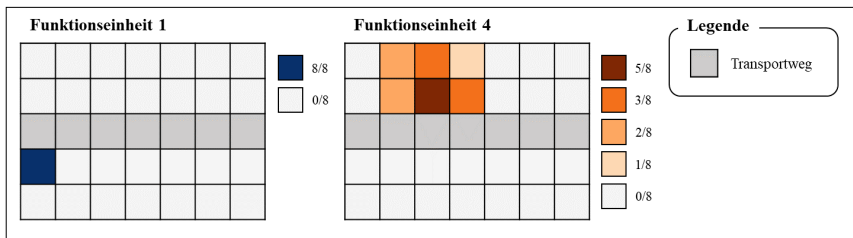


Abbildung 6-2: Verteilungsanalyse für Funktionseinheit 1 und 4

Die **Übereinstimmungsanalyse** aggregiert die Informationen der PSFU und der PSL. Ziel ist es zu analysieren, ob die besten Aktionen für eine individuelle Funktionseinheit auch zu den besten Layouts führen. Die besten Aktionen für eine Funktionseinheit werden anhand der realisierten Belohnung für die gewählte Aktion gemessen. Anschließend wird jedes der k besten Layouts hinsichtlich dieser Aktionen analysiert. Wurde ein Layout durch die besten funktionseinheitbezogenen Aktionen generiert, liegt eine Übereinstimmung vor. Folglich lässt sich durch die Übereinstimmungsanalyse aufzeigen, ob die Belohnungsfunktion auf das übergeordnete Optimierungsziel abgestimmt ist. Exemplarische Ergebnisse sind in Abbildung 6-3 dargestellt. Jeder Eintrag auf der y-Achse entspricht einer der k besten Lösungen, während die Funktionseinheiten auf der x-Achse zusammengefasst werden.

Z. B. steht $PSL = 2$ für das zweite Layout aus den besten $k = 8$ Layouts. $PSFU = 3$ symbolisiert die dritte Funktionseinheit bzw. die beste Aktion, gemessen an der erreichten Schrittbelohnung. Die Einträge der Übereinstimmungsanalyse sind binär. Eine Übereinstimmung von $PSL = 2$ und $PSFU = 1$ wird erreicht, wenn Funktionseinheit 1 im zweiten der besten $k = 8$ Layouts entsprechend der besten bekannten Aktion für diese

Funktionseinheit positioniert wird. Im Gegensatz dazu stellt eine fehlende Übereinstimmung eine Platzierungsentscheidung dar, bei der die Funktionseinheit nicht in der höchstmöglichen Schrittbelohnung resultiert, obwohl das Layout von hoher Qualität ist. Die Übereinstimmung symbolisiert folglich, dass hohe kurzfristige Belohnungen auch zu Layouts mit hoher Qualität führen.

Im ersten Beispiel in Abbildung 6-3 liegt für Funktionseinheit 1, 2 und 4 eine nahezu vollständige Übereinstimmung vor. Das bedeutet, dass die Aktionen mit den besten Belohnungen auch zu den besten Layouts führen. Für Funktionseinheit 3 wird nur ein Layout durch die beste, bekannte Aktion generiert. Eine fehlende Übereinstimmung kann unterschiedlich interpretiert werden. Entweder kann es darauf hinweisen, dass eine Aktion gewählt wurde, die kurzfristig nicht optimal ist, aber langfristig als vorteilhaft anzusehen ist. Die wäre mit einem Damenopfer im Schach vergleichbar. Alternativ könnte eine konfliktäre Positionierungsentscheidung die Ursache sein. In diesem Fall kann nur eine Funktionseinheit auf der optimalen Position platziert werden, weshalb eine fehlende Übereinstimmung für die zweite Funktionseinheit entsteht. Darüber hinaus kann eine fehlende Übereinstimmung ein Zeichen für ein überwiegend zufälliges Platzierungsverhalten, eine unzuweckmäßige Belohnungsfunktion oder eine fehlende Fähigkeit zur Überwindung lokaler Minima sein. Dieser Fall wird im zweiten Beispiel in Abbildung 6-3 visualisiert. Da in diesem Beispiel keine klare Struktur in der Übereinstimmungsanalyse zu erkennen ist, sollte ein Planer die Ergebnisse aus Beispiel 1 den Ergebnissen aus Beispiel 2 vorziehen. Darüber hinaus ist auch der zeitliche Verlauf der Übereinstimmungsanalyse relevant. Ein Ergebnis wird als vertrauenswürdig angesehen, wenn die Übereinstimmungsanalyse schrittweise von einem unstrukturierten Zustand in einen strukturierten Zustand übergeht.

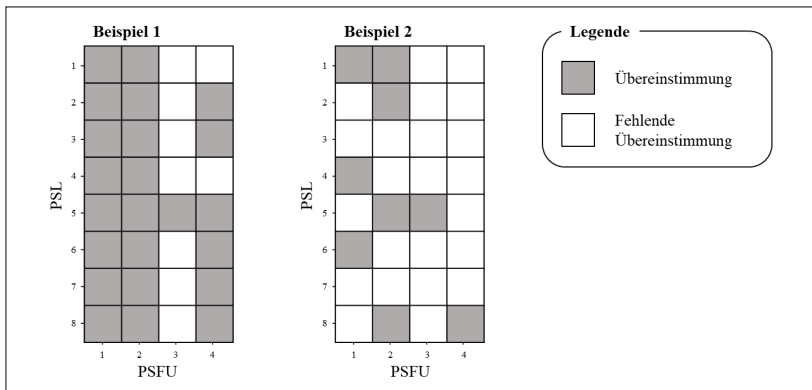


Abbildung 6-3: Exemplarische Ergebnisse der Übereinstimmungsanalyse

6.2.2 Occlusion-Sensitivitätsansätze

Die Ergebnisse der Occlusion-Sensitivitätsansätze können zur Erreichung von beiden XRL-Untersuchungszielen beitragen. Hierfür wird der Ansatz auf die materialfluss- und layoutbezogenen Input-Features angewendet. Bei den materialflussbezogenen Informationen werden einzelne Materialflussverbindungen zwischen zwei Funktionseinheiten schrittweise

ausgeblendet. Dazu wird die Transportintensität zwischen den Funktionseinheiten in der Transportmatrix auf den Wert 0 reduziert. Um den Effekt dieser Änderung zu quantifizieren, wird entweder die Differenz der Q-Werte mit und ohne ausgeblendeter Materialflussbeziehung berechnet oder es wird die Differenz der Zustandswerte (State-Value) bestimmt. Im ersten Fall erfolgt eine aktionsbezogene Auswertung, während im zweiten Fall eine aktionsunabhängige Auswertung durchgeführt wird.

Für die layoutbezogenen Informationen stehen folgende Analyseformen zur Verfügung:

- ☐ Veränderung einer freien Fläche zu einer belegten Fläche
- ☐ Veränderung einer belegten Fläche zu einer freien Fläche
- ☐ Entfernen einer bereits platzierten Funktionseinheit

Bei den ersten beiden Analyseformen wird ein Bereich von 2×2 verändert und über das gesamte Layout verschoben. Im letzten Fall wird nur die betrachtete Funktionseinheit ausgeblendet.

Für das primäre Untersuchungsziel wird die Veränderung der Differenzen im Trainingsverlauf analysiert. Es wird davon ausgegangen, dass der Agent zu Beginn des Trainings den Effekt der Veränderung eines einzelnen Input-Features überschätzt, da er noch keine Kenntnisse über die Relevanz der Input-Feature besitzt. Im fortgeschrittenen Trainingsstadium hat der Agent die Metriken des Problems gelernt und kann die Positionierung entsprechend der Änderungen anpassen. Daher wird erwartet, dass sich der Q-Wert und der Zustandswert entsprechend ändern, allerdings mit einer geringeren Gesamtabweichung. Demnach sollte die Gesamtabweichung zu Beginn des Trainings höher sein als zum Schluss.

Die Ergebnisse der Occlusion-Sensitivitätsanalyse, die auf die Zustandsinformationen der Transportmatrix angewendet wurden, sind in Abbildung 6-4 visualisiert. Zur Erstellung der Analyse wurde nach je 6000 Trainingsschritten eine Occlusion-Sensitivitätsanalyse durchgeführt, bei der die maximale Abweichung zwischen den initialen und den ausgeblendeten Zustandsinformationen bestimmt wird. Es kann beobachtet werden, dass die maximale Abweichung im Trainingsverlauf abnimmt. Folglich reagiert der Agent zu Beginn des Trainings sensitiver auf Änderungen der Zustandsinformationen als am Ende des Trainings, was als Zeichen für eine vertrauenswürdige Platzierungsstrategie interpretiert wird. Darüber hinaus ist die maximale Abweichung für die Platzierung von Funktionseinheit 1 am höchsten und nimmt mit den Funktionseinheiten schrittweise ab. Dies könnte dadurch erklärt werden, dass Änderungen am Start einer Episode größere Einflüsse auf das endgültige Layout haben als bei den letzten Platzierungsentscheidungen, da am Ende einer Episode die meisten Layoutmerkmale bereits festgelegt sind.

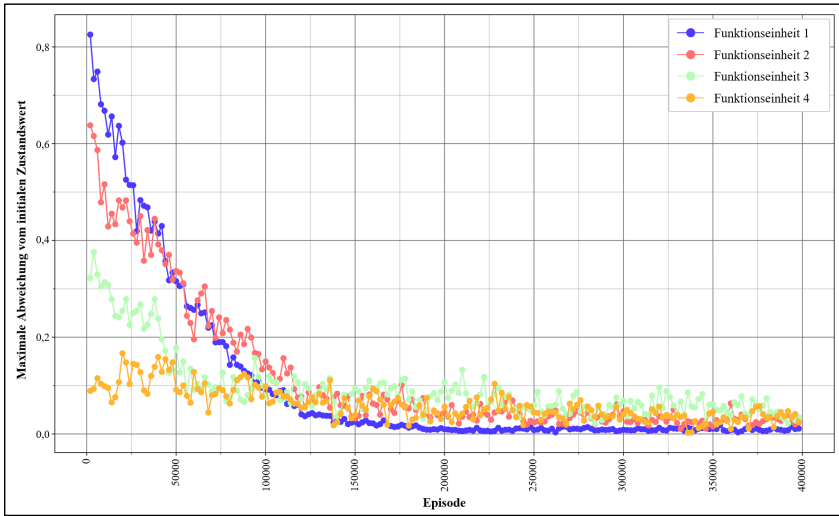


Abbildung 6-4: Ergebnisse der Occlusion-Sensitivitätsanalyse, angewendet auf die Zustandsinformationen der Transportmatrix

Zusätzlich kann zu jedem Trainingszeitpunkt eine Saliency Map hinsichtlich der Bedeutung der Input-Feature analysiert werden. Abbildung 6-5 zeigt die Ergebnisse einer Occlusion-Sensitivitätsanalyse, die auf die Transportwege angewendet wurde. Zur Erstellung werden die Elementarzellen, die als Transportweg klassifiziert sind, schrittweise in Freiflächen umgewandelt und die Reaktion des Agenten bestimmt. Die Analyse wird beispielhaft für den Platziervorgang von Funktionseinheit 3 durchgeführt. Daher wird der Q-Wert und nicht der Zustandswert als Referenz verwendet. Anhand der Abweichungen des Q-Werts kann analysiert werden, ob zusätzlicher freier Platz bevorzugt wird oder nicht. Die Elementarzellen direkt oberhalb der Position von Funktionseinheit 3 weisen die höchsten positiven Abweichungen im Q-Wert auf. Dies bedeutet, dass zusätzlicher Freiraum anstelle der Transportlinie vom Agenten bevorzugt wird und das Vertrauen des Agenten in die Positionierungsentscheidung erhöht. Dieses Ergebnis erscheint nachvollziehbar, da der Agent für die Platzierung der nachfolgenden Funktionseinheit 4 eine verbesserte Platzverfügbarkeit vorfinden würde.

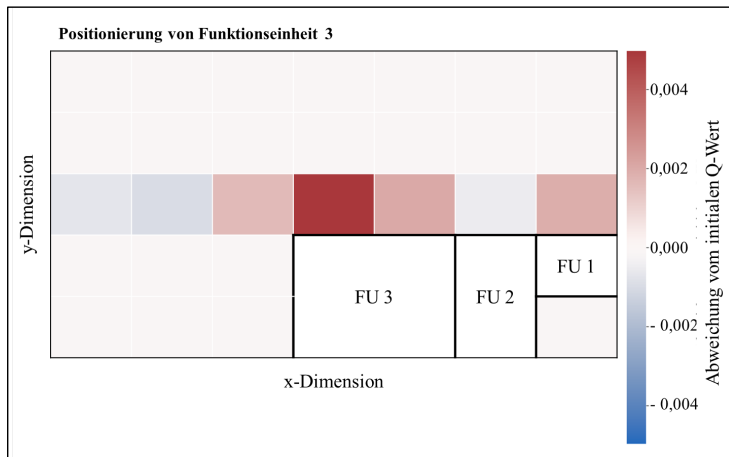


Abbildung 6-5: Detaildarstellung der Ergebnisse der Occlusion-Sensitivitätsanalyse bei Ausblendung der Transportwege

7 Zusammenfassendes Konzept zur Reinforcement Learning basierten Fabriklayoutplanung

In diesem Kapitel wird ein Konzept entwickelt, das alle Schritte zur RL-basierten Layoutplanung zusammenfasst (siehe Abbildung 7-1) und den Ausgangspunkt für den entwickelten Softwaredemonstrator bildet. Das Konzept umfasst 5 Hauptschritte, die in Unterschritte unterteilt sind und durch einen Informationsfluss verknüpft sind.

Im **ersten Schritt** werden die Eigenschaften des Layoutplanungsproblems festgelegt. Dazu wird zunächst der Planungsgrundfall (Neu- oder Umplanung) definiert, aus dem die weiteren benötigten Informationen abgeleitet werden. Anschließend werden die Layouteigenschaften initialisiert. Diese können in erforderliche und optionale Informationen unterteilt werden. Bei einer Neuplanung muss als Mindestinformation der verfügbare Platz für die Planung in 2D vorliegen. Für eine Umplanung wird zusätzlich ein fertig geplantes 2D-Layout benötigt, das den alten Zustand des Layouts beschreibt und als Referenz für die Berechnung des Belohnungswertes dient. Optional können Höheninformationen, Informationen zur Medienverfügbarkeit, über Bodentraglasten und zu belegten Bereichen wie Transportlinien einbezogen werden. Es ist wünschenswert, dass alle optionalen Informationen vorliegen, da nur in diesem Fall alle Zielgrößen in die Optimierung einfließen können. Demnach ist es nicht möglich die Bodentraglast als Zielgröße zu nutzen, wenn keine Informationen zur maximal zulässigen Bodentraglast für jede Elementarzelle vorliegen. Demnach beeinflusst der verfügbare Informationsumfang die Realitätsnähe der Planungsergebnisse.

Die Informationen zu den Funktionseinheiten müssen mit den Layoutinformationen abgestimmt sein. Dementsprechend muss für jede Funktionseinheit mindestens die 2D-Geometrie und optional Höheninformationen oder Anforderungen an Medienverfügbarkeit oder Bodenlasten vorliegen. Darüber hinaus werden die folgenden allgemeinen und prozessbezogenen Informationen zur Initialisierung jeder Funktionseinheit genutzt:

- ☐ Identifikationsnummer oder eindeutiger Name der Funktionseinheit
- ☐ Typ der Funktionseinheit: Maschine oder Lager
- ☐ Maximale vor- und nachgelagerte Lagerkapazität der Funktionseinheit
- ☐ Produktbestand, die der Funktionseinheit zu Beginn der Simulation bereitstehen
- ☐ Informationen über jeden Rüstzustand. Hierzu gehören pro Rüstzustand die Prozess- und Rüstzeiten sowie die Input-Output Beziehung
- ☐ Informationen darüber, ob eine vordefinierte Position im zukünftigen Layout vorgesehen ist oder ob diese frei positioniert werden kann

In einem weiteren Teilschritt werden die Materialflusseigenschaften definiert. Der Materialfluss umfasst die Transportmittel, die externen Zulieferer und PPS-bezogene Informationen. Die Transportmittel werden anhand folgender Informationen initialisiert:

- ☐ Identifikationsnummer oder eindeutiger Name des Transportmittels
- ☐ Typ des Transportmittels (z. B. Fahrerlose Transportfahrzeug oder Gabelstapler)

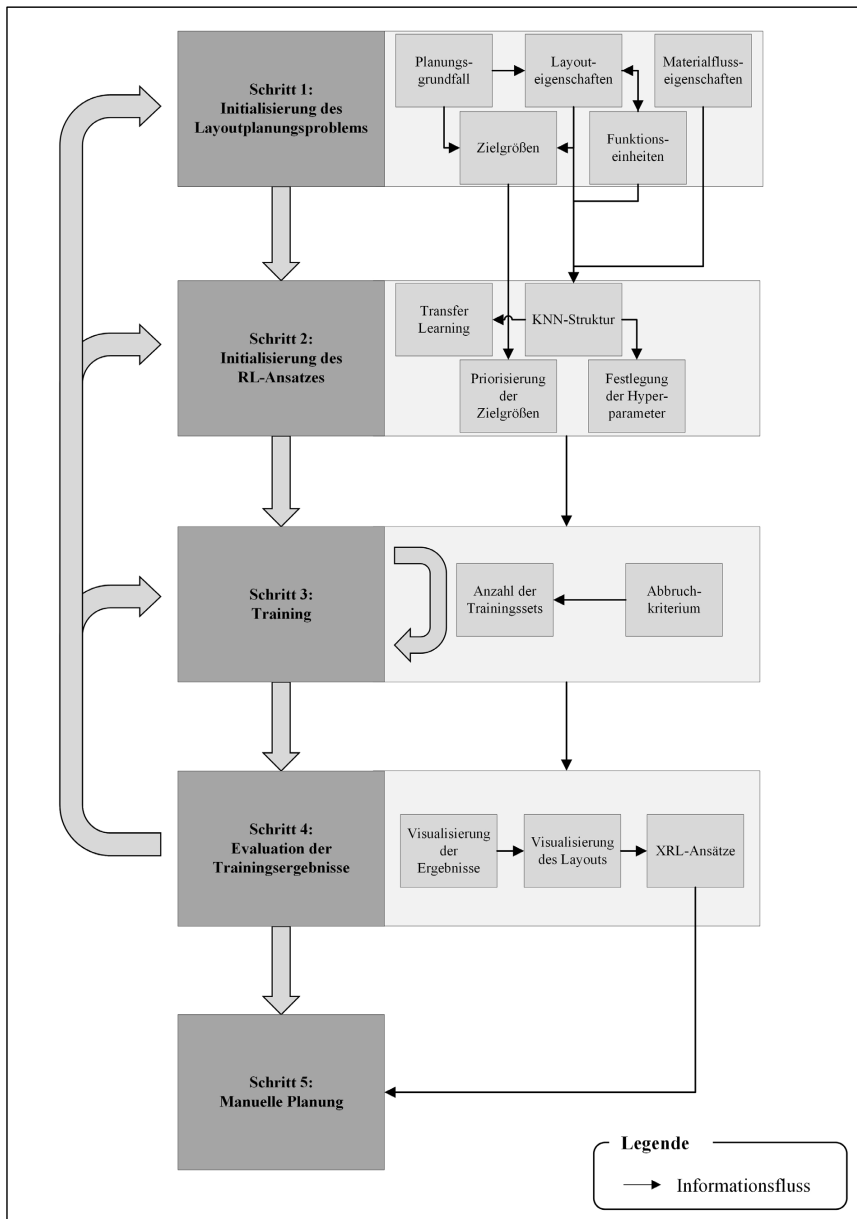


Abbildung 7-1: Konzept zur RL-basierten Fabriklayoutplanung

- ☐ Transportkapazität des Transportmittels
- ☐ Be- und Entladegeschwindigkeiten

□ Startposition zu Beginn der Simulation, falls diese vordefiniert ist

Informationen über externe Zulieferer werden benötigt, um die Materialversorgung ausgewählter Funktionseinheiten (z. B. Lager) zu berücksichtigen. Dabei werden die Menge und die Häufigkeit des zu liefernden Produkts sowie die Funktionseinheit für die Anlieferung definiert.

Die PPS-bezogenen Informationen umfassen das zu simulierende Produktionsprogramm, das durch die Anzahl der Produkte pro Produktvariante und die zugehörigen Produktionsschritte definiert ist. Für jeden Produktionsschritt wird angegeben, welche Funktionseinheit diesen ausführen kann. Mit Hilfe der zu definierenden Reihenfolgeplanungslogik (z. B. FiFo) kann dann vor der Simulation festgelegt werden, welche Funktionseinheit für welchen Prozessschritt verwendet werden soll. Abschließend ist die angestrebte Materialflussteuerungsstrategie, wie eine Pull-Steuerung unter Verwendung von Kanbans, festzulegen.

Abbildung 7-2 zeigt einen Ausschnitt des Softwaredemonstrators für den ersten Schritt. Die zugehörigen Informationen zur Layoutcharakterisierung, den Funktionseinheiten und den Materialflusseigenschaften können aus den zugehörigen Konfigurationsdateien extrahiert werden. Hierfür werden bevorzugt CSV- oder TXT-Dateien verwendet. Nach Festlegung des Dateipfades durch den Anwender werden die zugehörigen Informationen extrahiert und in der grafischen Benutzeroberfläche (GUI) visualisiert.

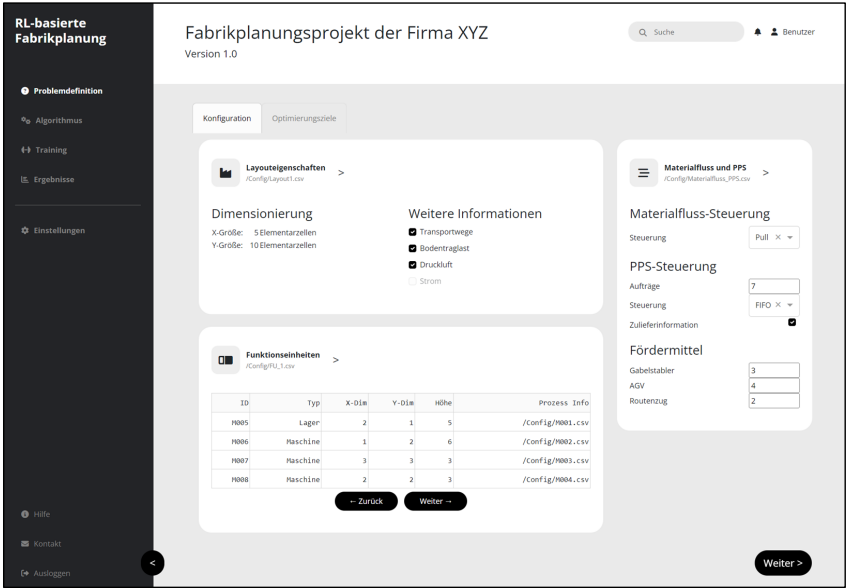


Abbildung 7-2: Visualisierung des Konfigurationsschritt der Problemdefinition

Den Übergang vom ersten zum **zweiten Schritt** bildet die Definition der Zielgrößen und deren Priorisierung. Die Zielgrößen sind modular und weitgehend heterogen definiert, sodass sie problemabhängig in einer beliebigen Kombination ausgewählt werden können. Eine Optimierung der Transportintensität oder der Durchlaufzeit kann jedoch je nach Planungsfall zu vergleichbaren Lösungen führen. Ein Beispiel hierfür ist eine starr verkettete Produktion,

bei der die Transportbänder für den Materialfluss genutzt werden. In diesem Fall ist es zu empfehlen, nur eine der beiden Zielgrößen in die multikriterielle Optimierung einzubeziehen, um eine Überpriorisierung zu vermeiden. Außerdem steht die Zielgröße der Ähnlichkeit nur dann zur Verfügung, wenn die Umplanung als Planungsfall gewählt wurde. In Abbildung 7-3 ist der Softwaredemonstrator für einen Neuplanungsfall visualisiert, daher ist die Ähnlichkeit ausgegraut und nicht anwählbar. Bei einer Umplanung sollte außerdem darauf geachtet werden, dass die Zielgröße der Ähnlichkeit mit mindestens einer weiteren Zielgröße kombiniert wird, da der Agent sonst lediglich versucht, die ursprüngliche Lösung zu rekonstruieren.

Die ausgewählten Zielgrößen werden zur Initialisierung der Belohnungsfunktion verwendet. Dazu ist zusätzlich die Priorisierung der einzelnen Zielgrößen festzulegen. Diese entsprechen den Gewichten w_1 bis w_9 und dürfen in Summe den Wert von 1,0 nicht überschreiten.

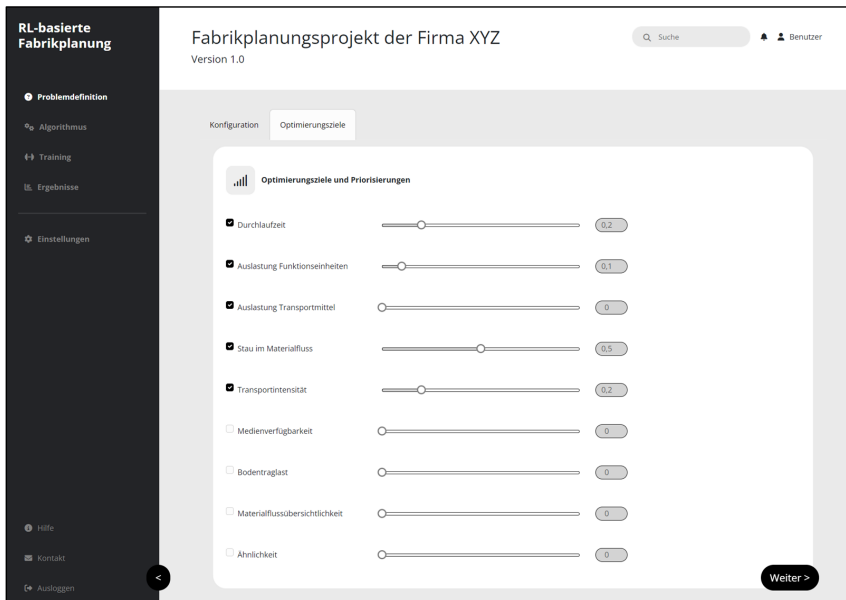


Abbildung 7-3: Visualisierung der Priorisierung der Optimierungsziele

Die weiteren Teilschritte befassen sich mit der Initialisierung des Agenten und der KNN-Struktur. Die Architekturen der Eingangs- und Ausgabeschicht werden durch die Problemcharakterisierung des ersten Schritts definiert. Die Eingabeschicht ist an die verfügbaren Informationen angepasst und besteht aus einem GANN zur Verarbeitung der Materialflussbeziehungen, einem CNN zur Verarbeitung der layoutbezogenen Informationen und einem MLP zur Verarbeitung der funktionseinheitbezogenen Informationen. Die Anzahl der Neuronen in der Ausgabeschicht wird an die Anzahl der Elementarzellen angepasst.

Daher kann nur die Architektur der versteckten Schichten konfiguriert werden (siehe Abbildung 7-4). Hierbei muss die Anzahl der versteckten Schichten, die zugehörige Anzahl der Neuronen und die Aktivierungsfunktion festgelegt werden.

Außerdem kann ein bereits trainiertes RL-Modell ganz oder teilweise geladen werden, sodass der RL-Ansatz nicht von Grund auf neu trainiert wird, sondern Vorwissen aufbaut. Dazu muss die KNN-Architektur der versteckten Schichten auf die KNN-Architektur des vortrainierten KNNs abgestimmt sein.

RL-basierte Fabrikplanung

Fabrikplanungsprojekt der Firma XYZ
Version 1.0

Suche Benutzer

Konfiguration des KNNs

ID	Typ	Neuronen Zahl	Aktivierungsfunktion	Vortrainiert
Hidden_Layer_1	Dense	100	ReLU	X
Hidden_Layer_2	Dense	50	ReLU	X
Hidden_Layer_3	Dense	20	ReLU	X
Hidden_Layer_4	Dense	10	ReLU	

Hinzufügen Reset

Transfer Learning >
/Config/Model35

Modellinfos
Trainingsdauer: 72 Stunden
CPU: 30 GB RAM
GPU: —

Konfiguration Training >
/Config/Training_1.csv

Konfiguration	Werte
Name:	gui_use_case
Anzahl Steps:	2000000
Buffer Größe:	1000000
Epsilon:	0.8
Zuf. Transportmatr.:	false

Hilfe Kontakt Ausloggen

Weiter >

Abbildung 7-4: Visualisierung der Algorithmuskonfiguration

Im **dritten Schritt** wird das Training initialisiert und durchgeführt. Für jede Priorisierung muss dieser individuell durchlaufen werden. In einer Konfigurationsdatei wird festgelegt, wie viele Trainingsepisoden durchgeführt werden sollen und nach welcher maximalen Trainingsdauer das Training abgebrochen wird. Zusätzlich können die Parameter der Explorationsstrategie definiert werden, falls von den Standardparametern (siehe Kapitel 5.2) abgewichen wird. Während des Trainings werden alle Transitionen und Verbesserungen der aktuell besten Lösung gespeichert. Außerdem wird das Modell des Agenten regelmäßig gespeichert, sodass es für zukünftige Analysen oder Transfer Learning genutzt werden kann. Zusätzlich kann der Trainingsfortschritt anhand der durchschnittlichen Belohnung, der Bewertung des besten Layouts und des durchschnittlichen Verlustwerts beobachtet werden. Abbildung 7-5 stellt die Visualisierung des Trainingsschritts im Softwaredemonstrator dar.

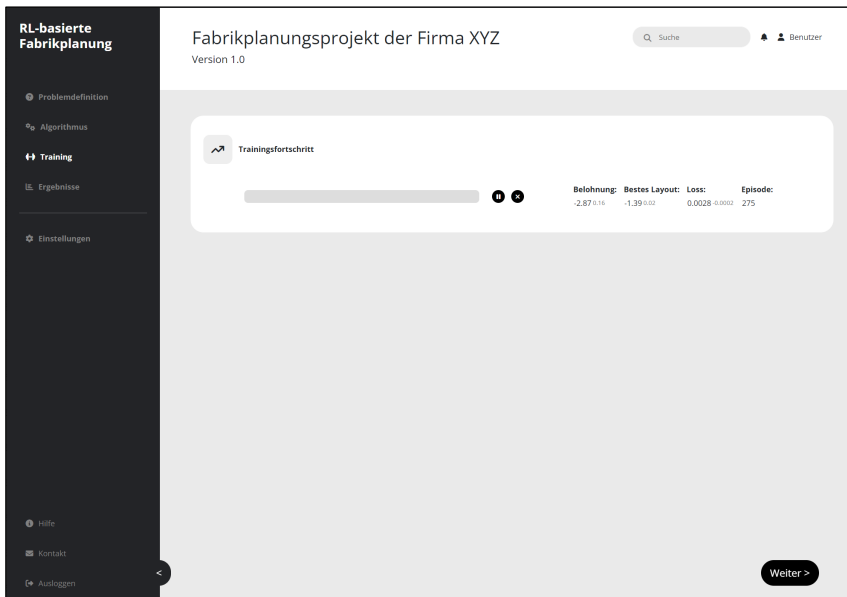


Abbildung 7-5: Visualisierung des Trainingsschritts

Nach dem Training werden die Ergebnisse im **vierten Schritt** ausgewertet, um ein oder mehrere Layouts für den manuellen Planungsschritt auszuwählen. Die temporär besten Lösungen aller Trainingsdurchläufe können in Form eines Diagramms zusammengefasst werden, dessen beste Lösungen eine Pareto-Front aufspannen (siehe Abbildung 7-6). Die Pareto-Front stellt nicht nur den Lösungsraum dar, sondern hilft den Planenden auch bei der Analyse von Zielkonflikten zwischen einzelnen Zielgrößen. Jeder Punkt der Pareto-Front repräsentiert ein nicht dominiertes Layout, das für den manuellen Planungsschritt relevant sein kann. Zur Evaluation können das Layout und die zugehörige Bewertung für alle Zielgrößen angezeigt werden. Außerdem können die XRL-Methoden verwendet werden, um die Vertrauenswürdigkeit der Ergebnisse im Detail zu evaluieren (siehe Kapitel 6). Für den Fall, dass kein vertrauenswürdiges Layout generiert wurde oder die Layoutvarianten nicht den Vorstellungen der Planenden entsprechen, können die vorangegangenen Schritte wiederholt werden. U. a. kann eine Änderung der KNN-Architektur das Trainingsverhalten verändern und wahrscheinlich zur Generierung neuer Layoutvarianten führen. Alternativ kann die Verlängerung der Trainingsdauer oder die Nutzung von Transfer Learning die Ergebnisqualität steigern. Schließlich werden ein oder mehrere Layouts unter Berücksichtigung der Planungspräferenzen und der strategischen Zielgrößen ausgewählt und in den manuellen **fünften Planungsschritt** überführt.

Dieser entspricht der Detailplanung der bestehenden Planungsansätze. Dabei werden die Positionen der Funktionseinheiten endgültig festgelegt und Detailspekte wie Beleuchtung, Emissionen, Ergonomie- und Arbeitssicherheitsaspekte für alle Arbeitsstationen geplant. Der Detailplanungsschritt kann durch Technologien wie AR und VR unterstützt werden, um eine realitätsnahe Planung der Arbeitsbereiche zu ermöglichen.

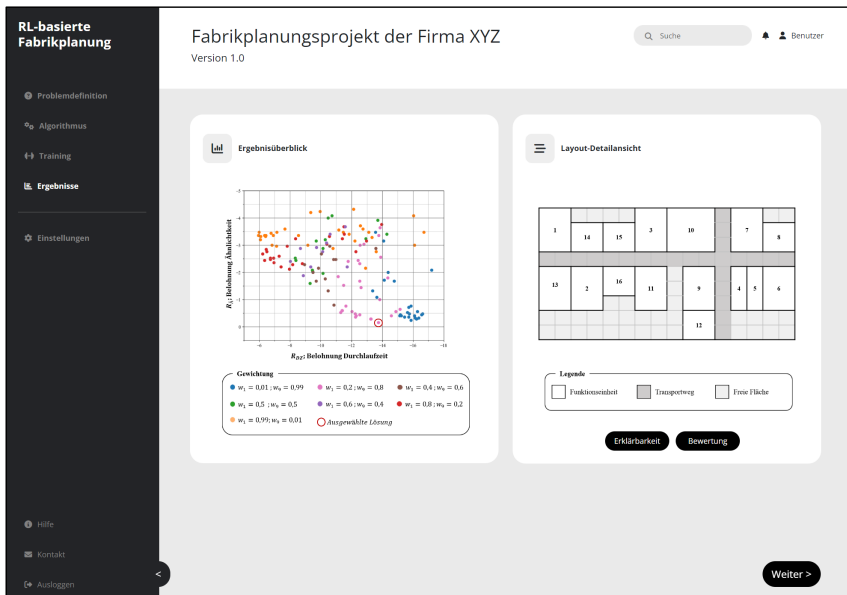


Abbildung 7-6: Visualisierung der Ergebnisevaluation

8 Validierung

Ziel dieses Kapitels ist es, den Nutzen und die Praxistauglichkeit des entwickelten Ansatzes anhand industrieller Praxisbeispiele zu validieren. Hierfür werden ein Umplanungsszenario für die Fertigung eines Stahlprobenherstellers (Kapitel 8.1) und ein Neuplanungsszenario eines Nutzfahrzeugherstellers (Kapitel 8.2) betrachtet. Die Daten der Anwendungsszenarien stammen aus Industrieberatungsprojekten, die am FBK durchgeführt wurden, und wurden abstrahiert, um eine Veröffentlichung zu ermöglichen.

8.1 Umplanungsszenario

Schritt 1: Initialisierung des Layoutplanungsproblems

Das erste Anwendungsbeispiel ist die Umplanung des Layouts eines Stahlprobenherstellers. Die Geometrie des Layouts wird mit 34 Positionen in x-Richtung und 7 Positionen in y-Richtung modelliert (siehe Abbildung 8-1). Das bestehende Layout enthält einen vordefinierten Transportweg, der U-förmig durch das Layout verläuft und auch im neuen Layout beibehalten werden soll. Außerdem ist der rechte Bereich aufgrund einer dickeren Wand schmaler als der linke Bereich des Layouts. Für das bestehende Layout und die Funktionseinheiten liegen keine Informationen über die Höhe und die benötigten Medien und Bodenlasten vor. Eine Optimierung des Layouts hinsichtlich Bodenlasten und Medienverfügbarkeit kann daher nicht durchgeführt werden.

Für die Herstellung der Stahlproben werden 43 Funktionseinheiten verwendet. Die Größe der Funktionseinheiten wird in Elementarzellen überführt und ist in Tabelle 8-1 zusammengefasst. Die prozessbezogenen Kenngrößen (u. a. Rüstzeiten und Prozesszeiten) können aus den historischen Produktionsdaten, die über einen Zeitraum von einem Jahr aufgezeichnet wurden, extrahiert werden. Diese Daten werden für die Festlegung der Materialfluss- und PPS-Eigenschaften genutzt. Eine Analyse der Daten zeigt, dass die Ausbringungsmenge maßgeblich durch 12 Produktvarianten beeinflusst wird. Diese werden zur Initialisierung der Simulation verwendet. Die Anzahl der in einem Jahr hergestellten Produkte wird reduziert, um die Simulationszeit zu verkürzen. Dabei wird darauf geachtet, dass die relativen Mengenunterschiede zwischen den Produktvarianten bestehen bleiben. Der Materialtransport wird durch vier Gabelstapler durchgeführt, die anhand einer Push-Strategie gesteuert werden. Außerdem wird die Reihenfolgeplanung mittels FiFo-Logik initialisiert.

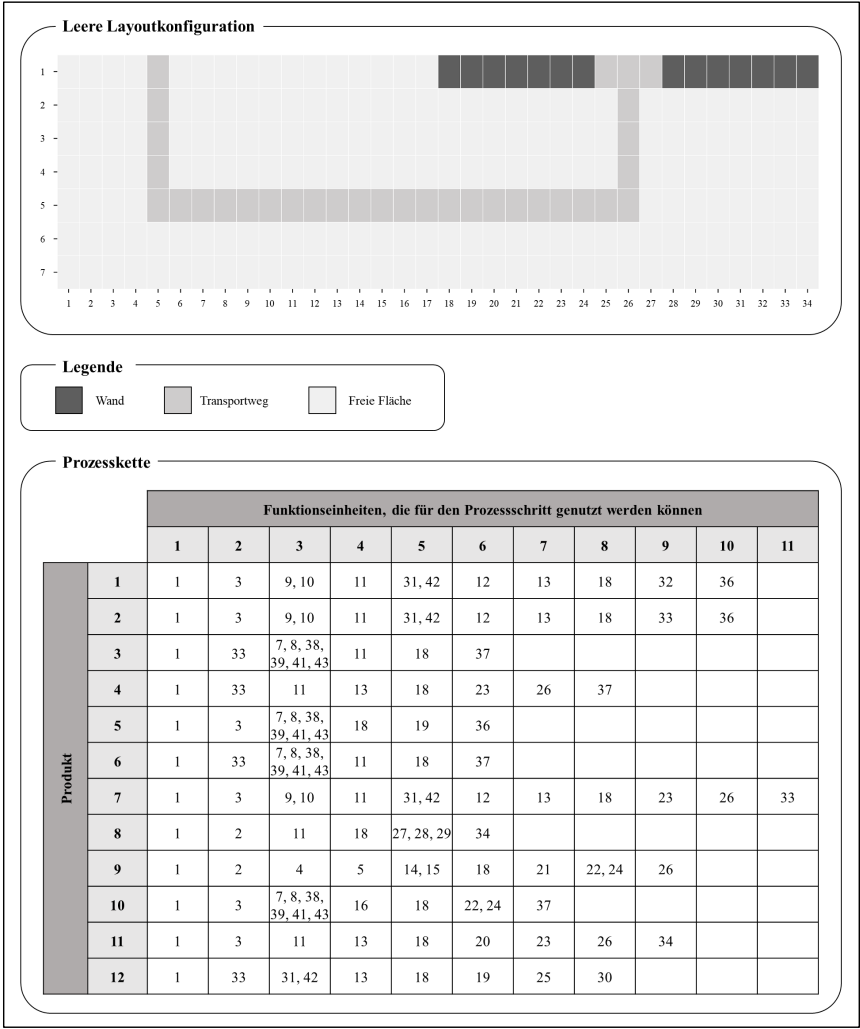


Abbildung 8-1: Industrielles Umplanungsszenario

Tabelle 8-1: Größe der Funktionseinheiten in Elementarzellen

Funktionseinheit	1	2	3	4	5	6	7	8	9	10	11
x-Dimension	3	1	3	2	1	2	1	1	1	1	1
y-Dimension	2	2	2	3	1	3	1	1	1	1	2

Funktionseinheit	12	13	14	15	16	17	18	19	20	21	22
x-Dimension	1	2	1	1	3	1	2	1	2	1	1
y-Dimension	2	3	1	1	3	1	2	1	2	2	1

Funktionseinheit	23	24	25	26	27	28	29	30	31	32	33
x-Dimension	1	1	1	1	1	1	1	2	1	2	1
y-Dimension	1	1	1	3	2	2	2	1	1	2	3

Funktionseinheit	34	35	36	37	38	39	40	41	42	43
x-Dimension	2	1	1	1	1	1	1	1	1	1
y-Dimension	1	2	1	2	1	1	1	1	2	2

Schritt 2: Initialisierung des RL-Ansatzes

Der Übergang zwischen Schritt 1 und 2 erfolgt durch die Definition und Priorisierung der Zielgrößen für die Optimierung. Für das Anwendungsszenario werden die Durchlaufzeit und die Übersichtlichkeit des Materialflusses als die beiden zentralen Zielgrößen gewählt. Es sollen verschiedene Layoutvarianten erzeugt werden, sodass die manuell durchzuführende Detailplanung auf verschiedene Layoutvarianten zurückgreifen kann. Daher werden die Priorisierungsgewichte der Durchlaufzeit (w_1) und der Übersichtlichkeit des Materialflusses (w_8) gemäß Tabelle 8-2 systematisch variiert, während alle anderen Gewichte den Wert von Null annehmen. Als Referenz für die Bewertung der Lösungsqualität sind die Versuche 1 und 9 zu betrachten, da hier nur eine einzelkriterielle Optimierung durchgeführt wird.

Anschließend wird für jeden Versuch ein eigenständiger RL-Agent entsprechend der Standardparameter aus Kapitel 5.2 initialisiert. Jeder Agent wird ohne Vorwissen trainiert, da nicht davon ausgegangen werden kann, dass ein Unternehmen bereits ein trainiertes RL-Modell besitzt.

Tabelle 8-2: Übersicht über die Priorisierungsgewichte

Versuch	Durchlaufzeit w_1	Übersichtlichkeit w_8
1	1,00	0,00
2	0,99	0,01
3	0,80	0,20
4	0,60	0,40
5	0,50	0,50
6	0,40	0,60
7	0,20	0,80
8	0,01	0,99
9	0,00	1,00

Schritt 3: Training

Die Trainingsdauer wird für jeden der 9 Agenten auf maximal 72 Stunden unter Verwendung von 30 GB RAM eines Intel Xeon E5-2670 Prozessors begrenzt. Während des Trainings werden alle Transitionen und die besten Layouts gespeichert. Zusätzlich werden die KNN-Modelle der RL-Agenten alle 6000 Transitionen extrahiert, sodass eine Bewertung mittels der XRL-Methoden durchgeführt werden kann.

Schritt 4: Evaluation der Trainingsergebnisse

Die Ergebnisse können nach dem Training aggregiert und visualisiert werden. Zur besseren Evaluation der Leistungsfähigkeit werden weitere Referenzgrößen eingeführt. Dazu werden die Positionen der Funktionseinheiten im *bestehenden, manuell geplanten Layout* des Stahlprobenherstellers in Aktionen umgewandelt, sodass die Belohnungswerte für jedes Optimierungsziel bestimmt werden können.

Eigene vorangegangene Untersuchungen haben gezeigt, dass die *Kombination von GA und TS* zu den leistungsfähigsten Metaheuristiken für das FLP gehört [Klar23a, Klar24d]. Der GA fungiert als globale Suchheuristik, während die TS als lokale Suchheuristik genutzt wird. Im Detail werden für den GA n_{GA} Lösungen generiert. Zwei der besten $n_{GA,P}$ Lösungen werden als Ausgangspunkt für die Generierung von $n_{GA} - n_{GA,P}$ neuen Lösungen verwendet. Der hierfür eingesetzte Crossover Operator wird für jede neue Lösung an einer zufälligen Stelle platziert, sodass die besten Lösungen zu unterschiedlichen Anteilen in die neuen Lösungen einfließen. Anschließend werden n_{mut} neue Lösungen zufällig durch den Mutationsoperator verändert. Eine Mutation kann auf drei unterschiedliche Arten umgesetzt werden.

1. Eine Funktionseinheit wird einer neuen, zufällig gewählten Position zugeordnet
2. Die Position von zwei Funktionseinheiten wird getauscht
3. Eine Funktionseinheit wird auf eine benachbarte Position verschoben

Die TS wird auf die beste Lösung des GA angewendet, um diese innerhalb von N_{TS} Iterationen weiter zu optimieren. Eine benachbarte Lösung wird generiert, indem die gleichen Mutationsoperatoren genutzt werden wie zuvor. Jede berechnete Lösung wird der Tabu Liste mit der Länge $N_{TS,list}$ hinzugefügt. Die Tabu Liste ist eine fortlaufende Liste, sodass sie die Lösungen mehrerer TS-Durchläufe enthält. Nachdem die Lösung mittels TS optimiert wurde, wird die schlechteste Lösung des GA ersetzt, bevor eine neue GA-Iteration durchgeführt wird. Die Hyperparameter der Metaheuristik ($n_{GA} = 20$, $n_{GA,P} = 4$, $n_{mut} = 5$, $N_{TS} = 200$, $N_{TS,list} = 250$) wurden in der Vorstudie vollfaktoriell optimiert, um einen leistungsfähigen Vergleichsansatz zu gewährleisten [Klar23a, Klar24d]. Der integrierte Ansatz aus GA und TS wird entsprechend der Versuche 2 bis 8 aus Tabelle 8-2 initialisiert. Für die Berechnung wird die gleiche Hardwarekonfiguration genutzt wie für den RL-Ansatz.

Zusätzlich wird ein *NSGA-Ansatz*, der auf der zuvor beschriebenen Funktionsweise und den Hyperparametern des GA basiert, als Referenz verwendet. Der NSGA-Ansatz ermöglicht im Vergleich zu den zuvor beschriebenen Ansätzen die direkte Berechnung einer Pareto-Front. Daher wird nur eine Instanz des NSGA initialisiert.

Abbildung 8-2 fasst die Ergebnisse der Versuche nach 1, 24 und 72 Stunden zusammen. Die beiden gestrichelten Linien stellen die Ergebnisse der einzelkriteriellen Optimierung (Versuche 1 und 9) nach 72 Stunden dar und gelten als Referenzwerte für die Beurteilung der Leistungsfähigkeit.

Alle automatisierten Ansätze generieren bereits nach einer Stunde bessere Ergebnisse als das Ausgangslayout. Darüber hinaus ist zu beobachten, dass die meisten durch Metaheuristiken berechneten Layouts eine bessere Ergebnisqualität aufweisen als die durch RL generierten Layouts (10 % geringere Durchlaufzeit und 4 % bessere Materialflussübersichtlichkeit beim Vergleich der besten Lösungen in jeder Dimension). Nach 24 Stunden kann gezeigt werden, dass die Qualität aller Lösungen weiter gesteigert wurde. Die Metaheuristiken erzeugen in der Dimension Durchlaufzeit bessere Ergebnisse (8 % kürzere Durchlaufzeit), während die RL-Ansätze in der zweiten Dimension bessere Lösungen generieren (30 % bessere Materialflussübersichtlichkeit). Am Ende des Trainings (72 Stunden) verbessert der RL-Ansatz das bestehende Layout um bis zu 26 % hinsichtlich der Durchlaufzeit und um bis zu 78 % in der Übersichtlichkeit des Materialflusses. Ein möglicher Grund für diesen großen Unterschied könnte die veraltete Struktur des bestehenden Layouts sein, die in den letzten Jahren nicht an die veränderten Produktionsmengen angepasst wurde. Im Vergleich zu den Metaheuristiken ergeben sich nach 72 Stunden vergleichbare Ergebnisse hinsichtlich der Durchlaufzeit (1 % kürzere Durchlaufzeit bei Einsatz von RL) und eine Verbesserung in der Übersichtlichkeit des Materialflusses um bis zu 34 %.

In beiden Fällen wird die integrierte Version von GA und TS als Vergleichsbasis herangezogen, da die Ergebnisse des NSGA schlechter sind. Im Vergleich zur einzelkriteriellen Optimierung erreicht nur ein generiertes Layout den Bestwert in der Übersichtlichkeit des Materialflusses. Ein möglicher Grund für die Überlegenheit des RL-Ansatzes nach 72 Stunden liegt in der Fähigkeit von RL, die Eigenschaften des zugrundeliegenden Problems zu erlernen. Konventionelle Metaheuristiken nutzen hingegen eine systematische, jedoch zufallsgeprägte Vorgehensweise zur Berechnung von Lösungen. Das bedeutet, dass die Ansätze unterschiedlich auf zufällig gewählte Aktionen oder Positionierungen einer Funktionseinheit reagieren. Für den Fall, dass eine Funktionseinheit zufällig in einer Ecke des Layouts platziert wird, reagiert der RL-Ansatz entsprechend und positioniert alle folgenden Funktionseinheiten so, dass eine bestmögliche Gesamtlösung realisiert wird. Für die Metaheuristik kann diese Anpassungsfähigkeit jedoch nicht beobachtet werden, da kein Wissen über die Problemmerkmale zur Berechnung der Lösung genutzt wird, weshalb alle verbleibenden Funktionseinheiten an der gleichen Position wie zuvor positioniert werden. Daraus kann gefolgert werden, dass RL lokale Minima besser überwinden kann als konventionelle Metaheuristiken, da dies eine Neupositionierung von mehr als einer Funktionseinheit erfordert. Das Erlernen der Interdependenzen des Problems erfordert jedoch Zeit, da explorative Aktionen ausgewählt werden müssen, um zielführende von nicht zielführenden Aktionen zu unterscheiden. Dieses explorative Verhalten zu Beginn des Trainings kann erklären, warum die Metaheuristiken bei kurzen Rechenzeiten bessere Ergebnisse erzielen.

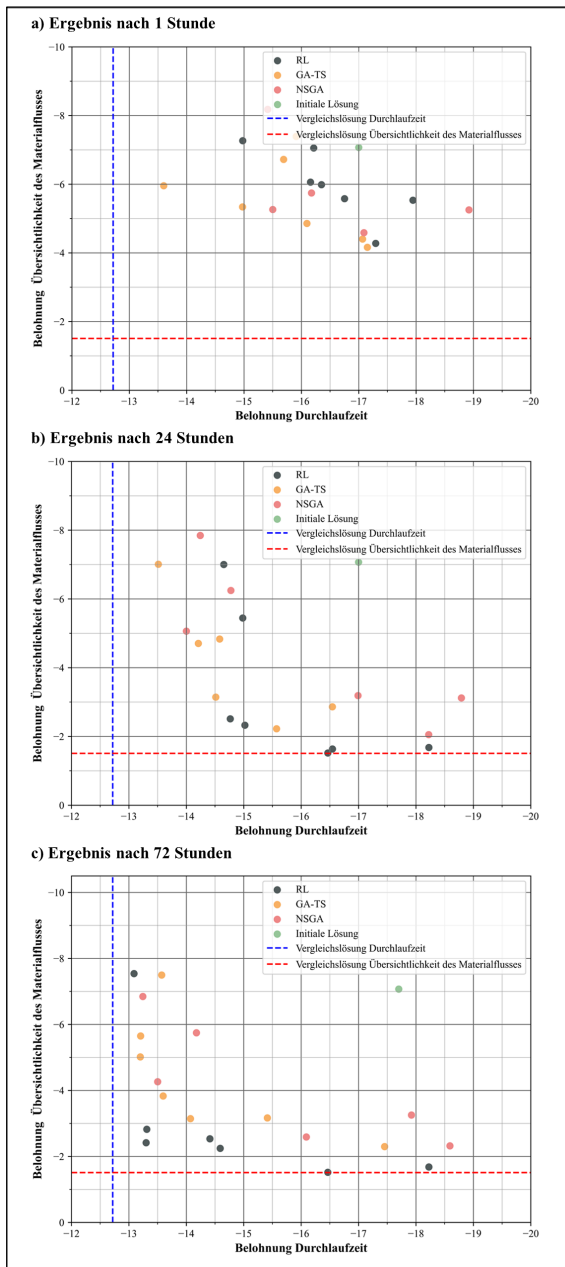


Abbildung 8-2: Ergebniszusammenfassung nach 1 Stunde (a), 24 Stunden (b) und 72 Stunden (c)

In einem weiteren Teilschritt werden die in Kapitel 6 entwickelten *XRL-Ansätze* zur Analyse eines ausgewählten Layouts genutzt. Im Rahmen der Validierung wird hierfür beispielhaft das in Abbildung 8-3 visualisierte Layout analysiert.

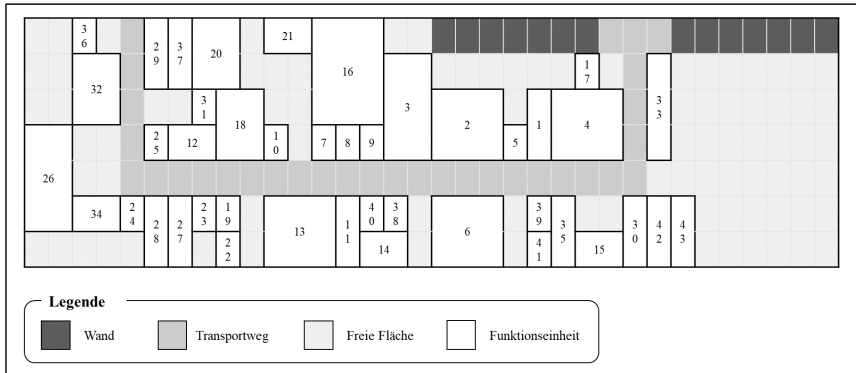


Abbildung 8-3: Layout, das mittels XRL analysiert wird

Zur Untersuchung des primären XRL-Ziels wird zunächst der Verlauf der Belohnungskurve analysiert. Die Kurve sollte zu Beginn des Trainings ansteigen und im Verlauf des Trainings zu einem konstanten Belohnungswert konvergieren. Der zugehörige Graph ist in Abbildung 8-4 dargestellt. Der Verlauf der Kurve entspricht dem erwarteten Verhalten, da die gleitende Durchschnittsbelohnung in den ersten 35000 Episoden steil ansteigt und dann konvergiert. Da dieses Lernverhalten eine erste, notwendige Beobachtung für einen vertrauenswürdigen Lernprozess darstellt, können nachfolgend die entwickelten XRL-Ansätze angewendet werden.

Zu Beginn wird der Policy Summarization Ansatz zur Erstellung der Verteilungs- und Übereinstimmungsanalyse genutzt. Hierfür wird der Parameter $k = 10$ gewählt. Das bedeutet, dass die 10 besten Layoutvarianten, die vom RL-Agenten erstellt wurden, für die PSL und PSFU berücksichtigt werden. Zur Evaluation des Trainingsverhaltens werden drei Analysezeitpunkte festgelegt, die es ermöglichen, das zeitliche Verhalten des RL-Ansatzes zu bewerten. Die erste Analyse wird nach 1000 Episoden, die zweite Analyse nach 15000 Episoden und die letzte Analyse nach 45000 Episoden durchgeführt. Nachfolgend werden die Funktionseinheiten 3 und 38 als exemplarische Analyseobjekte herangezogen. Grundsätzlich sind die Analysen in analoger Vorgehensweise für alle Funktionseinheiten anwendbar.

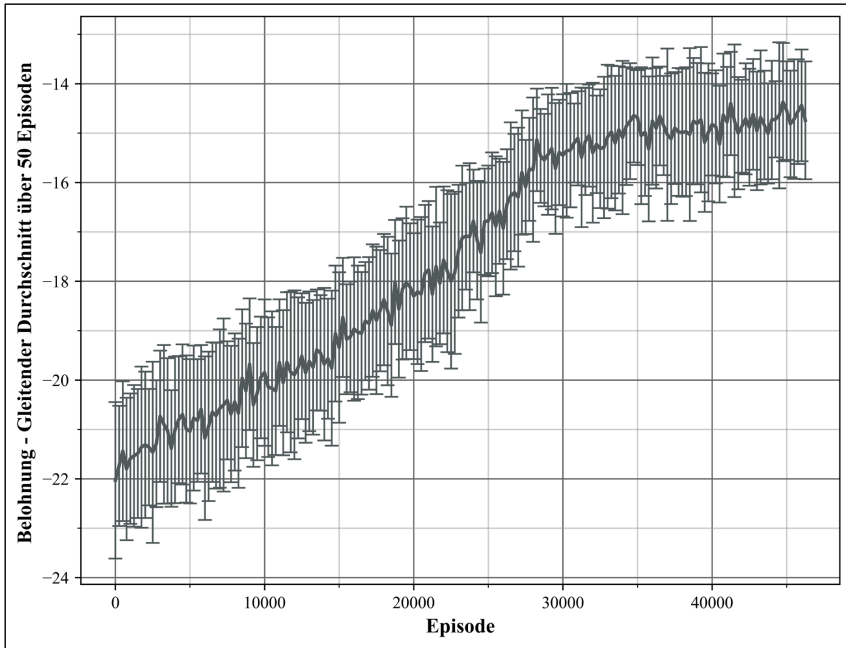


Abbildung 8-4: Gleitender Belohnungsdurchschnitt

Die Ergebnisse der Verteilungsanalyse sind in Abbildung 8-5 dargestellt. Es ist zu erkennen, dass die Streuung in der Positionierung mit zunehmender Trainingsdauer abnimmt. Nach 1000 Episoden werden beide Funktionseinheiten an zufälligen Positionen im Layout platziert, die Streuung in der Positionierung ist entsprechend groß. Nach 15000 Episoden ist für Funktionseinheit 3 bereits eine konzentrierte Positionierung zu beobachten, die sich bis Episode 45000 weiter stabilisiert. Für Funktionseinheit 38 kann nach 15000 Episoden noch keine Verbesserung beobachtet werden. Diese wird erst nach 45000 deutlich, jedoch schwächer ausgeprägt als bei Funktionseinheit 3. Daher ist davon auszugehen, dass die Lösungen nach 1000 und 15000 Trainingsepisoden noch teilweise durch zufälliges Verhalten des RL-Agenten beeinflusst werden, während die Ergebnisse nach 45000 Episoden, aufgrund der vorherrschenden Positionierungskonzentration, ein Ergebnis der erlernten Strategie darstellen.

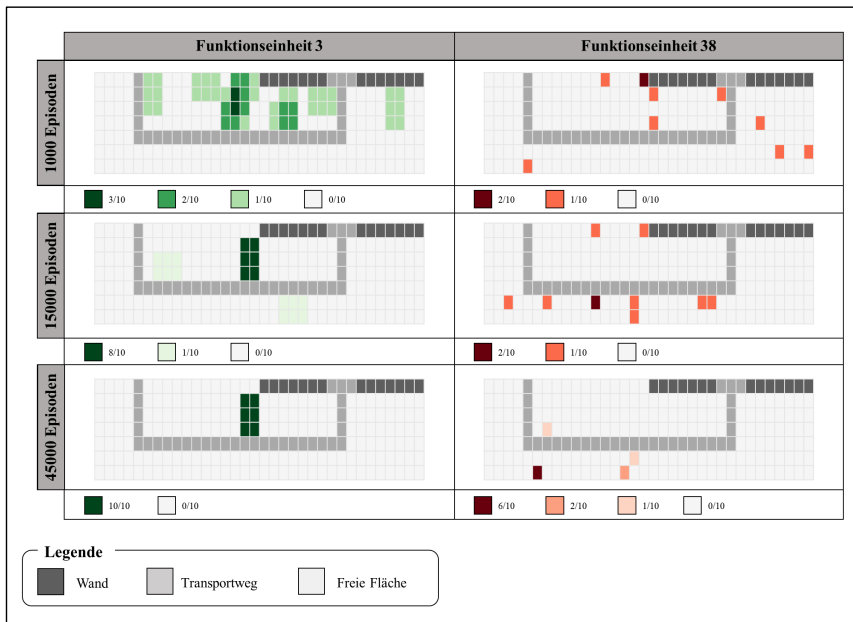


Abbildung 8-5: Ergebnisse der Verteilungsanalyse für die Funktionseinheiten 3 und 38 nach 1000, 15000, und 45000 Episoden

Die Ergebnisse der Übereinstimmungsanalyse, die angibt, ob die beste funktionseinheitbezogene Positionierung auch zur Generierung der besten Layoutvarianten genutzt wurde, unterstützen diese Schlussfolgerung (siehe Abbildung 8-6). Eine fehlende Übereinstimmung am Ende des Trainings kann ein Zeichen für eine unzuweckmäßig gewählte Belohnungsfunktion oder fehlende Fähigkeit zur Überwindung lokaler Minima sein. Es wird erwartet, dass die Übereinstimmungsanalyse zu einer klaren Struktur konvergiert. Diese muss jedoch keine vollständige Übereinstimmung aufweisen, da Zielkonflikte in der Positionierung vorliegen und daher erwartet wird, dass nicht jede Funktionseinheit optimal platziert werden kann.

Die Übereinstimmungsanalyse nach 1000 Episoden zeigt fast keine Übereinstimmungen, was auf ein überwiegend zufälliges Platzierungsverhalten des Agenten schließen lässt. Mit zunehmender Trainingsdauer sind bessere Ergebnisse zu beobachten, da der Anteil der Übereinstimmungen zunimmt. Dies kann als Zeichen der Vertrauenswürdigkeit angesehen werden, da die Übereinstimmungsanalyse von einem unstrukturierten zu einem überwiegend strukturierten Zustand übergeht. Dennoch kann keine vollständige Übereinstimmung festgestellt werden, was vermutlich auf die Problemkomplexität zurückzuführen ist. Aufgrund der begrenzten zur Verfügung stehenden Fläche und der komplexen Materialflussbeziehungen treten Positionierungskonflikte auf, die auch mit zunehmender Trainingszeit nicht überwunden werden können.

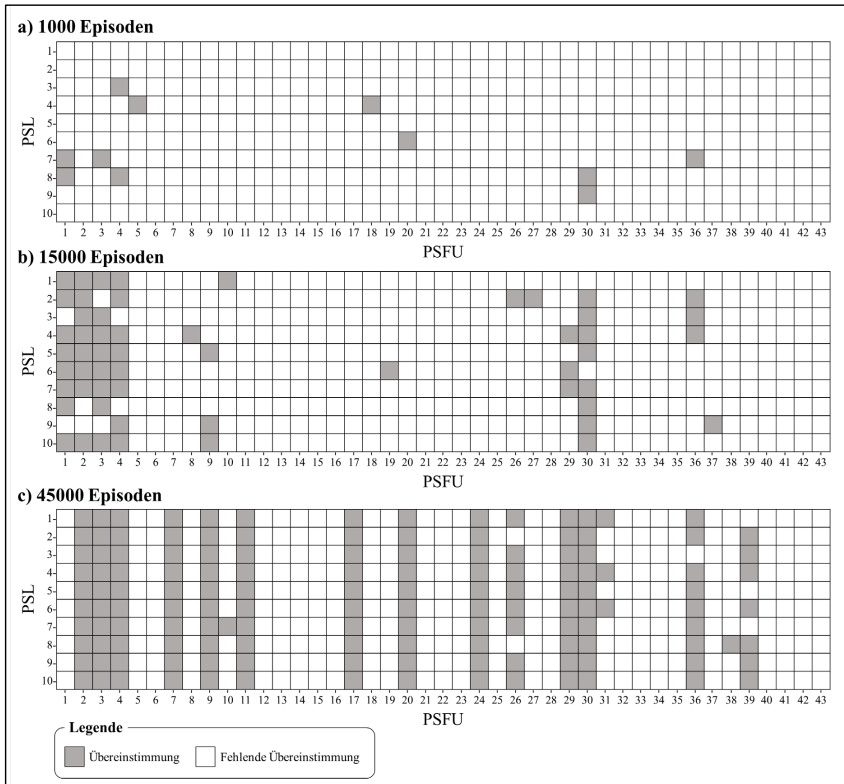


Abbildung 8-6: Übereinstimmungsanalyse nach 1000 (a), 15000 (b) und 45000 Episoden (c)

Die Vertrauenswürdigkeit der Ergebnisse kann weiterhin anhand der Ergebnisse der Occlusion-Sensitivitätsanalyse beurteilt werden. Die KNN-Modelle des RL-Agenten werden alle 6000 Transitionen gespeichert und verwendet, um die Occlusion-Sensitivitätsanalyse für die Transportmatrix durchzuführen. Die Ergebnisse sind in Abbildung 8-7 dargestellt. Es ist zu erkennen, dass beide Kurven mit zunehmender Episodenzahl gegen 0 konvergieren. Die Unterschiede zwischen den beiden Kurven sind relativ gering. Folglich kann auch aus den Ergebnissen der Occlusion-Sensitivitätsanalyse geschlossen werden, dass der Agent am Ende des Trainings eine vertrauenswürdige Strategie erlernt hat und die generierten Layoutvarianten ein Ergebnis dieser Strategie sind und nicht rein zufällig generiert werden.

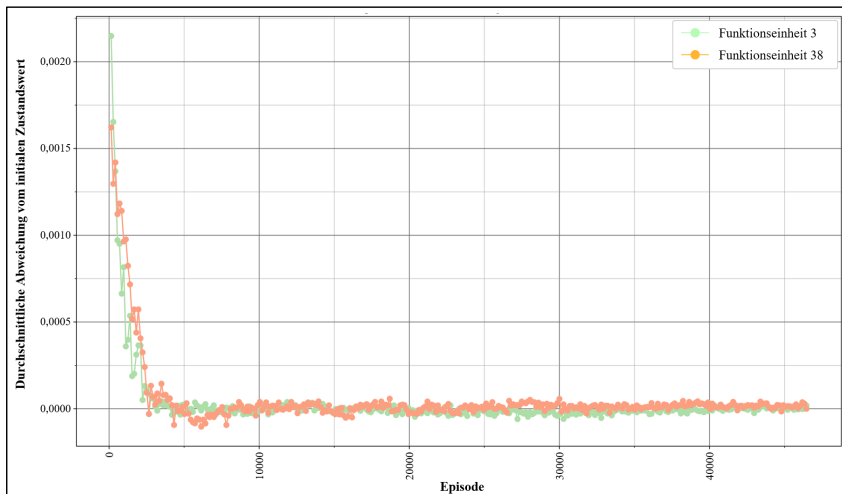


Abbildung 8-7: Ergebnisse der Occlusion-Sensitivitätsanalyse für Funktionseinheit 3 und 38

Aufbauend auf diesen Erkenntnissen können die verbleibenden XRL-Methoden genutzt werden, um eine Detailanalyse der Eingangsinformationen für das endgültige RL-Modell durchzuführen. Die Interpretation der Ergebnisse ist jedoch im Vergleich zur vorangegangenen Analyse anspruchsvoller und kann daher nur von Planenden mit Kenntnissen im Bereich der Layoutplanung und RL durchgeführt werden.

Abbildung 8-8 zeigt eine Saliency Map, die durch Anwendung der Occlusion-Sensitivitätsanalyse auf die Wandelemente des Layouts erstellt wird. Der Zustandswert wird als Referenzgröße genutzt, um die Auswirkung der Elementarzellenänderung vom Zustand „Wand“ zu „Frei“ zu quantifizieren. Positive Abweichungen symbolisieren, dass der Agent ein Layout mit zusätzlichem Freiraum als wertiger als zuvor einschätzt. Die Ergebnisse zeigen, dass der Agent diese Zustandsänderung überwiegend als wünschenswert einschätzt, da die Abweichungsdifferenz in der Regel positive Werte annimmt. Ein Vergleich der Ergebnisse für Funktionseinheit 3 und 38 zeigt, dass zusätzlicher, freier Platz als wichtiger angesehen wird, wenn das Layout bereits stärker gefüllt ist, da die Abweichungswerte ansteigen.

Es muss jedoch darauf hingewiesen werden, dass die Interpretation der Ergebnisse nicht immer so eindeutig ist wie im vorangegangenen Fall. Zur Verdeutlichung zeigt Abbildung 8-9 die Ergebnisse der Occlusion-Sensitivitätsanalyse für die freien Flächen bei der Positionierung von Funktionseinheit 38. In diesem Fall werden freie Elementarzellen schrittweise in den Zustand „Wand“ versetzt. Die Ergebnisse zeigen kein eindeutiges Muster, da für einige Elementarzellen eine Änderung als vorteilhaft und für andere Bereiche als nachteilig bewertet wird. Aus Sicht der Planenden kann das Ergebnis dennoch von Interesse sein, da anhand der absoluten Sensitivitätswerte die Relevanz einzelner Bereiche abgeschätzt werden kann. In Abbildung 8-9 sind die drei Bereiche mit den höchsten absoluten Sensitivitätswerten orange markiert. Die Information über die relevantesten Bereiche können im anschließenden, manuellen Planungsschritt in die Detailplanung einfließen. Mit den vorgestellten Ansätzen ist es jedoch nicht möglich, zu erklären, warum einige Bereiche im Layout als relevanter bewertet werden.

Zudem ist davon auszugehen, dass die Komplexität der Interpretation mit der Komplexität der Problemstellung zunimmt.

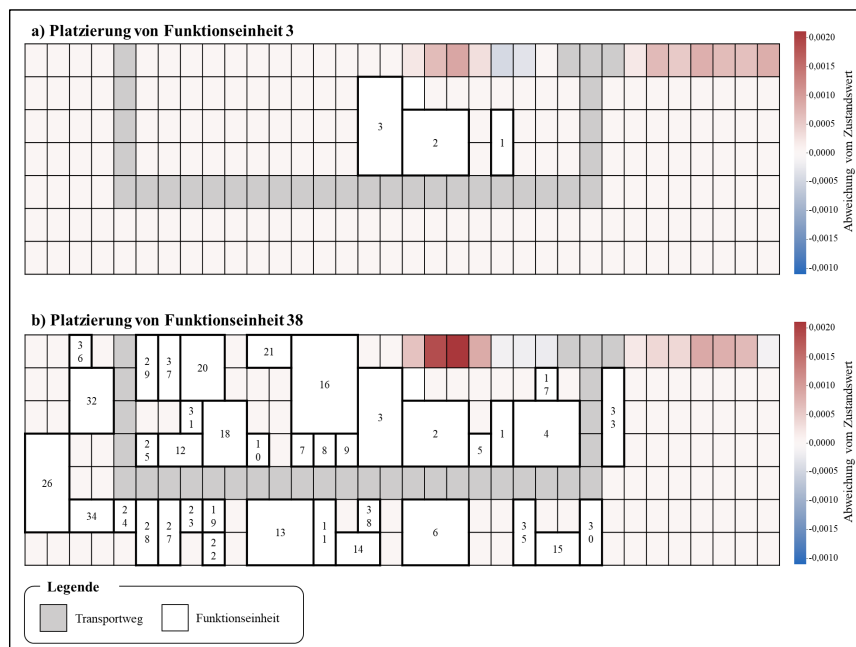


Abbildung 8-8: Saliency Map der Occlusion-Sensitivitätsanalyse der Wand für den Positionierungsprozess von Funktionseinheit 3 (a) und 38 (b)

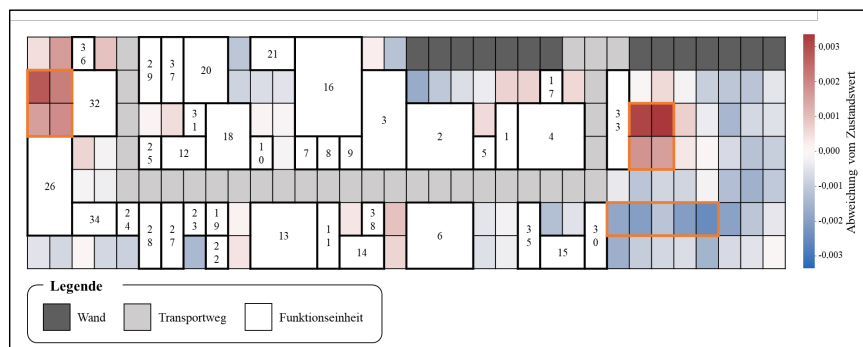


Abbildung 8-9: Saliency Map der Occlusion-Sensitivitätsanalyse der freien Fläche für den Positionierungsprozess von Funktionseinheit 38

Zusammenfassend werden im vierten Schritt erfolgreich verschiedene Layoutvarianten unter Berücksichtigung mehrerer Zielgrößen generiert. Weiterhin kann durch die Anwendung der XRL-Ansätze sichergestellt werden, dass die Ergebnisse vertrauenswürdig und nicht nur das Ergebnis einer zufälligen Platzierung sind.

Schritt 5: Manuelle Planung

Im fünften Schritt werden eine oder mehrere Layoutvarianten ausgewählt und die Detailplanung unter Berücksichtigung aller vorliegender Informationen, Anforderungen und Details durchgeführt. Für die Auswahl der Layoutvarianten kann die bereits visualisierte Pareto-Front (siehe Abbildung 8-2) verwendet werden. Alternativ können die Layoutvarianten anhand aller implementierten Zielgrößen bewertet werden, auch wenn diese nicht in der Optimierung berücksichtigt wurden.

8.2 Neuplanungsszenario

Schritt 1: Initialisierung des Layoutplanungsproblems

Das zweite Validierungsszenario umfasst die Neuplanung des Layouts der Achsenfertigung eines Nutzfahrzeugherstellers. Nach der Planung sollen verschiedene Vorprodukte wie Wellen oder Zahnräder im Hartfeinbereich spanend bearbeitet werden. Ziel ist es, 32 Funktionseinheiten so anzuordnen, dass die Transportintensitäten minimiert werden. Die Maße der Funktionseinheiten sind in Tabelle 8-3 aufgeführt. Zur Bestimmung der Elementarzellengröße wird die kleinste Funktionseinheit als Referenz genutzt. Anschließend wird diese zur Festlegung der Größe der verbleibenden Funktionseinheiten verwendet. Weiterhin gibt es aufgrund der Neuplanung keinen definierten Warenein- (WE) und -ausgang (WA). Daher werden diese als zwei zusätzliche Funktionseinheiten mit der Größe 1 x 1 initialisiert.

Tabelle 8-3: Größe der Funktionseinheiten in Elementarzellen

Funktionseinheit	WE	1	2	3	4	5	6	7	8
x-Dimension	1	1	1	1	1	1	1	2	1
y-Dimension	1	2	1	2	1	1	1	2	2

Funktionseinheit	9	10	11	12	13	14	15	16	17
x-Dimension	2	1	1	2	1	1	1	2	2
y-Dimension	2	1	3	1	2	2	1	2	2

Funktionseinheit	18	19	20	21	22	23	24	25	26
x-Dimension	1	2	1	3	2	1	1	2	1
y-Dimension	2	1	2	2	2	1	1	3	2

Funktionseinheit	27	28	29	30	31	32	WA
x-Dimension	3	3	3	1	3	3	1
y-Dimension	3	2	3	1	2	2	1

Da kein bestehendes Layout vorhanden ist, wird ein Raster der Größe 30 x 30 initialisiert. Dieses enthält ausschließlich freie Flächen, sodass die Funktionseinheiten frei positioniert werden können. Weiterhin wird die Positionierung nicht durch Medienverfügbarkeiten, Bodenlasten, oder Höhenrestriktionen eingeschränkt, da eine Neuplanung durchgeführt wird.

Die Festlegung der Materialfluss- und PPS-Eigenschaften basiert auf den prognostizierten zukünftigen Transportmengen der zu planenden Fertigung. Diese sind in Anhang B in Form einer Transportmatrix aufgeführt (siehe Abbildung 11-2). Darüber hinaus liegen dem Unternehmen zu diesem Planungszeitpunkt noch keine Informationen über das angestrebte Materialflusssystem vor, weshalb ausschließlich die Transportintensität als Optimierungsziel

genutzt wird. Folglich werden für die Initialisierung des Ansatzes keine weiteren Informationen zu Transportmitteln, Auftragsreihenfolgeplanung und Materialflusssteuerung benötigt.

Schritt 2: Initialisierung des RL-Ansatzes

Der Übergang von Schritt 1 zu Schritt 2 erfolgt durch die Priorisierung der Zielgrößen für die Optimierung. Wie geschildert, wird nur die Transportintensität als Zielgröße berücksichtigt. Demnach wird die entsprechende Zielgrößenvorwahl $w_5 = 1$ gewählt, während alle anderen Gewichte den Wert von Null annehmen. Da es sich in diesem Validierungsszenario um ein einzelkriterielles Optimierungsproblem handelt, wird nur ein RL-Agent mit den Hyperparametern aus Kapitel 5.2 initialisiert.

Schritt 3: Training

Die Trainingsdauer ist für den Agenten auf 72 Stunden unter Nutzung von 30 GB RAM eines Intel Xeon E5-2670 Prozessors begrenzt. Während des Trainings werden alle Transitionen und die besten Layouts gespeichert. Parallel hierzu werden die KNN-Modelle des RL-Agenten alle 6000 Transitionen extrahiert und gespeichert.

Schritt 4: Evaluation der Trainingsergebnisse

Die Ergebnisse werden nach dem Training aggregiert und visualisiert. Als Vergleichsmaßstab wird nur die Kombination aus GA und TS herangezogen, da der NSGA für einzelkriterielle Problemstellungen einem konventionellen GA entspricht, der bereits in der integrierten Version aus GA und TS enthalten ist.

Die schrittweise Verbesserung der bestehenden besten Lösung ist in Abbildung 8-10 dargestellt. Es kann beobachtet werden, dass die Lösung des RL-Ansatzes mit einer Episodenbewertung von -3,14 um ca. 7 % besser ist als die Lösung der Metaheuristik, die eine Bewertung von -3,36 erreicht. Außerdem wird deutlich, dass die Überlegenheit des Ansatzes von der Rechenzeit abhängt. Im Gegensatz zu den Ergebnissen aus Kapitel 8.1 erzielt der RL-Ansatz bereits zu Beginn und erneut am Ende des Trainings eine bessere Lösung, während der TS-GA-Ansatz im mittleren Trainingsabschnitt überlegene Lösungen errechnet. Die höhere Lösungsqualität zu Beginn des Trainingsprozesses kann möglicherweise auf die Zielfunktionskomplexität zurückgeführt werden, da in diesem Anwendungsszenario nur einzelkriteriell optimiert wird. Daher wird vermutet, dass die Zusammenhänge schneller erlernt wurden als im multikriteriellen Fall. Weiterhin kann beobachtet werden, dass der Verbesserungsprozess des TS-GA-Ansatzes überwiegend inkrementell verläuft, während beim RL-Ansatz vergleichsweise sprunghafte Anstiege zu beobachten sind.

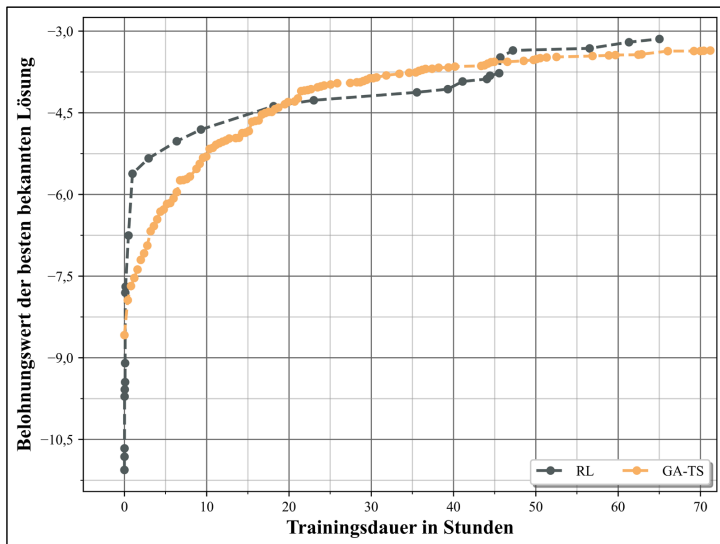


Abbildung 8-10: Vergleich des Trainingsverhaltens zwischen RL und GA-TS

Die Veränderung der generierten Layouts ist in Abbildung 8-11 dargestellt. Diese zeigt das finale Layout und die besten Layouts nach 1000 und 10000 Trainingsepisoden, die mit dem RL-Ansatz generiert wurden. Nach 1000 Episoden sind die Funktionseinheiten noch weitestgehend zufällig im Raster positioniert. Die Funktionseinheiten stehen mitunter weit voneinander entfernt, was zu längeren Transportwegen führt. Die Lösung wird mit zunehmender Trainingsdauer verbessert. Dies kann daran erkannt werden, dass das Layout nach 10000 Episoden stärker um den Kern des Bildausschnitts zentriert ist. In diesem ist bereits eine freie Fläche zu erkennen, die von Funktionseinheiten umgeben ist und für Materialtransporte verwendet werden kann. Die Funktionseinheiten 25 und 27 sind jedoch am Rand des Bildausschnitts positioniert, sodass hier lange Transportwege entstehen, die vermieden werden können. Im finalen Layout werden diese Defizite reduziert. Es wird deutlich, dass Merkmale des vorherigen Layouts bestehen bleiben, wie die Materialflusslinie, die in der Mitte des Layouts verläuft. Somit kann das generierte Layout einen wertvollen Ausgangspunkt für die weiteren manuellen Planungsschritte darstellen.

In einem weiteren Teilschritt werden die XRL-Methoden zur vertieften Analyse des Trainingsprozesses und zur Bewertung der Vertrauenswürdigkeit der Ergebnisse genutzt. Die Vorgehensweise der Analyse erfolgt analog zu Kapitel 8.1, sodass zunächst die Policy Summarization Methode zur Verteilungs- und Übereinstimmungsanalyse durchgeführt wird und anschließend die Ergebnisse der Occlusion-Sensitivitätsanalyse betrachtet werden. Die zugehörigen Abbildungen sind in Anhang B aufgeführt und die Ergebnisse können wie folgt zusammengefasst werden:

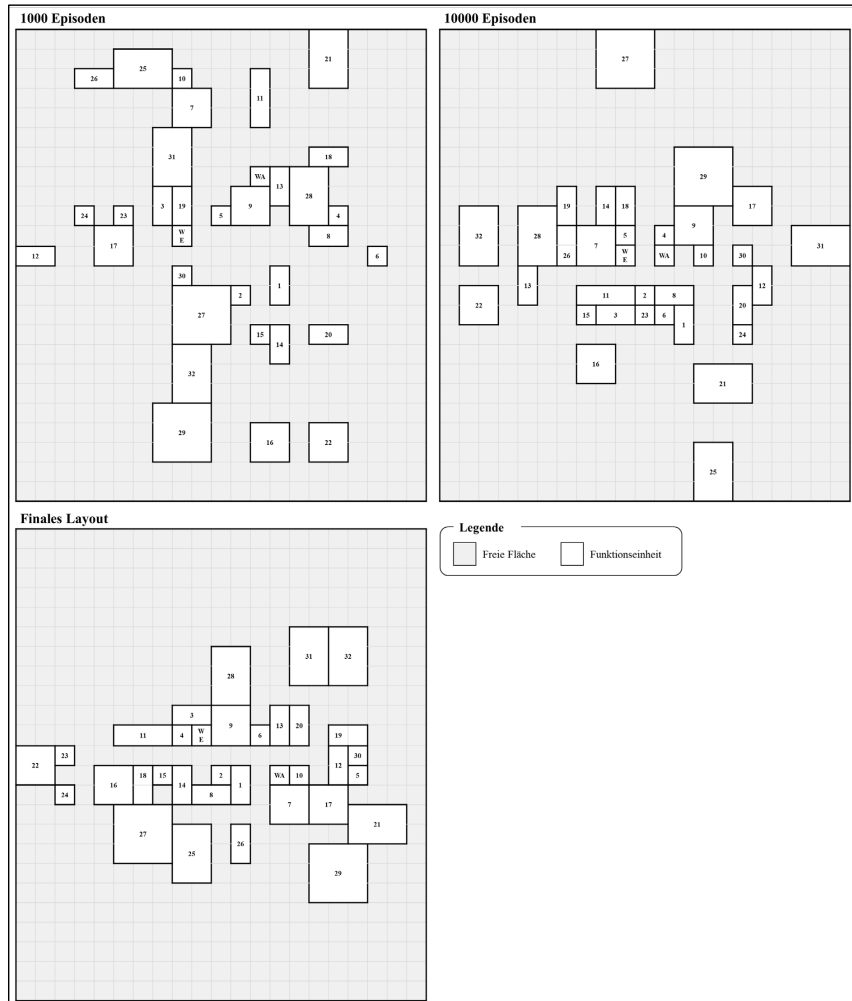


Abbildung 8-11: Visualisierung des generierten Layouts zu verschiedenen Trainingszeitpunkten

- Die Ergebnisauswertung der Verteilungsanalyse zeigt, dass die Streuung in der Positionierung mit zunehmender Trainingsdauer abnimmt. Die Positionierungsstreuung von Funktionseinheiten, die früh in einer Episode platziert werden, nimmt schneller ab als bei Funktionseinheiten, die später positioniert werden. Mit zunehmender Trainingsdauer kann auch für die letzten Funktionseinheiten einer Episode eine Konzentration in der Positionierung beobachtet werden. Daraus kann geschlossen werden, dass das finale Layout ein Ergebnis der erlernten Strategie ist (siehe Anhang B: Abbildung 11-3 bis Abbildung 11-8).

- Die Ergebnisse der Übereinstimmungsanalyse unterstützen diese Beobachtung, da der Übereinstimmungsanteil mit zunehmender Trainingsdauer zunimmt. Es wird deutlich, dass für die ersten 11 Funktionseinheiten eine nahezu vollständige Übereinstimmung zwischen PSL und PSFU besteht. Diese Beobachtung lässt darauf schließen, dass die Positionierungsentscheidungen der ersten Funktionseinheiten nur geringfügig durch Positionierungskonflikte geprägt sind (siehe Anhang B: Abbildung 11-9).
- Schließlich zeigt auch die Auswertung der Occlusion-Sensitivitätsanalyse, die für die Funktionseinheiten 5 und 25 durchgeführt wird, dass die Kurven gegen Null konvergieren (siehe Anhang B: Abbildung 11-10).

Zusammenfassend zeigen alle Ergebnisse, dass das generierte Layout vorrangig durch die erlernte Strategie des RL-Agenten geprägt ist und zufällige Entscheidungen eine untergeordnete Rolle bei der Entscheidungsfindung für das finale Layout spielen.

Schritt 5: Manuelle Planung

Aufbauend auf der positiven Bewertung der Vertrauenswürdigkeit erfolgt im fünften Schritt die manuelle Detailplanung des Layouts, die nicht Bestandteil dieser Arbeit ist. Hierbei wird u. a. die Halle geometrie des Layouts festgelegt und die Positionierung der Funktionseinheiten wird verfeinert. Im Einklang mit der Festlegung der materialflussbezogenen Elemente kann die Detailplanung der Arbeitsstationen erfolgen.

8.3 Diskussion und Schlussbetrachtung

Zielerreichung

Das übergeordnete Ziel dieser Dissertation ist die Erhöhung der Qualität der Layoutplanung durch die kombinierte Nutzung von RL-, EDS-Ergebnissen und XRL-Ansätzen. Der hierfür entwickelte RL-Ansatz wird anhand von zwei industriellen Anwendungsszenarien zur Neu- und Umplanung validiert. Die Ergebnisse bestätigen das Leistungspotential von RL für die Layoutplanung. Demnach kann RL zur Generierung verschiedener Layoutvarianten unter Berücksichtigung von EDS-Ergebnissen eingesetzt werden. Weiterhin kann gezeigt werden, dass die Güte der generierten Lösungen die Lösungsgüte konventioneller Metaheuristiken bei gleicher Rechendauer übertrifft, wenn die Rechenzeit ausreichend lang ist. Ein weiterer Vorteil gegenüber den Metaheuristiken ist die Bündelung der Lösungsstrategie im KNN. Hierdurch ist es möglich, die entwickelten XRL-Ansätze zur Bewertung der Vertrauenswürdigkeit anzuwenden und den Trainings- und Lösungsprozess im Detail zu analysieren.

Neben der übergeordneten Zielsetzung wurden fünf Anforderungen aus dem Stand der Wissenschaft und Technik abgeleitet. Die Erfüllung dieser Anforderungen wird nachfolgend erläutert und die Stärken und Schwächen des Gesamtkonzepts diskutiert.

Anforderung 1: Anwendbarkeit für die Neu- und Umplanung

Anhand der zuvor aufgeführten Validierungsszenarien wird deutlich, dass sowohl Neuplanungs- als auch Umplanungsprobleme mit dem entwickelten Ansatz gelöst werden können. Für die Unterscheidung ist die Probleminitialisierung ausschlaggebend. Durch das Einbringen von optionalen Randbedingungen oder die Beschränkung auf eine vorgegebene Fläche können beide Planungsfälle modelliert und anschließend gelöst werden.

Anforderung 2: Multikriterielle Optimierung

Der entwickelte Ansatz ermöglicht die Optimierung von einzel- und multikriteriellen Problemstellungen. Um dieser Anforderung gerecht zu werden, kann der Ansatz analytisch beschreibbare Zielgrößen und dynamische Zielgrößen des Materialflusses, die mittels EDS bestimmt werden, in der Optimierung berücksichtigen. Durch die Priorisierungsmöglichkeit können Planende die Optimierung problemspezifisch initialisieren. Zudem können durch unterschiedliche Zielgrößengewichtungen unterschiedliche Fabriklayoutvarianten generiert werden, die einen Ausgangspunkt für den manuellen Detailplanungsschritt darstellen. Abschließend ist festzuhalten, dass die implementierten Zielgrößen nicht als abschließend zu betrachten sind, sondern als eine erste Auswahl relevanter Optimierungsgrößen.

Anforderung 3: Realitätsnähe der Planungsergebnisse

Der Erfüllungsgrad der dritten Anforderung hängt von den verfügbaren Informationen ab. Grundsätzlich werden die Eingangsinformationen in erforderliche und optionale Informationen unterteilt. Somit können unterschiedliche Problemstellungen von Single-Row bis zum Open-Field Layoutplanungsproblem modelliert werden. Dabei werden die Funktionseinheiten als Einheitsgrößen oder entsprechend ihrer 3D-Geometrie unter Berücksichtigung von Aufnahme- und Abgabepunkten für den Materialtransport initialisiert. Zudem können vielfältige Nebenbedingungen in der Optimierung berücksichtigt werden, wenn die entsprechenden Informationen vorliegen. Folglich wird die Anforderung erfüllt, wenn alle Eingangsinformationen für das System vorliegen, ohne dass die Anwendbarkeit des Ansatzes eingeschränkt wird, wenn weniger Informationen zur Verfügung stehen.

Anforderung 4: Skalierbarkeit hinsichtlich der Problemgröße

Der entwickelte Ansatz kann auf Probleme mit industriell relevanten Problemgrößen angewendet werden und findet in den Validierungsszenarien in weniger als 72 Stunden eine optimierte Lösung für die Layoutplanungsprobleme. Darüber hinaus werden in Kapitel 5.2 Probleme mit unterschiedlichen Problemgrößen gelöst. Dies zeigt, dass der Ansatz übertragbar einsetzbar und damit skalierbar ist. Jedoch muss angemerkt werden, dass die benötigte Trainingszeit mit der Problemgröße korreliert und dass diese durch die Nutzung der EDS im Vergleich zum rein analytischen Fall steigt. Eine optimale Lösung kann nicht in jedem Fall garantiert werden. Es kann festgestellt werden, dass der RL-Ansatz die betrachteten Metaheuristiken hinsichtlich der Lösungsgüte übertrifft, wenn ausreichend viel Rechenzeit zur Verfügung steht. Weiterhin ist zu beachten, dass sowohl im Bereich der konventionellen Metaheuristiken als auch RL kontinuierliche, neue Entwicklungen mit leistungsfähigeren Ansätzen publiziert werden, sodass der Vergleich nicht als endgültig angesehen werden kann.

Anforderung 5: Nachvollziehbarkeit der Planungsergebnisse

Da der entwickelte Ansatz KNNs verwendet, um die mitunter komplexen Zustandsinformationen zu verarbeiten, stellt die Nachvollziehbarkeit aufgrund des Black-Box-Charakters des KNNs eine natürliche Einschränkung des RL-Ansatzes dar. Sowohl RL als auch konventionelle Metaheuristiken lösen das Optimierungsproblem mit Hilfe stochastischer Operatoren. Da RL jedoch die Zusammenhänge zur Lösung des Problems im KNN speichert, können die entwickelten XRL-Ansätze genutzt werden, um den Lösungsprozess zu analysieren. Eine solche Analyse ist mit konventionellen Metaheuristiken nicht möglich, da diese kein Wissen über das Problem erlernen. Die XRL-Ansätze ermöglichen somit die Bewertung der

Vertrauenswürdigkeit der generierten Lösung. Sie können zur Steigerung der Nachvollziehbarkeit des Lösungsprozesses beitragen. Nichtsdestotrotz stellt die Erklärbarkeit von KNNs einen weiterführenden Forschungsbedarf dar.

9 Zusammenfassung und Ausblick

Die Fabriklayoutplanung als zentraler Bestandteil der Fabrikplanung hat einen erheblichen Einfluss auf die Wirtschaftlichkeit und den Betrieb des resultierenden Produktionssystems, sodass eine hohe Qualität des Planungsprozesses von Relevanz ist. Um diese Planungsqualität sicherzustellen, werden in der frühen Phase der Fabriklayoutplanung häufig Layoutvarianten unter Berücksichtigung von Konflikten zwischen Planungszielen und Positionierungskonflikten entworfen. Da die manuelle Fabriklayoutplanung häufig zeitintensiv ist und auf Erfahrungswissen basiert, werden automatisierte Methoden als Unterstützungswerkzeug eingesetzt. Diese ermöglichen die Generierung von Layoutvarianten, wodurch die benötigte Planungszeit reduziert und eine bessere Planungsgrundlage gewährleistet werden kann. Approximative automatisierte Verfahren können industriell relevante Problemstellungen lösen, erfordern jedoch häufig eine analytische und vereinfachende Beschreibung, wodurch dynamische und layoutbestimmende Effekte eines Produktionssystems unberücksichtigt bleiben. Mit Hilfe von EDS können diese Einflüsse realitätsnah simuliert werden. Entwicklungen im Bereich von RL zeigen, dass RL in der Lage ist, selbstständig Lösungsstrategien für komplexe Problemstellungen zu erlernen, deren Systemverhalten analytisch nicht vollständig beschreibbar ist. Die Kopplung von RL und EDS bietet somit das Potential, durch eine realitätsnahe Planungsgrundlage automatisiert optimierte Layoutvarianten zu erzeugen, die eine hohe Wirtschaftlichkeit des Produktionssystembetriebs ermöglichen. Darüber hinaus können XRL-Ansätze genutzt werden, um Einblicke in das Lösungsverhalten des RL-Ansatzes zu erlangen.

Das Ziel dieser Dissertation ist die Verbesserung der Qualität in der frühen Layoutplanungsphase durch die kombinierte Nutzung von RL, EDS-Ergebnissen und XRL-Ansätzen. Um dieses Ziel zu erreichen, wurde ein RL-Ansatz entwickelt, der eine multikriterielle Neu- und Umlanung von Fabriklayouts unter Berücksichtigung der dynamischen Eigenschaften des Materialflusses ermöglicht.

Zunächst wurde das Fabriklayoutplanungsproblem in Kapitel 4 als RL-kompatibles sequentielles Entscheidungsproblem modelliert. Hierfür wurden die Systemgrenzen (Kapitel 4.1), mögliche Aktionsspielräume (Kapitel 4.2) und die Zustandsinformationen (Kapitel 4.3) modelliert. Letztere umfassen Informationen über das Layout, den Materialfluss und die Funktionseinheiten und ermöglichen es dem RL-Agenten, den aktuellen Zustand des Planungsproblems zu erfassen. Zur Verarbeitung dieser Informationen wurde eine Kombination aus einer MLP-, CNN- und GANN-KNN-Architektur verwendet (siehe Kapitel 4.4). In Kapitel 4.5 wurden 9 Zielgrößen der Belohnungsfunktion modelliert und implementiert, die analytisch oder mittels EDS bestimmt werden konnten. Für die EDS-basierten Zielgrößen wurde ein auf Python aufbauendes EDS-Modul entwickelt. Dieses wurde modular gestaltet, sodass unterschiedliche Produktionssysteme mit derselben Schnittstelle simuliert und evaluiert werden können. Hierfür wurden die Kernfunktionalitäten der Funktionseinheiten, Transportmittel, Produktionsplanung und -steuerung, Materialflussteuerung und externe Logistik genutzt.

Da im Vorfeld nicht abgeschätzt werden konnte, welcher RL-Agententyp das Problem am besten löst, wurde in Kapitel 5 die Leistungsfähigkeit verschiedener Agententypen anhand von drei Planungsproblemen mit unterschiedlicher Komplexität verglichen. Die Ergebnisse zeigten, dass der *Rainbow DQN*-Agent mit multidiskretem Aktionsspielraum die beste

Leistungsfähigkeit besitzt. Weiterhin wurden Strategien zum Umgang mit ungünstigen Aktionen untersucht. Hierbei zeigte der Einsatz einer Maskierungsfunktion, die eine Auswahl von ungünstigen Aktionen vorbeugt, das beste Resultat. Abschließend wurde die Struktur zur multikriteriellen Optimierung auf Basis von Versuchsergebnissen festgelegt. Die Ergebnisse zeigten, dass ein einzelner Agent, der die Linearkombination der Belohnungsfunktion erlernt, zu besseren Resultaten als die Nutzung von mehreren Agenten führt. Mit Kapitel 5 lag ein funktionsfähiger RL-Ansatz zur multikriteriellen und simulationsgestützten Neu- und Umlanung von Fabriklayouts vor.

In Kapitel 6 wurden XRL-Ansätze entwickelt, die es ermöglichen, die Vertrauenswürdigkeit der generierten Lösungen zu bewerten. Nachdem in Kapitel 6.1 primäre und sekundäre Erklärbarkeitsziele definiert wurden, wurde die Funktionsweise der Policy Summarization (Kapitel 6.2.1) und Occlusion-Sensitivitätsansätze (Kapitel 6.2.2) anhand beispielhafter Ergebnisse beschrieben.

Die Ergebnisse der vorangegangenen Schritte wurden in Kapitel 7 in ein zusammenfassendes Konzept zur RL-basierten Layoutplanung überführt, das die Grundlage für einen Python-basierten Softwaredemonstrator bildet.

Der RL-Ansatz wurde in Kapitel 8 anhand von zwei industriellen Anwendungsszenarien zur Neu- und Umlanung validiert. Die Ergebnisse bestätigten das Leistungspotential von RL für die Layoutplanung. RL konnte zur Generierung verschiedener Layoutvarianten genutzt werden und dabei EDS-Ergebnisse und analytische Zielgrößen berücksichtigen. Weiterhin konnte gezeigt werden, dass die Qualität der generierten Lösungen die Lösungsgüte konventioneller Metaheuristiken bei ausreichender Rechendauer überstieg. Ein weiterer Vorteil gegenüber Metaheuristiken bestand in der Bündelung der Lösungsstrategie im KNN des Agenten. Hierdurch war es möglich, die entwickelten XRL-Ansätze zur Bewertung der Vertrauenswürdigkeit anzuwenden und den Trainings- und Lösungsprozess im Detail zu analysieren.

Es existieren **weiterführende Entwicklungspotentiale**, die u. a. das Generalisierbarkeitspotential von RL oder die gewählten Eingangsinformationen adressieren. Das Generalisierbarkeitspotential beschreibt die Fähigkeit von RL eine übergreifende Problemlösungsstrategie zu erlernen. Im Fall der RL-basierten Fabriklayoutplanung bedeutet dies, dass ein RL-Ansatz trainiert wird, der ohne weiteren Trainingsaufwand auf unterschiedliche Fabrikplanungsprobleme anwendbar ist. Erste eigene Ergebnisse wurden als Vorarbeit in [Klar24d] publiziert. Dabei werden die Zustandsinformationen und das Training so angepasst, dass fünf Stufen der Übertragbarkeit realisiert werden:

1. Veränderbare Transportintensitäten
2. Veränderbare Layoutgröße und Layoutinformationen
3. Veränderbare Funktionseinheitsanzahl
4. Veränderbare Priorisierung der Zielfunktion
5. Kombination der zuvor genannten Stufen

Die Ergebnisse zeigen, dass RL eine übertragbare Lösungsfähigkeit erlernt, sodass Planungsprobleme mit deutlich reduziertem Trainingsaufwand (Faktor bis zu 1/100) gelöst werden und zeitgleich die Lösungsqualität gesteigert wird. Dieser als Transfer Lernen bezeichnete Ansatz weist erhebliche Leistungspotentiale auf (siehe Anhang D, Tabelle 11-4)

[Klar24d]. Ein Forschungsbedarf besteht im Training eines generalisierten RL-Agenten, der ohne ein weiteres Training eingesetzt werden kann. Dazu müssen die Zustandsinformationen erweitert und möglicherweise neue, leistungsfähigere KNN-Architekturen entwickelt werden.

10 Quellenverzeichnis

- [Adad18] A. Adadi, M. Berrada: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018): S. 52138-52160. DOI: 10.1109/ACCESS.2018.2870052
- [Aggt87] B. Aggteleky: Fabrikplanung. 2., durchges. Aufl. der Neuausg. Hanser, (1987). ISBN: 9783446148604
- [Ahma16] A. Ahmadi, M.R. Akbari Jokar: An efficient multiple-stage mathematical programming method for advanced single and multi-floor facility layout problems. *Applied Mathematical Modelling* 40/9-10 (2016): S. 5605-5620. DOI: 10.1016/j.apm.2016.01.014
- [Aiel12] G. Aiello, G. La Scalia, M. Enea: A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding. *Expert Systems with Applications* 39/12 (2012): S. 10352-10358. DOI: 10.1016/j.eswa.2012.01.125
- [Alba17] S. Albawi, T.A. Mohammed, S. Al-Zawi: Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET) - Proceedings of the 2017 International Conference on Engineering and Technology (ICET) (2017): S. 1-6. DOI: 10.1109/ICEngTechnol.2017.8308186
- [Al-S99] K.S. Al-Sultan, M.A. Al-Fawzan: A tabu search approach to the uncapacitated facility location problem. *Annals of Operations Research* 86 (1999): S. 91-103. DOI: 10.1023/A:1018956213524
- [Amar09] A.R. Amaral: A new lower bound for the single row facility layout problem. *Discrete Applied Mathematics* 157/1 (2009): S. 183-190. DOI: 10.1016/j.dam.2008.06.002
- [Amar06] A.R. Amaral: On the exact solution of a facility layout problem. *European Journal of Operational Research* 173/2 (2006): S. 508-518. DOI: 10.1016/j.ejor.2004.12.021
- [Amir18] D. Amir, O. Amir: Summarizing Agent Behavior to People. *AAMAS: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (2018): S. 1168-1176.
- [Anjo17] M.F. Anjos, M.V. Vieira: Mathematical optimization approaches for facility layout problems: The state-of-the-art and future research directions. *European Journal of Operational Research* 261/1 (2017): S. 1-16. DOI: 10.1016/j.ejor.2017.01.049
- [Arno19] D. Arnold, K. Furmans: Planung von Materialflusssystemen. (2019): S. 249-348. DOI: 10.1007/978-3-662-60388-8_7
- [Arno08] D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier, K. Furmans: *Handbuch Logistik*. (2008).
- [Arri19] A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, F. Herrera: Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *arXiv*, (2019). DOI: 10.48550/arXiv.1910.10045
- [ASIM97] ASIM: Leitfaden für Simulationsbenutzer in Produktion und Logistik. *ASIM-Mitteilungen* 58 (1997).
- [Auri04] J.C. Aurich, P. Barbian, C. Naab: Produktionsstrategie im Produktlebenszyklus. *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 99/5 (2004): S. 218-223. DOI: 10.3139/104.100768
- [Auri05] J.C. Aurich, K. Köklü, M. Röbbing: Verteilte kooperative Fabrikplanung. *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 100/6 (2005): S. 340-344. DOI: 10.3139/104.100906
- [Auri20] P. Aurich: Entwicklung einer Methode zur quantitativen, mehrdimensionalen Fabriklayoutplanung mittels mathematischer Modellierung von fabrikplanungsrelevanten Eigenschaften. *Schlussbericht*. (2020).
- [Azad16] A. Azadeh, M. Moghaddam, T. Nazari, M. Sheikhalishahi: Optimization of facility layout design with ambiguity by an efficient fuzzy multivariate approach. *The International Journal of Advanced Manufacturing Technology* 84/1-4 (2016): S. 565-579. DOI: 10.1007/s00170-015-7714-x
- [Azim13] P. Azimi, E. Saberi: An Efficient Hybrid Algorithm for Dynamic Facility Layout Problem using Simulation Technique and PSO. *Economic Computation and Economic Cybernetics Studies and Research* (2013).
- [Bang08] S. Bangsow: *Fertigungssimulationen mit Plant Simulation und SimTalk – Anwendung und Programmierung mit Beispielen und Lösungen*. Hanser, (2008). ISBN: 978-3446414907

- [Bank10] J. Banks: Discrete-event system simulation. 5th ed. Pearson/Prentice Hall, (2010). ISBN: 978-0136062127
- [Bell17] M.G. Bellemare, W. Dabney, R. Munos: A Distributional Perspective on Reinforcement Learning. arXiv, (2017). DOI: 10.48550/arXiv.1707.06887
- [Bell98] M. Bellgran: Systematic design of assembly systems – Preconditions and design process planning. Assembly Technology Department of Mechanical Engineering, (1998). ISBN: 9172191422
- [Benk14] N. Benkamoun, W. ElMaraghy, A.-L. Huyet, K. Kouiss: Architecture Framework for Manufacturing System Design. Procedia CIRP 17 (2014): S. 88-93. DOI: 10.1016/j.procir.2014.01.101
- [Berg06] M. Bergholz: Objektorientierte Fabrikplanung. Shaker, (2006). ISBN: 978-3832257705
- [Bert93] D. Bertsimas, J. Tsitsiklis: Simulated Annealing. Statistical Science 8/1 (1993). DOI: 10.1214/ss/1177011077
- [Besb21] M. Besbes, M. Zolghadri, R. Costa Affonso, F. Masmoudi, M. Haddar: 3D facility layout problem. Journal of Intelligent Manufacturing 32/4 (2021): S. 1065-1090. DOI: 10.1007/s10845-020-01603-z
- [Bian09] L. Bianchi, M. Dorigo, L.M. Gambardella, W.J. Gutjahr: A survey on metaheuristics for stochastic combinatorial optimization. Natural Computing 8/2 (2009): S. 239-287. DOI: 10.1007/s11047-008-9098-4
- [Brac18a] U. Bracht, M. Dahlbeck, A. Fischer, T. Krüger: Combining Simulation and Optimization for Extended Double Row Facility Layout Problems in Factory Planning 889 (2018): S. 39-59. DOI: 10.1007/978-3-319-96271-9_3
- [Brac18b] U. Bracht, M. Schlegel: AR- und VR-Brillen in der Fabrikplanung*/Current AR- and VR-glasses in factory planning. wt Werkstattstechnik online 108/04 (2018): S. 251-256. DOI: 10.37544/1436-4980-2018-04-57
- [Brai93] L.C. Braid: Advantage Updating. Technical Report WL-TR-93-1146 (1993).
- [Bron21] M.M. Bronstein, J. Bruna, T. Cohen, P. Veličković: Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. arXiv, (2021). DOI: 10.48550/arXiv.2104.13478
- [Budi22] H.S. Budi, M. Elveny, P. Zhuravlev, A.T. Jalil, S. Al-Janabi, A.F. Alkaim, M.M. Saleh, R.A. Shichiyakh, Sutarto: Development of an Adaptive Genetic Algorithm to Optimize the Problem Of Unequal Facility Location. Foundations of Computing and Decision Sciences 47/2 (2022): S. 111-125. DOI: 10.2478/fcds-2022-0006
- [Buff64] E.S. Buffa, G.C. Armour, T.E. Vollman: Allocating facilities with CRAFT. Harvard Business Review 42 (1964): 2.
- [Bung13] H.-J. Bungartz, S. Zimmer, M. Buchholz, D. Pflüger: Modellbildung und Simulation. Springer Berlin Heidelberg, (2013). DOI: 10.1007/978-3-642-37656-6
- [Burg21a] P. Burggräf, T. Adlon, V. Hahn, T. Schulz-Isenbeck: Fields of action towards automated facility layout design and optimization in factory planning – A systematic literature review. CIRP Journal of Manufacturing Science and Technology 35 (2021): S. 864-871. DOI: 10.1016/j.cirpj.2021.09.013
- [Burg21b] P. Burggräf, G. Schuh: Fabrikplanung. Springer Berlin Heidelberg, (2021). DOI: 10.1007/978-3-662-61969-8
- [Büsc14] C. Büscher, H. Voet, T. Meisen, M. Krunke, K. Kreisköther, A. Kampker, D. Schilberg, S. Jeschke: Improving Factory Planning by Analyzing Process Dependencies. Procedia CIRP 17 (2014): S. 38-43. DOI: 10.1016/j.procir.2014.01.142
- [Cast16] D. Castelvocchi: Can we open the black box of AI? Nature 538/7623 (2016): S. 20-23. DOI: 10.1038/538020a
- [Chau95] Y. Chauvin, D.E. Rumelhart: Backpropagation – Theory, architectures, and applications. Lawrence Erlbaum Associates, Inc, (1995). ISBN: 9781134775811
- [Chen98] M. Chen: A mathematical programming model for system reconfiguration in a dynamic cellular manufacturing environment. Annals of Operations Research 77 (1998): S. 109-128. DOI: 10.1023/A:1018917109580

- [Chia96] W.C. Chiang, P. Kouvelis: An improved tabu search heuristic for solving facility layout design problems. *International Journal of Production Research* 34/9 (1996): S. 2565-2585. DOI: 10.1080/00207549608905045
- [Cina75] U. Cinar: Facilities planning: A systems analysis and space allocation approach. *Spatial synthesis in computer-aided building design* (1975): S. 19-40.
- [Conf21] R. Confalonieri, L. Coba, B. Wagner, T.R. Besold: A historical perspective of explainable Artificial Intelligence. *WIREs Data Mining and Knowledge Discovery* 11/1 (2021): e1391. DOI: 10.1002/widm.1391
- [Cruz23] F. Cruz, R. Dazeley, P. Vamplew, I. Moreira: Explainable robotic systems: understanding goal-driven actions in a reinforcement learning scenario. *Neural Computing and Applications* 35/25 (2023): S. 18113-18130. DOI: 10.1007/s00521-021-06425-5
- [Cunh20] B. Cunha, A.M. Madureira, B. Fonseca, D. Coelho: Deep Reinforcement Learning as a Job Shop Scheduling Solver: A Literature Review 923 (2020): S. 350-359. DOI: 10.1007/978-3-030-14347-3_34
- [DAS93] S.K. DAS: A facility layout method for flexible manufacturing systems*. *International Journal of Production Research* 31/2 (1993): S. 279-297. DOI: 10.1080/00207549308956725
- [Domb15] U. Dombrowski: *Ganzzeitliche Produktionssysteme*. (2015). DOI: 10.1007/978-3-662-46164-8
- [Domb10] U. Dombrowski, S. Hennersdorf: Systematische Methodenauswahl in der Fabrikplanung: Eine Auswahlssystematik für Methoden und Werkzeuge der Fabrikplanung. *Werkstattstechnik online* 100 (2010): 4.
- [Dong20] H. Dong, Z. Ding, S. Zhang: *Deep Reinforcement Learning*. Springer Singapore, (2020). DOI: 10.1007/978-981-15-4095-0
- [Dong09] M. Dong, C. Wu, F. Hou: Shortest path based simulated annealing algorithm for dynamic facility layout problem under dynamic business environment. *Expert Systems with Applications* 36/8 (2009): S. 11221-11232. DOI: 10.1016/j.eswa.2009.02.091
- [Dosi18] F.K. Dosilovic, M. Brcic, N. Hlupic: Explainable artificial intelligence: A survey. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) - Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (2018): S. 210-215. DOI: 10.23919/MIPRO.2018.8400040
- [Drir07] A. Drira, H. Pierrel, S. Hajri-Gabouj: Facility layout problems: A survey. *Annual Reviews in Control* 31/2 (2007): S. 255-267. DOI: 10.1016/j.arcontrol.2007.04.001
- [Dunk05] T. Dunker, G. Radons, E. Westkämper: Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem. *European Journal of Operational Research* 165/1 (2005): S. 55-69. DOI: 10.1016/j.ejor.2003.01.002
- [Dürr01] S. Dürrschmidt: *Planung und Betrieb wandlungsfähiger Logistiksysteme in der variantenreichen Serienproduktion*. Utz, (2001). ISBN: 978-3-8316-0023-6
- [Emam13] S. Emami, A. S. Nookabadi: Managing a new multi-objective model for the dynamic facility layout problem. *The International Journal of Advanced Manufacturing Technology* 68/9-12 (2013): S. 2215-2228. DOI: 10.1007/s00170-013-4820-5
- [Enge84] K.H. Engel: *Handbuch der Techniken des Industrial Engineering*. (1984).
- [Etsp96] M. Etspüler, M.D. Kippels: Simulation steht vor Investition. *VDI-Nachrichten* 23 (1996): 21.
- [Ever96] W. Eversheim: *Organisation in der Produktionstechnik – Band 1: Grundlagen*. Dritte, neu bearbeitete und erweiterte Auflage. Springer Berlin Heidelberg, (1996). DOI: 10.1007/978-3-642-87737-7
- [Ever99] W. Eversheim, G. Schuh: *Produktion und Management*. (1999).
- [Feli98] H. Felix: *Unternehmens- und Fabrikplanung – Planungsprozesse, Leistungen und Beziehungen*. 1. Auflage. Carl Hanser Verlag, (1998). ISBN: 9783446192522
- [Forr96] S. Forrest: Genetic algorithms. *ACM Computing Surveys* 28 (1996): 1.
- [Fort17] M. Fortunato, M.G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, S. Legg: Noisy Networks for Exploration. *arXiv*, (2017). DOI: 10.48550/arXiv.1706.10295

- [Frei14] A.A. Freitas: Comprehensible classification models. ACM SIGKDD Explorations Newsletter 15/1 (2014): S. 1-10. DOI: 10.1145/2594473.2594475
- [Frug06] F. Fruggiero, A. Lambiase, F. Negri: Design and optimization of a facility layout problem in virtual environment. Proceeding of ICAD 2006 (2006): S. 1-7.
- [Fuku17] Y. Fukuchi, M. Osawa, H. Yamakawa, M. Imai: Application of Instruction-Based Behavior Explanation to a Reinforcement Learning Agent with Changing Policy 10634 (2017): S. 100-108. DOI: 10.1007/978-3-319-70087-8_11
- [Gare79] M.R. Garey, D.S. Johnson: Computers and intractability – A guide to the theory of NP-completeness. Freeman, (1979). ISBN: 0-7167-1044-7
- [Gevr03] M. Gevrey, I. Dimopoulos, S. Lek: Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecological Modelling 160/3 (2003): S. 249-264. DOI: 10.1016/S0304-3800(02)00257-0
- [Ghos16] T. Ghosh, B. Doloi, P.K. Dan: Applying soft-computing techniques in solving dynamic multi-objective layout problems in cellular manufacturing system. The International Journal of Advanced Manufacturing Technology 86/1-4 (2016): S. 237-257. DOI: 10.1007/s00170-015-8070-6
- [Glas08] A. Glass, D.L. McGuinness, M. Wolverton: Toward establishing trust in adaptive agents. Proceedings of the 13th international conference on Intelligent user interfaces - Proceedings of the IUI08: 13th International Conference on Intelligent User Interfaces (2008): S. 227-236. DOI: 10.1145/1378773.1378804
- [Glov90] F. Glover: Tabu Search: A Tutorial. Interfaces 20/4 (1990): S. 74-94. DOI: 10.1287/inte.20.4.74
- [Gold21] C.V. Goldman, M. Baltaxe, D. Chakraborty, J. Arinez: Explaining Learning Models in Manufacturing Processes. Procedia Computer Science 180 (2021): S. 259-268. DOI: 10.1016/j.procs.2021.01.163
- [Good16] I. Goodfellow, Y. Bengio, A. Courville: Deep learning. The MIT Press, (2016). ISBN: 978-0262035613
- [Gron12] I. Grondman, L. Busoniu, G.A.D. Lopes, R. Babuska: A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42/6 (2012): S. 1291-1307. DOI: 10.1109/TSMCC.2012.2218595
- [Grun21] C.-G. Grundig: Fabrikplanung – Planungssystematik – Methoden – Anwendungen. 7., aktualisierte Auflage. Hanser, (2021). DOI: 10.3139/9783446470064
- [Gu16] S. Gu, T. Lillicrap, Z. Ghahramani, R.E. Turner, S. Levine: Q-Prop: Sample-Efficient Policy Gradient with An Off-Policy Critic. arXiv, (2016). DOI: 10.48550/arXiv.1611.02247
- [Guan16] J. Guan, G. Lin: Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem. European Journal of Operational Research 248/3 (2016): S. 899-909. DOI: 10.1016/j.ejor.2015.08.014
- [Gunn19] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, G.-Z. Yang: XAI-Explainable artificial intelligence. Science robotics 4/37 (2019). DOI: 10.1126/scirobotics.aay7120
- [Günt09] H.-O. Günther, H. Tempelmeier: Produktion und Logistik. 8., überarb. und erw. Aufl. Springer, (2009). ISBN: 978-3642003790
- [Gute17] K. Gutenschwager, M. Rabe, S. Spieckermann, S. Wenzel: Simulation in Produktion und Logistik. Springer Berlin Heidelberg, (2017). DOI: 10.1007/978-3-662-55745-7
- [Haar18a] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, S. Levine: Learning to Walk via Deep Reinforcement Learning. arXiv, (2018). DOI: 10.48550/arXiv.1812.11103
- [Haar18b] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine: Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. arXiv, (2018). DOI: 10.48550/arXiv.1801.01290
- [Habe12] R. Haberfellner: Systems Engineering. (2012).
- [Hadi13] A. Hadi-Vencheh, A. Mohamadghasemi: An integrated AHP–NLP methodology for facility layout design. Journal of Manufacturing Systems 32/1 (2013): S. 40-45. DOI: 10.1016/j.jmsy.2012.07.009

- [Hagr18] H. Hagras: Toward Human-Understandable, Explainable AI. *Computer* 51/9 (2018): S. 28-36. DOI: 10.1109/MC.2018.3620965
- [Hamm16] A. Hammad, A. Akbarnezhad, D. Rey: A multi-objective mixed integer nonlinear programming model for construction site layout planning to minimise noise pollution and transport costs. *Automation in Construction* 61 (2016): S. 73-85. DOI: 10.1016/j.autcon.2015.10.010
- [Harm04] T. Harms: Agentenbasierte Fabrikstrukturplanung. PZH Produktionstechn. Zentrum, (2004). DOI: 2004
- [Hart91] H. Hartberger: Wissensbasierte Simulation Komplexer Produktionssysteme. Springer Berlin / Heidelberg, (1991). ISBN: 978-3-540-53326-9
- [Hass15] H.v. Hasselt, A. Guez, D. Silver: Deep Reinforcement Learning with Double Q-learning. *arXiv*, (2015). DOI: 10.48550/arXiv.1509.06461
- [Hath13] J. Hathhorn, E. Sisikoglu, M.Y. Sir: A multi-objective mixed-integer programming model for a multi-floor facility layout. *International Journal of Production Research* 51/14 (2013): S. 4223-4239. DOI: 10.1080/00207543.2012.753486
- [Haye17] B. Hayes, J.A. Shah: Improving Robot Controller Transparency Through Autonomous Policy Explanation. *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - Proceedings of the HRI '17: ACM/IEEE International Conference on Human-Robot Interaction* (2017): S. 303-312. DOI: 10.1145/2909824.3020233
- [Haye22] C.F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L.M. Zintgraf, R. Dazeley, F. Heintz, E. Howley, A.A. Irissappane, P. Mannion, A. Nowé, G. Ramos, M. Restelli, P. Vamplew, D.M. Roijers: A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems* 36/1 (2022). DOI: 10.1007/s10458-022-09552-y
- [Hees17] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S.M.A. Eslami, M. Riedmiller, D. Silver: Emergence of Locomotion Behaviours in Rich Environments. *arXiv*, (2017). DOI: 10.48550/arXiv.1707.02286
- [Hein23] B.T. Heinbach, P. Burggräf, J. Wagner: gym-flp: A Python Package for Training Reinforcement Learning Algorithms on Facility Layout Problems. (2023). DOI: 10.21203/rs.3.rs-371586/v1
- [Heis11] O.-E. Heiserich, K. Helbig, W. Ullmann: *Logistik – Eine praxisorientierte Einführung*. 4., vollst. überarb. und erw. Aufl. Gabler, (2011). ISBN: 978-3-8349-1852-9
- [Herr18] D. Herr, J. Reinhardt, G. Reina, R. Krüger, R.V. Ferrari, T. Ertl: Immersive Modular Factory Layout Planning using Augmented Reality. *Procedia CIRP* 72 (2018): S. 1112-1117. DOI: 10.1016/j.procir.2018.03.200
- [Hess17] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, D. Silver: Rainbow: Combining Improvements in Deep Reinforcement Learning. *arXiv*, (2017). DOI: 10.48550/arXiv.1710.02298
- [Heui20] A. Heuillet, F. Couthouis, N. Diaz-Rodríguez: Explainability in Deep Reinforcement Learning. *arXiv*, (2020). DOI: 10.48550/arXiv.2008.06693
- [Hick76] P.E. Hicks, T.E. Cowan: CRAFT-M for layout rearrangement. *Industrial Engineering* (1976): S. 30-35.
- [Hilb17] S. Hilbrich, H. Köck, J. Hinkeldeyn, J. Kreutzfeldt: Entwicklung einer Simulationsmethodik zur schnellen Dimensionierung komplexer Materialflusssysteme. *Wissenschaftliche Gesellschaft für Technische Logistik*, (2017). DOI: 10.2195/lj_Proc_hilbrich_de_201710_01
- [Homp18] M. ten Hompel, T. Schmidt, J. Dregger: *Materialflusssysteme*. Springer Berlin Heidelberg, (2018). DOI: 10.1007/978-3-662-56181-2
- [Hoss18] H. Hosseini-Nasab, S. Fereidouni, S.M.T. Fatemi Ghomi, M.B. Fakhrazad: Classification of facility layout problems: a review study. *The International Journal of Advanced Manufacturing Technology* 94/1-4 (2018): S. 957-977. DOI: 10.1007/s00170-017-0895-8
- [Hu22] X. Hu, Y.-F. Chuang: E-commerce warehouse layout optimization: systematic layout planning using a genetic algorithm. *Electronic Commerce Research* (2022). DOI: 10.1007/s10660-021-09521-9
- [Huan20] S. Huang, S. Ontañón: A Closer Look at Invalid Action Masking in Policy Gradient Algorithms. (2020). DOI: 10.48550/arXiv.2006.14171

- [Hube22] T. Huber, B. Limmer, E. André: Benchmarking Perturbation-Based Saliency Maps for Explaining Atari Agents. *Frontiers in artificial intelligence* 5 (2022): 903875. DOI: 10.3389/frai.2022.903875
- [Huo21] J. Huo, J. Liu, H. Gao: An NSGA-II Algorithm with Adaptive Local Search for a New Double-Row Model Solution to a Multi-Floor Hospital Facility Layout Problem. *Applied Sciences* 11/4 (2021): 1758. DOI: 10.3390/app11041758
- [Iked22] H. Ikeda, H. Nakagawa, T. Tsuchiya: Towards Automatic Facility Layout Design Using Reinforcement Learning. *Communication Papers of the 17th Conference on Computer Science and Intelligence Systems - Proceedings of the 17th Conference on Computer Science and Intelligence Systems* (2022): S. 11-20. DOI: 10.15439/2022F25
- [Ivan21] M. Ivanovs, R. Kadikis, K. Ozols: Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters* 150 (2021): S. 228-234. DOI: 10.1016/j.patrec.2021.06.030
- [Izad16] N. Izadinia, K. Eshghi: A robust mathematical model and ACO solution for multi-floor discrete layout problem with uncertain locations and demands. *Computers & Industrial Engineering* 96 (2016): S. 237-248. DOI: 10.1016/j.cie.2016.02.026
- [Jeri13] I. Jerin Leno, S. Saravana Sankar, S.G. Ponnambalam: An elitist strategy genetic algorithm using simulated annealing algorithm as local search for facility layout design. *The International Journal of Advanced Manufacturing Technology* (2013). DOI: 10.1007/s00170-013-5519-3
- [Jith10] I. Jithavech, K.K. Krishnan: A simulation-based approach for risk assessment of facility layout designs under stochastic product demands. *The International Journal of Advanced Manufacturing Technology* 49/1-4 (2010): S. 27-40. DOI: 10.1007/s00170-009-2380-5
- [Joha19] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J.A. Ojea, E. Solowjow, S. Levine: Residual Reinforcement Learning for Robot Control. *2019 International Conference on Robotics and Automation (ICRA) - Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)* (2019): S. 6023-6029. DOI: 10.1109/ICRA.2019.8794127
- [Kael96] L.P. Kaelbling, M.L. Littman, A.W. Moore: Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research* 4 (1996): S. 237-285. DOI: 10.1613/jair.301
- [Kamp14] A. Kampker, P. Burggräf, M. Krunke, K. Kreisköther, H. Voet, M. Backs: Das Aachener Fabrikplanungsvorgehen - Agile Fabrikplanung im turbulenten Umfeld. *Werkstattstechnik online* 104 (2014): 4.
- [Kerk14] T. Kerkenberg: Innovative und effiziente Fabrikplanung. *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 109/6 (2014): S. 435-438. DOI: 10.3139/104.111164
- [Kett84] H. Kettner, J. Schmidt, H.-R. Greim: *Leitfaden der systematischen Fabrikplanung*. Hanser, (1984). ISBN: 978-3-446-13825-4
- [Khan13] A. Khan, D. Tidke: Designing Facilities Layout for Small and Medium Enterprises. *International Journal of Engineering Research and General Science* 1 (2013): 2.
- [Kim00] J.-G. Kim, Y.-D. Kim: Layout planning for facilities with fixed shapes and input and output points. *International Journal of Production Research* 38/18 (2000): S. 4635-4653. DOI: 10.1080/00207540050205550
- [Kipf16] T.N. Kipf, M. Welling: Semi-Supervised Classification with Graph Convolutional Networks. *arXiv*, (2016). DOI: 10.48550/arXiv.1609.02907
- [Klar23a] M. Klar, M. Glatt, J.C. Aurich: Performance comparison of reinforcement learning and metaheuristics for factory layout planning. *CIRP Journal of Manufacturing Science and Technology* 45 (2023): S. 10-25. DOI: 10.1016/j.cirpj.2023.05.008
- [Klar21] M. Klar, M. Glatt, J.C. Aurich: An implementation of a reinforcement learning based algorithm for factory layout planning. *Manufacturing Letters* 30 (2021): S. 1-4. DOI: 10.1016/j.mfglet.2021.08.003
- [Klar23b] M. Klar, M. Glatt, B. Ravani, J.C. Aurich: A simulation-based factory layout planning approach using reinforcement learning. *Procedia CIRP* 120 (2023). DOI: 10.1016/j.procir.2023.08.023
- [Klar22a] M. Klar, M. Hussong, P. Ruediger-Flore, L. Yi, M. Glatt, J.C. Aurich: Scalability investigation of Double Deep Q Learning for factory layout planning. *Procedia CIRP* 107 (2022): S. 161-166. DOI: 10.1016/j.procir.2022.04.027

- [Klar24a] M. Klar, A. Jawaid, P. Simon, M. Wagner, J. Seewig, B. Ravani, J.C. Aurich: Performance Comparison of Single-Agent and Multi-Agent Reinforcement Learning for Factory Layout Planning (accepted). *Procedia CIRP* 114 (2024).
- [Klar22b] M. Klar, P. Langlotz, J.C. Aurich: A Framework for Automated Multiobjective Factory Layout Planning using Reinforcement Learning. *Procedia CIRP* 112 (2022): S. 555-560. DOI: 10.1016/j.procir.2022.09.099
- [Klar23c] M. Klar, J. Mertes, M. Glatt, B. Ravani, J.C. Aurich: A Holistic Framework for Factory Planning Using Reinforcement Learning. (2023): S. 129-148. DOI: 10.1007/978-3-031-35779-4_8
- [Klar24b] M. Klar, P. Ruediger, M. Schuermann, G.T. Gören, M. Glatt, B. Ravani, J.C. Aurich: Explainable generative design in manufacturing for reinforcement learning based factory layout planning. *Journal of Manufacturing Systems* 72 (2024): S. 74-92. DOI: 10.1016/j.jmsy.2023.11.012
- [Klar24d] M. Klar, P. Schworm, X. Wu, P. Simon, M. Glatt, B. Ravani, J.C. Aurich: Transferable multi-objective factory layout planning using simulation-based deep reinforcement learning. *Journal of Manufacturing Systems* (2024). DOI: 10.1016/j.jmsy.2024.04.007
- [Klar24e] M. Klar, P.M. Simon, M. Glatt, B. Ravani, J.C. Aurich: Simulation-based Brownfield Factory Planning Using Deep Reinforcement Learning (under Review). *Manufacturing Letters* (2024).
- [Kram17] O. Kramer: *Genetic Algorithms* 679 (2017): S. 11-19. DOI: 10.1007/978-3-319-52156-5_2
- [Krüg19] T. Krüger: *Entwicklung einer Gesamtmethodik zur Kombination von mathematischer Anordnungsoptimierung und Materialflusssimulation für die Produktionslayoutplanung*, Universitätsbibliothek der TU Clausthal, (2019).
- [Kühn06] W. Kühn: *Digitale Fabrik – Fabriksimulation für Produktionsplaner*. Hanser, (2006). ISBN: 978-3-446-40619-3
- [Kult07] S. Kulturel-Konak: Approaches to uncertainties in facility layout problems: Perspectives at the beginning of the 21st Century. *Journal of Intelligent Manufacturing* 18/2 (2007): S. 273-284. DOI: 10.1007/s10845-007-0020-1
- [Kusi87] A. Kusiak, S.S. Heragu: The facility layout problem. *European Journal of Operational Research* 29/3 (1987): S. 229-251. DOI: 10.1016/0377-2217(87)90238-4
- [Laar87] P.J.M. van Laarhoven, E.H.L. Aarts: *Simulated Annealing: Theory and Applications*. (1987). DOI: 10.1007/978-94-015-7744-1
- [Lahm05] M. Lahmar, S. Benjaafar: Design of distributed layouts. *IIE Transactions* 37/4 (2005): S. 303-318. DOI: 10.1080/07408170590517015
- [Lai15] M. Lai: *Giraffe: Using Deep Reinforcement Learning to Play Chess*. arXiv, (2015). DOI: 10.48550/arXiv.1509.01549
- [Law13] A.M. Law: *Simulation modeling and analysis*. Fifth edition. McGraw-Hill Education, (2013). ISBN: 978-0-07-340132-4
- [LeCu15] Y. LeCun, Y. Bengio, G. Hinton: Deep learning. *Nature* 521/7553 (2015): S. 436-444. DOI: 10.1038/nature14539
- [Lee12] H.-Y. Lee: Integrating Simulation and Ant Colony Optimization to Improve the Service Facility Layout in a Station. *Journal of Computing in Civil Engineering* 26/2 (2012): S. 259-269. DOI: 10.1061/(ASCE)CP.1943-5487.0000128
- [Lee67] R.C. Lee, J.M. Moore: CORELAP--Computerized Relationship Layout Planning. *Journal of Industrial Engineering* 18 (1967): 3.
- [Li23] C. Li, P. Zheng, Y. Yin, B. Wang, L. Wang: Deep reinforcement learning in smart manufacturing: A review and prospects. *CIRP Journal of Manufacturing Science and Technology* 40 (2023): S. 75-101. DOI: 10.1016/j.cirpj.2022.11.003
- [Li17] Y. Li: *Deep Reinforcement Learning: An Overview*. arXiv, (2017). DOI: 10.48550/arXiv.1701.07274
- [Lian08] L.Y. Liang, W.C. Chao: The strategies of tabu search technique for facility layout optimization. *Automation in Construction* 17/6 (2008): S. 657-669. DOI: 10.1016/j.autcon.2008.01.001
- [Lieb16] T. Liebethuth: *Prozessmanagement in Einkauf und Logistik*. Springer Fachmedien Wiesbaden, (2016). DOI: 10.1007/978-3-658-09759-2

- [Lim09] B.Y. Lim, A.K. Dey, D. Avrahami: Why and why not explanations improve the intelligibility of context-aware intelligent systems. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - Proceedings of the CHI '09: CHI Conference on Human Factors in Computing Systems (2009): S. 2119-2128. DOI: 10.1145/1518701.1519023
- [Lipt16] Z.C. Lipton: The Mythos of Model Interpretability. arXiv, (2016). DOI: 10.48550/arXiv.1606.03490
- [Lyu19] D. Lyu, F. Yang, B. Liu, S. Gustafson: SDRL: Interpretable and Data-Efficient Deep Reinforcement Learning Leveraging Symbolic Planning. Proceedings of the AAAI Conference on Artificial Intelligence 33/01 (2019): S. 2970-2977. DOI: 10.1609/aaai.v33i01.33012970
- [Mani11] K.D. Maniya, M.G. Bhatt: An alternative multiple attribute decision making methodology for solving optimal facility layout design selection problems. Computers & Industrial Engineering 61/3 (2011): S. 542-549. DOI: 10.1016/j.cie.2011.04.009
- [Mart11] H. Martin: Transport- und Lagerlogistik. Vieweg+Teubner, (2011). DOI: 10.1007/978-3-8348-9242-3
- [Matt89] F. Mattern, H. Mehl: Diskrete Simulation - Prinzipien und Probleme der Effizienzsteigerung durch Parallelisierung. Informatik Spektrum Berlin 12 (1989): 4.
- [Mazy21] N. Mazyavkina, S. Sviridov, S. Ivanov, E. Burnaev: Reinforcement learning for combinatorial optimization: A survey. Computers & Operations Research 134 (2021): 105400. DOI: 10.1016/j.cor.2021.105400
- [McKe06a] A.R. McKendall, J. Shang: Hybrid ant systems for the dynamic facility layout problem. Computers & Operations Research 33/3 (2006): S. 790-803. DOI: 10.1016/j.cor.2004.08.008
- [McKe06b] A.R. McKendall, J. Shang, S. Kuppasamy: Simulated annealing heuristics for the dynamic facility layout problem. Computers & Operations Research 33/8 (2006): S. 2431-2444. DOI: 10.1016/j.cor.2005.02.021
- [Meck15] A. Meckelnborg: Integrative Fabrikplanung durch effiziente Koordinationsmodelle. 1. Aufl. Apprimus Verlag, (2015). ISBN: 9783863593292
- [Mell96] R.D. Meller, K.-Y. Gau: The facility layout problem: Recent and emerging trends and perspectives. Journal of Manufacturing Systems 15/5 (1996): S. 351-366. DOI: 10.1016/0278-6125(96)84198-7
- [Mila22] S. Milani, N. Topin, M. Veloso, F. Fang: A Survey of Explainable Reinforcement Learning. arXiv, (2022). DOI: 10.48550/arXiv.2202.08434
- [Mill94] G.A. Miller: The magical number seven, plus or minus two: some limits on our capacity for processing information. 1956. Psychological review 101/2 (1994): S. 343-352. DOI: 10.1037/0033-295x.101.2.343
- [Mins61] M. Minsky: Steps toward Artificial Intelligence. Proceedings of the IRE 49/1 (1961): S. 8-30. DOI: 10.1109/JRPROC.1961.287775
- [Mnih16] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T.P. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu: Asynchronous Methods for Deep Reinforcement Learning. (2016). DOI: 10.48550/arXiv.1602.01783
- [Mnih13] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller: Playing Atari with Deep Reinforcement Learning. (2013).
- [Mont18] G. Montavon, W. Samek, K.-R. Müller: Methods for interpreting and understanding deep neural networks. Digital Signal Processing 73 (2018): S. 1-15. DOI: 10.1016/j.dsp.2017.10.011
- [Moss16] H. Mossalam, Y.M. Assael, D.M. Roijers, S. Whiteson: Multi-Objective Deep Reinforcement Learning. arXiv, (2016). DOI: 10.48550/arXiv.1610.02707
- [Muth99] R. Muther, L. Hales: Systematic planning of industrial facilities – S.P.I.F. 3rd ed. Management and Industrial Research Publishers, (1999). ISBN: 9780933684003
- [Muth20] N. Muthukrishnan, F. Maleki, K. Ovens, C. Reinhold, B. Forghani, R. Forghani: Brief History of Artificial Intelligence. Neuroimaging clinics of North America 30/4 (2020): S. 393-399. DOI: 10.1016/j.nic.2020.07.004
- [Nara99] S. Narayanan, S. Azarm: On improving multiobjective genetic algorithms for design optimization. Structural Optimization 18/2-3 (1999): S. 146-155. DOI: 10.1007/BF01195989

- [Nee12] A. Nee, S.K. Ong, G. Chrysosolouris, D. Mourtzis: Augmented reality applications in design and manufacturing. *CIRP Annals* 61/2 (2012): S. 657-679. DOI: 10.1016/j.cirp.2012.05.010
- [Niem21] J. Niemann, B. Reich, C. Stöhr: Spaghetti-Diagramm. (2021): S. 121-129. DOI: 10.1007/978-3-662-63008-2_7
- [Nöck12] J.C. Nöcker: Zustandsbasierte Fabrikplanung. 1. Aufl. Apprimus-Verl., (2012). ISBN: 978-3-86359-059-8
- [Nuge68] C.E. Nugent, T.E. Vollmann, J. Ruml: An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research* 16 (1968): S. 150-173.
- [O'Sh15] K. O'Shea, R. Nash: An Introduction to Convolutional Neural Networks. arXiv, (2015). DOI: 10.48550/arXiv.1511.08458
- [Page91] B. Page: Diskrete Simulation. Springer Berlin Heidelberg, (1991). DOI: 10.1007/978-3-642-76862-0
- [Pan10] S.J. Pan, Q. Yang: A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22/10 (2010): S. 1345-1359. DOI: 10.1109/TKDE.2009.191
- [Papa87] C.H. Papadimitriou, J.N. Tsitsiklis: The Complexity of Markov Decision Processes. *Mathematics of Operations Research* 12 (1987): 3.
- [Pawe14] G. Pawellek: Ganzheitliche Fabrikplanung – Grundlagen, Vorgehensweise, EDV-Unterstützung. 2. Aufl. Springer-Vieweg, (2014). DOI: 10.1007/978-3-662-43728-5
- [Pegd85] C.D. Pegden: Introduction to SIMAN. *Proceedings of the 17th conference on Winter simulation - WSC '85 - Proceedings of the the 17th conference* (1985): S. 66-72. DOI: 10.1145/21850.253077
- [Pére21] P. Pérez-Gosende, J. Mula, M. Díaz-Madroño: Facility layout planning. An extended literature review. *International Journal of Production Research* 59/12 (2021): S. 3777-3816. DOI: 10.1080/00207543.2021.1897176
- [Pour17] M.R. Pourhassan, S. Raissi: An integrated simulation-based optimization technique for multi-objective dynamic facility layout problem. *Journal of Industrial Information Integration* 8 (2017): S. 49-58. DOI: 10.1016/j.jii.2017.06.001
- [Pour14] H. Pourvaziri, B. Naderi: A hybrid multi-population genetic algorithm for the dynamic facility layout problem. *Applied Soft Computing* 24 (2014): S. 457-469. DOI: 10.1016/j.asoc.2014.06.051
- [Prit95] A.A.B. Pritsker: Introduction to simulation and SLAM II. 4th ed. Wiley, (1995). ISBN: 978-0470234570
- [Pui20] E. Puiutta, E.M. Veith: Explainable Reinforcement Learning: A Survey. arXiv, (2020). DOI: 10.48550/arXiv.2005.06247
- [Rama09] D. Raman, S.V. Nagalingam, B.W. Gurd: A genetic algorithm and queuing theory based methodology for facilities layout problem. *International Journal of Production Research* 47/20 (2009): S. 5611-5635. DOI: 10.1080/00207540802057352
- [REFA90] REFA: Planung und Gestaltung komplexer Produktionssysteme. 2. Aufl., 6. - 10. Tsd. Hanser, (1990). ISBN: 3446159673
- [Regg11] T. Reggelin: Mesoskopische Modellierung und Simulation logistischer Flusssysteme. docupoint, (2011). ISBN: 978-3-940961-63-1
- [Robi14] S. Robinson: Simulation – The practice of model development and use. 2. ed. Palgrave Macmillan, (2014). ISBN: 978-1-137-32802-1
- [Robi04] S. Robinson: Simulation – The practice of model development and use. Wiley, (2004). ISBN: 0-470-84772-7
- [Rock63] W. Rockstroh: Zur Projektierung von Industriebetrieben Dissertation. Dresden: Technische Universität, Fakultät für Technologie, (1963).
- [Rose58] F. Rosenblatt: The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* 65/6 (1958): S. 386-408.
- [Rubi16] M. Rubio-Sánchez, M. Gallego, F. Gortázar, A. Duarte: GRASP with path relinking for the single row facility layout problem. *Knowledge-Based Systems* 106 (2016): S. 1-13. DOI: 10.1016/j.knosys.2016.05.030

- [Rute89] R.A. Rutenbar: Simulated annealing algorithms: an overview. IEEE Circuits and Devices Magazine 5/1 (1989): S. 19-26. DOI: 10.1109/101.17235
- [Saat89] T.L. Saaty: Group Decision Making and the AHP. (1989): S. 59-67. DOI: 10.1007/978-3-642-50244-6_4
- [Sae23] W. Saeed, C. Omlin: Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. Knowledge-Based Systems 263 (2023): 110273. DOI: 10.1016/j.knsys.2023.110273
- [Şahi10] R. Şahin, K. Ertoğral, O. Türkbey: A simulated annealing heuristic for the dynamic layout problem with budget constraint. Computers & Industrial Engineering 59/2 (2010): S. 308-313. DOI: 10.1016/j.cie.2010.04.013
- [Sale22] A. Salem, H. Hegab, S. Rahnamayan, H.A. Kishawy: Multi-objective optimization and innovation-based knowledge discovery of sustainable machining process. Journal of Manufacturing Systems 64 (2022): S. 636-647. DOI: 10.1016/j.jmsy.2022.04.013
- [Salm15] M.H. Salmani, K. Eshghi, H. Neghabi: A bi-objective MIP model for facility layout problem in uncertain environment. The International Journal of Advanced Manufacturing Technology 81/9-12 (2015): S. 1563-1575. DOI: 10.1007/s00170-015-7290-0
- [Sama10] H. Samarghandi, P. Taabayan, F.F. Jahantigh: A particle swarm optimization for the single row facility layout problem. Computers & Industrial Engineering 58/4 (2010): S. 529-534. DOI: 10.1016/j.cie.2009.11.015
- [Samu59] A.L. Samuel: Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development 3/3 (1959): S. 210-229. DOI: 10.1147/rd.33.0210
- [Scar99] M. Scardi, L.W. Harding: Developing an empirical model of phytoplankton primary production: a neural network case study. Ecological Modelling 120/2-3 (1999): S. 213-223. DOI: 10.1016/S0304-3800(99)00103-9
- [Scar09] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini: The graph neural network model. IEEE transactions on neural networks 20/1 (2009): S. 61-80. DOI: 10.1109/TNN.2008.2005605
- [Scha15] T. Schaul, J. Quan, I. Antonoglou, D. Silver: Prioritized Experience Replay. arXiv, (2015). DOI: 10.48550/arXiv.1511.05952
- [Schm96] H. Schmigalla: Fabrikplanung: Begriffe und Zusammenhänge. Hanser Verlag, (1996).
- [Schm70] H. Schmigalla: Methoden zur optimalen Maschinenanordnung. Technik, (1970).
- [Schn91] C. Schneeweiss: Der Analytic Hierarchy Process als spezielle Nutzwertanalyse. (1991): S. 183-195. DOI: 10.1007/978-3-642-76537-7_13
- [Schn23] J. Schneidewind, S. Galka: Interactive Reinforcement Learning-Based Factory Layout Planning. SSRN Electronic Journal (2023). DOI: 10.2139/ssrn.4469300
- [Schu09] G. Schuh, B. Franzkoch, P. Burggräf, J.C. Nöcker, C. Wesch-Potente: Frei konfigurierbare Planungsprozesse in der Fabrikplanung. Werkstattstechnik online 99 (2009): 4.
- [Schu07] G. Schuh, S. Gottschalk, F. Lösch, C. Wesch: Fabrikplanung im Gegenstromverfahren. Werkstattstechnik online 97 (2007): 4.
- [Schu11] G. Schuh, A. Kampker, C. Wesch-Potente: Condition based factory planning. Production Engineering 5/1 (2011): S. 89-94. DOI: 10.1007/s11740-010-0281-y
- [Schu15] J. Schulman, P. Moritz, S. Levine, M. Jordan, P. Abbeel: High-Dimensional Continuous Control Using Generalized Advantage Estimation. arXiv, (2015). DOI: 10.48550/arXiv.1506.02438
- [Schu17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov: Proximal Policy Optimization Algorithms. arXiv, (2017). DOI: 10.48550/arXiv.1707.06347
- [Schu10] C.P. Schulze, C. Reinema, P. Nyhuis: Merkmalsystemaizur zur Entwicklung von Fabrikstrukturvarianten. ZWF-Zeitschrift für Wirtschaftlichen Fabrikbetrieb 105 (2010): 9.
- [Schü98] R. Schütte: Grundsätze ordnungsmäßiger Referenzmodellierung. Gabler Verlag, (1998). DOI: 10.1007/978-3-663-10233-5
- [Seid22] T. Seidelmann, S. Mostaghim: Finding Cost-Effective Re-Laying Solutions in Modern Brownfield Facility Layout Planning. 2022 IEEE Congress on Evolutionary Computation (CEC) -

- Proceedings of the 2022 IEEE Congress on Evolutionary Computation (CEC) (2022): S. 1-8. DOI: 10.1109/CEC55065.2022.9870345
- [Silv16] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis: Mastering the game of Go with deep neural networks and tree search. *Nature* 529/7587 (2016): S. 484-489. DOI: 10.1038/nature16961
- [Sing06] S.P. Singh, R.R.K. Sharma: A review of different approaches to the facility layout problems. *The International Journal of Advanced Manufacturing Technology* 30/5-6 (2006): S. 425-433. DOI: 10.1007/s00170-005-0087-9
- [Sinn20] C. Sinnwell: Methode zur Produktionssystemkonzipierung auf Basis früher Produktinformationen. Dissertation, Universität Kaiserslautern, (2020).
- [Soli08] M. Solimanpur, A. Jafari: Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. *Computers & Industrial Engineering* 55/3 (2008): S. 606-619. DOI: 10.1016/j.cie.2008.01.018
- [Song14] X. Song, Y. Wang, C. Feng: Multi-objective Dynamic Layout Problems for Unequal-Area Workshop Facilities Based on NSGA-II 242 (2014): S. 1511-1520. DOI: 10.1007/978-3-642-40081-0_129
- [Sout21] G. Souto, I. Morais, G.R. Mauri, G.M. Ribeiro, P.H. González: A hybrid matheuristic for the Two-Stage Capacitated Facility Location problem. *Expert Systems with Applications* 185 (2021): 115501. DOI: 10.1016/j.eswa.2021.115501
- [Srid19] M. Sridharan, B. Meadows: Towards a Theory of Explanations for Human–Robot Collaboration. *KI - Künstliche Intelligenz* 33/4 (2019): S. 331-342. DOI: 10.1007/s13218-019-00616-y
- [Ster00] J.D. Sterman: Business dynamics – Systems thinking and modeling for a complex world. Irwin McGraw-Hill, (2000). ISBN: 978-0072389159
- [Süße21] M. Süße, M. Putz: Generative design in factory layout planning. *Procedia CIRP* 99 (2021): S. 9-14. DOI: 10.1016/j.procir.2021.03.002
- [Sutt88] R.S. Sutton: Learning to predict by the methods of temporal differences. *Machine Learning* 3/1 (1988): S. 9-44. DOI: 10.1007/BF00115009
- [Sutt00] R.S. Sutton, D. McAllester, S. Singh, >. Mansour: Policy Gradient Methods for Reinforcement Learning with Function Approximation. *Advances in Neural Information Processing Systems MIT Press* 12 (2000): S. 1057-1063.
- [Sutt18] R.S. Sutton, A. Barto: Reinforcement learning – An introduction. Second edition. The MIT Press, (2018). ISBN: 9780262352703
- [Szep10] C. Szepesvári: Algorithms for Reinforcement Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 4/1 (2010): S. 1-103. DOI: 10.2200/S00268ED1V01Y201005AIM009
- [Tajm18] T. Tajmager: Modular Multi-Objective Deep Reinforcement Learning with Decision Values. *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems - Proceedings of the 2018 Federated Conference on Computer Science and Information Systems* (2018): S. 85-93. DOI: 10.15439/2018F231
- [Todo08] E. Todorov: General duality between optimal control and estimation. 2008 47th IEEE Conference on Decision and Control - Proceedings of the 2008 47th IEEE Conference on Decision and Control (2008): S. 4286-4292. DOI: 10.1109/CDC.2008.4739438
- [Tomp10] J. Tompkins, J.A. White, Y.A. Bozer: Facilities planning. 4th edition. Wiley, (2010). ISBN: 978-0-470-44404-7
- [Torr10] L. Torrey, J. Shavlik: Transfer Learning. (2010): S. 242-264. DOI: 10.4018/978-1-60566-766-9.ch011
- [Turg18] S. Turgay: Multi Objective Simulated Annealing Approach for Facility Layout Design. *International Journal of Mathematical, Engineering and Management Sciences* 3/4 (2018): S. 365-380. DOI: 10.33889/IJMEMS.2018.3.4-026

- [Tuzk13] G. Tuzkaya, B. Gülsün, U.R. Tuzkaya, S. Onut, E. Bildik: A comparative analysis of meta-heuristic approaches for facility layout design problem: a case study for an elevator manufacturer. *Journal of Intelligent Manufacturing* 24/2 (2013): S. 357-372. DOI: 10.1007/s10845-011-0599-0
- [Ulut15] B. Ulutas, A.A. Islier: Dynamic facility layout problem in footwear industry. *Journal of Manufacturing Systems* 36 (2015): S. 55-61. DOI: 10.1016/j.jmsy.2015.03.004
- [Ulut12] B. Ulutas, T. Sarac: Determining the parameters of MSG algorithm for multi period layout problem. *Journal of Manufacturing Technology Management* 23/7 (2012): S. 922-936. DOI: 10.1108/17410381211267736
- [Unge21] H. Unger, F. Börner: Reinforcement Learning for Layout Planning – Modelling the Layout Problem as MDP 632 (2021): S. 471-479. DOI: 10.1007/978-3-030-85906-0_52
- [Urba09] F.K. Urban, A. Braune: PDCA als Methode der qualitäts- und zielorientierten Fabrikplanung. *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 104/1-2 (2009): S. 60-63. DOI: 10.3139/104.110011
- [VDI 07] VDI: Kosten des Materialflusses. (2007).
- [VDI 99] VDI: Methoden zur materialflußgerechten Zuordnung von Betriebsbereichen und -mitteln. (1999).
- [VDI 14a] VDI: Simulation von Logistik-, Materialfluss- und Produktionssystemen- Grundlagen. (2014).
- [VDI 14b] VDI: Simulation von Logistik-, Materialfluss- und Produktionssystemen - Simulation und Optimierung. (2014).
- [VDI 08] VDI: Digitale Fabrik - Grundlagen. (2008).
- [VDI 11a] VDI: Fabrikplanung - Planungsvorgehen. (2011).
- [VDI 11b] VDI: Fabrikplanung - Morphologisches Modell der Fabrik zur Zielfestlegung in der Fabrikplanung. (2011).
- [Veli17] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio: Graph Attention Networks. *arXiv*, (2017). DOI: 10.48550/arXiv.1710.10903
- [Verm18] A. Verma, V. Murali, R. Singh, P. Kohli, S. Chaudhuri: Programmatically Interpretable Reinforcement Learning. (2018). DOI: 10.48550/arXiv.1804.02477
- [Vite21] D. Viteri-Viteri, A. Rodríguez-Luna, S. Avilés-Sacoto, G. Mosquera-Recalde: Using Simulation for Facility Layout Problems: A Case Study of an Ecuadorian Handcraft Chocolate Company. (2021): S. 389-434. DOI: 10.1007/978-3-030-71579-3_17
- [Walt65] M. Waltz, K. Fu: A heuristic approach to reinforcement learning control systems. *IEEE Transactions on Automatic Control* 10/4 (1965): S. 390-398. DOI: 10.1109/TAC.1965.1098193
- [Wang15] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, N. de Freitas: Dueling Network Architectures for Deep Reinforcement Learning. *arXiv*, (2015). DOI: 10.48550/arXiv.1511.06581
- [Warn95] H.-J. Warnecke: Der Produktionsbetrieb 1 – Organisation, Produkt, Planung. Dritte, unveränderte Auflage. Springer Berlin Heidelberg, (1995). DOI: 10.1007/978-3-642-79239-7
- [Watk92] C.J.C.H. Watkins, P. Dayan: Q-learning. *Machine Learning* 8/3-4 (1992): S. 279-292. DOI: 10.1007/BF00992698
- [Weis16] K. Weiss, T.M. Khoshgoftaar, D. Wang: A survey of transfer learning. *Journal of Big Data* 3/1 (2016). DOI: 10.1186/s40537-016-0043-6
- [Well21] L. Wells, T. Bednarz: Explainable AI and Reinforcement Learning-A Systematic Review of Current Approaches and Trends. *Frontiers in artificial intelligence* 4 (2021): 550030. DOI: 10.3389/frai.2021.550030
- [Wien14a] H.-P. Wiendahl, J. Reichardt, P. Nyhuis: Handbuch Fabrikplanung – Konzept, Gestaltung und Umsetzung wandlungsfähiger Produktionsstätten. 2., überarb. und erw. Aufl. Hanser, (2014). ISBN: 978-3-446-43702-9
- [Wien14b] H.-P. Wiendahl, J. Reichardt, P. Nyhuis: Synergetische Fabrikplanung. (2014): S. 441-532. DOI: 10.3139/9783446437029.015
- [Wier12] M. Wiering, M. van Otterlo: Reinforcement Learning. Springer Berlin Heidelberg, (2012). DOI: 10.1007/978-3-642-27645-3
- [Wu01] J. Wu, S. Azarm: Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set. *Journal of Mechanical Design* 123/1 (2001): S. 18-25. DOI: 10.1115/1.1329875

- [Xie08] W. Xie, N.V. Sahinidis: A branch-and-bound algorithm for the continuous facility layout problem. *Computers & Chemical Engineering* 32/4-5 (2008): S. 1016-1028. DOI: 10.1016/j.compchemeng.2007.05.003
- [Yang13] L. Yang, J. Deuse, P. Jiang: Multiple-attribute decision-making approach for an energy-efficient facility layout design. *The International Journal of Advanced Manufacturing Technology* 66/5-8 (2013): S. 795-807. DOI: 10.1007/s00170-012-4367-x
- [Yang20] Q. Yang, Y. Zhang, W. Dai, S.J. Pan: *Transfer learning*. Cambridge University Press, (2020). DOI: 10.1017/9781139061773
- [Yang05] T. Yang, B.A. Peters, M. Tu: Layout design for flexible manufacturing systems considering single-loop directional flow patterns. *European Journal of Operational Research* 164/2 (2005): S. 440-455. DOI: 10.1016/j.ejor.2003.04.004
- [Ye23] X. Ye, Z. Deng, Y. Shi, W. Shen: Toward Energy-Efficient Routing of Multiple AGVs with Multi-Agent Reinforcement Learning. *Sensors (Basel, Switzerland)* 23/12 (2023). DOI: 10.3390/s23125615
- [Zeil13] M.D. Zeiler, R. Fergus: Visualizing and Understanding Convolutional Networks. *arXiv*, (2013). DOI: 10.48550/arXiv.1311.2901
- [Zhao20] Y. Zhao, J. Lu, Q. Yan, L. Lai, L. Xu: Research on Cell Manufacturing Facility Layout Problem Based on Improved NSGA-II. *Computers, Materials & Continua* 62/1 (2020): S. 355-364. DOI: 10.32604/cmc.2020.06396
- [Zhu18] T. Zhu, J. Balakrishnan, C.H. Cheng: Recent advances in dynamic facility layout research. *INFOR: Information Systems and Operational Research* 56/4 (2018): S. 428-456. DOI: 10.1080/03155986.2017.1363591

Verzeichnis im Themenbereich betreuter Studien- und Diplomarbeiten:

Gerste, Jan	Entwicklung einer Bewertungsmethodik für die Groblayoutplanung, 2021.
Gören, Tobias	Implementation of an explainable reinforcement learning method for factory layout planning, 2022.
Pastler, Boris	Implementierung eines Actor-Critic Modells für die Fabriklayoutplanung, 2022.
Nale, Prashanta	Entwicklung eines Multiagent Reinforcement Learning Ansatz für die Fabriklayoutplanung, 2023.
Schultz, Benedikt	Untersuchung der Leistungsfähigkeit von Reinforcement Learning basierter Fabriklayoutplanung unter Verwendung unterschiedlicher Replay Buffer, 2023.

11 Anhang

11.1 Anhang A: Schematischer Aufbau des Simulationsmoduls

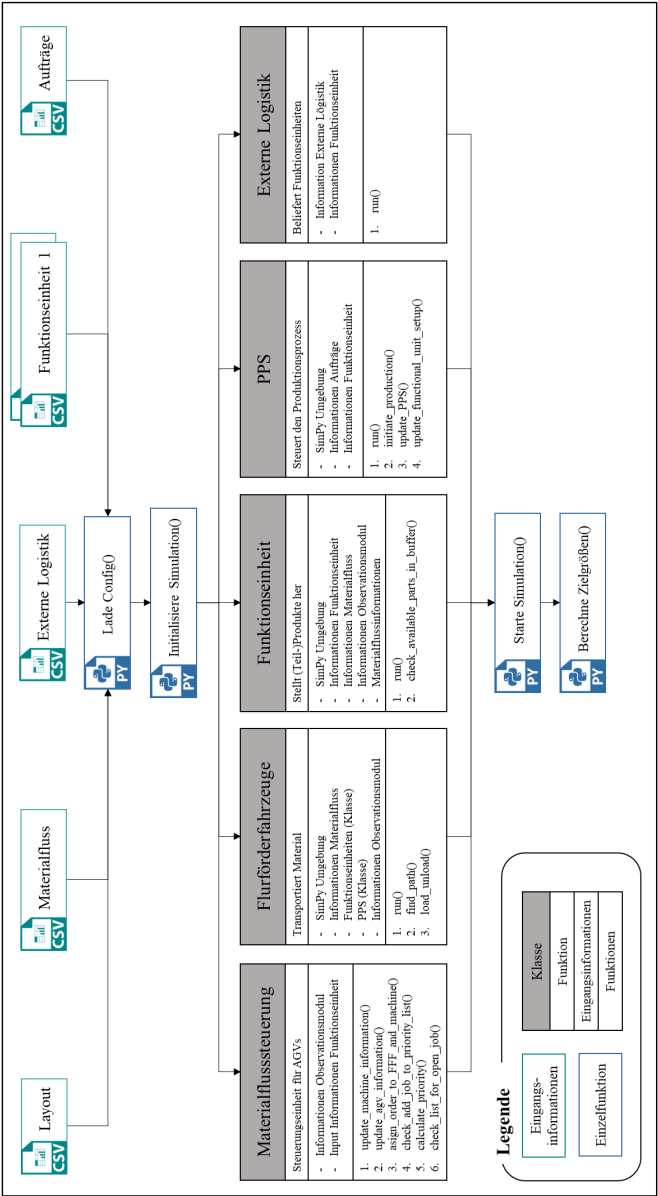


Abbildung 11-1: Schematischer Aufbau des Simulationsmoduls

11.2 Anhang B: Abbildungen zur Anwendung des RL-Ansatzes auf ein Neuplanungsproblem

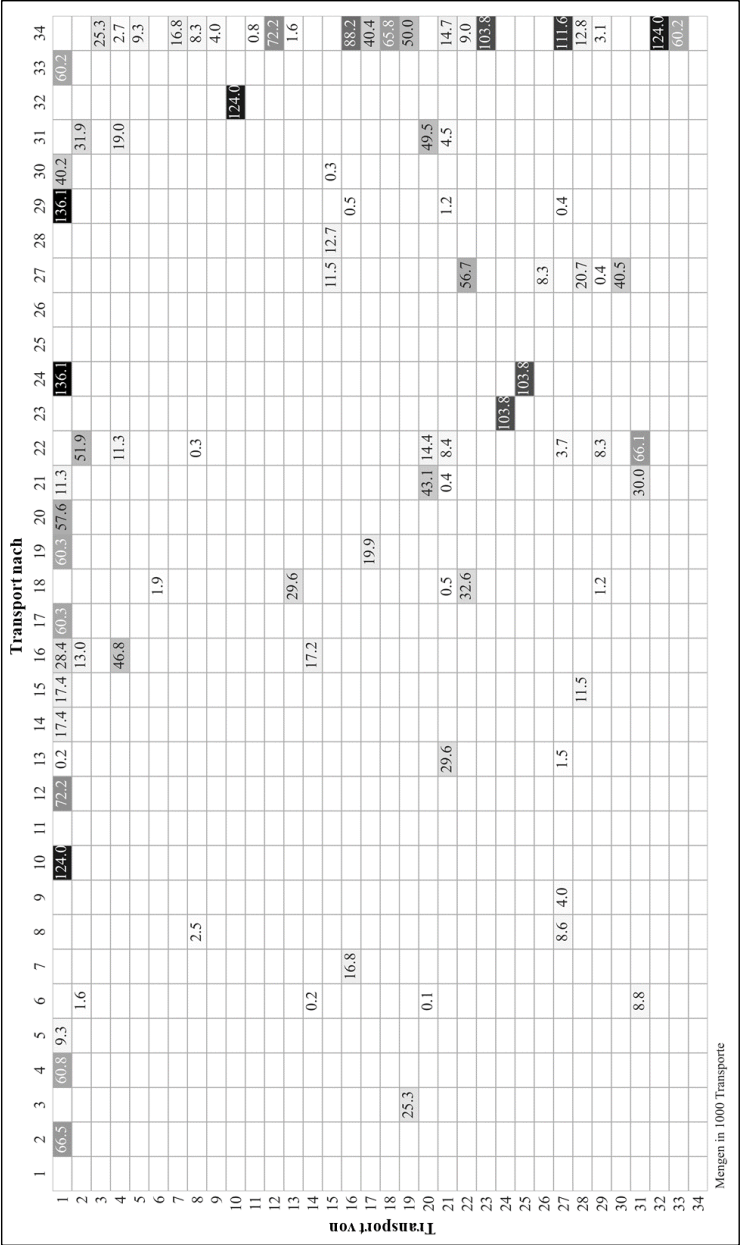


Abbildung 11-2: Transportmatrix des Neuplanungsproblems

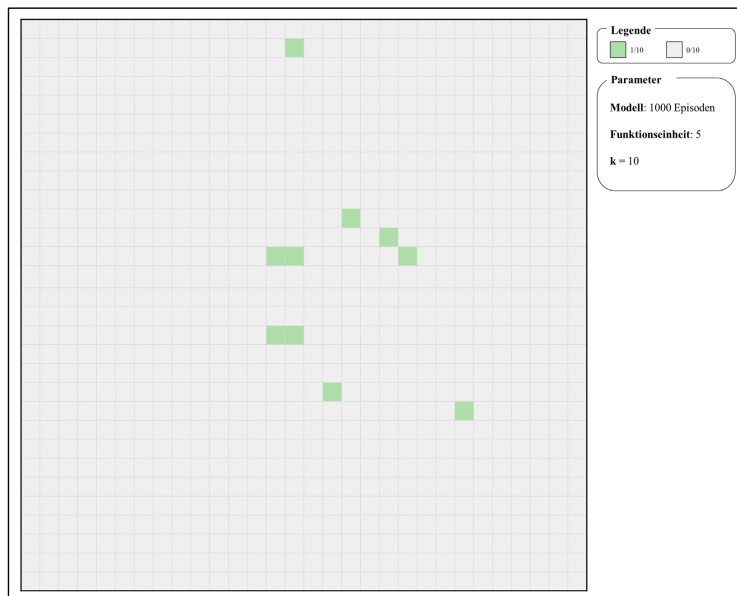


Abbildung 11-3: Ergebnisse der Verteilungsanalyse für Funktionseinheit 5 nach 1000 Episoden

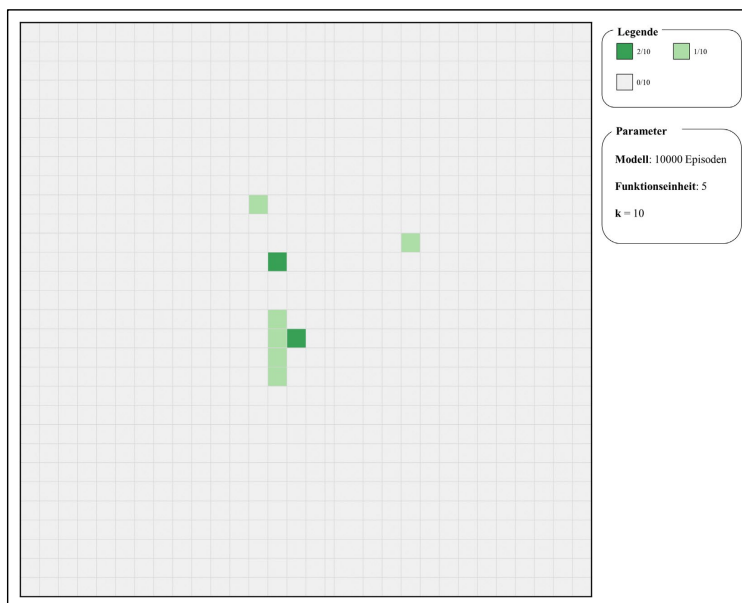


Abbildung 11-4: Ergebnisse der Verteilungsanalyse für Funktionseinheit 5 nach 10000 Episoden

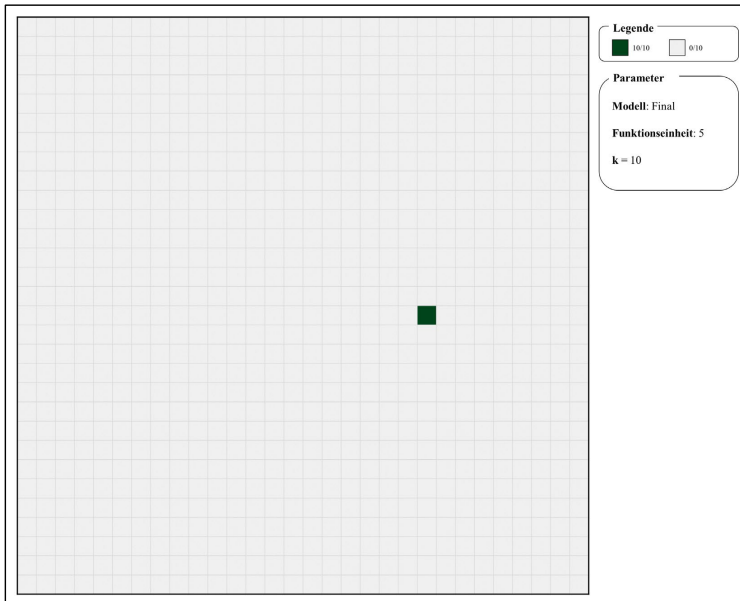


Abbildung 11-5: Ergebnisse der Verteilungsanalyse für Funktionseinheit 5 für das finale Modell

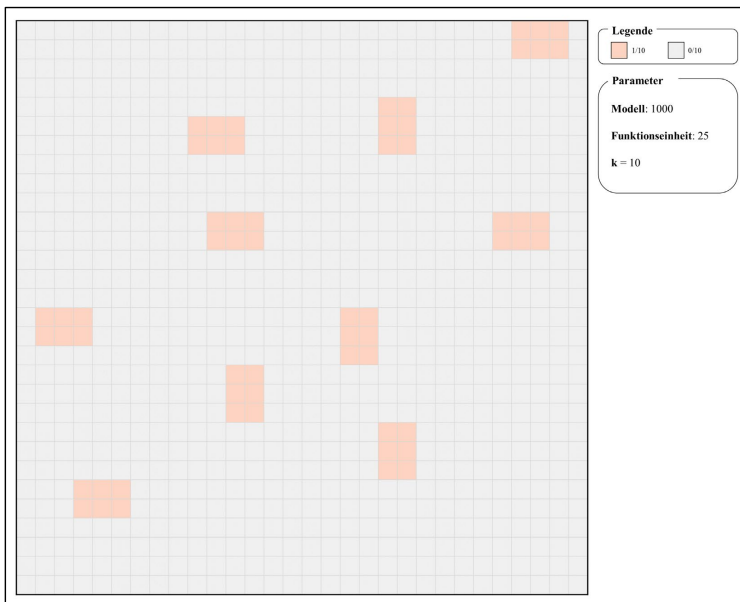


Abbildung 11-6: Ergebnisse der Verteilungsanalyse für Funktionseinheit 25 nach 1000 Episoden

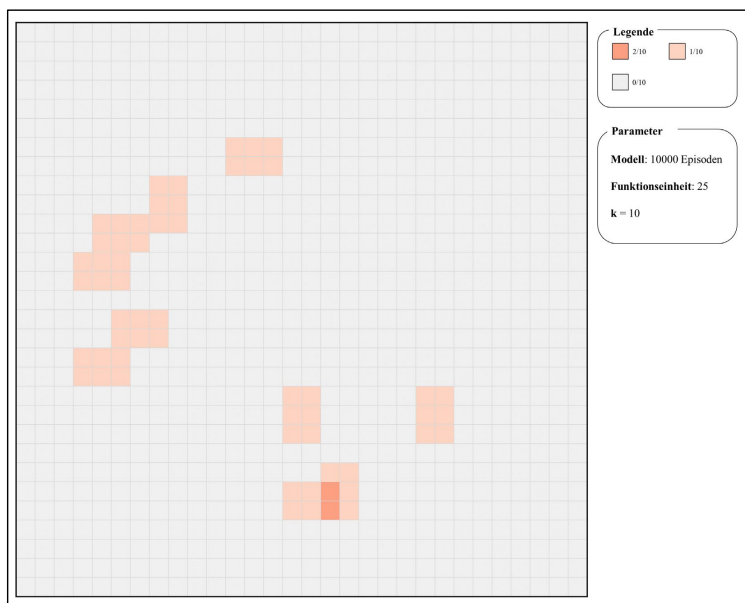


Abbildung 11-7: Ergebnisse der Verteilungsanalyse für Funktionseinheit 25 nach 10000 Episoden

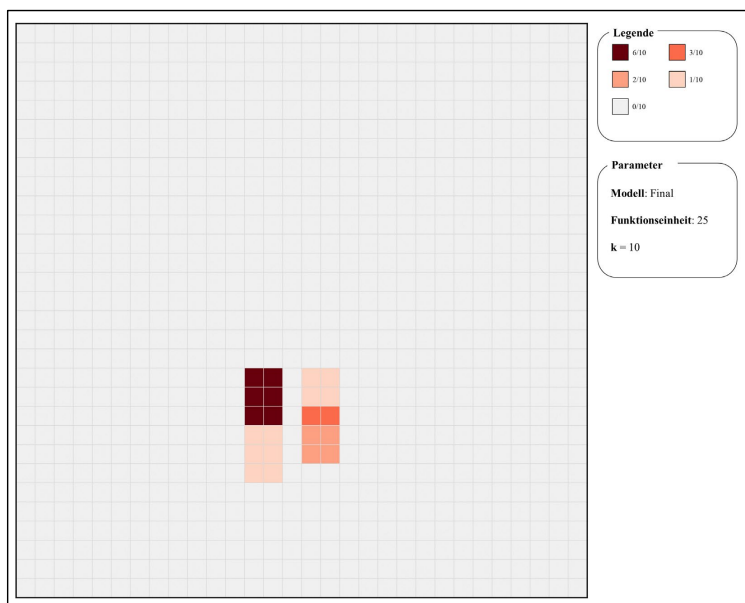


Abbildung 11-8: Ergebnisse der Verteilungsanalyse für Funktionseinheit 25 für das finale Modell

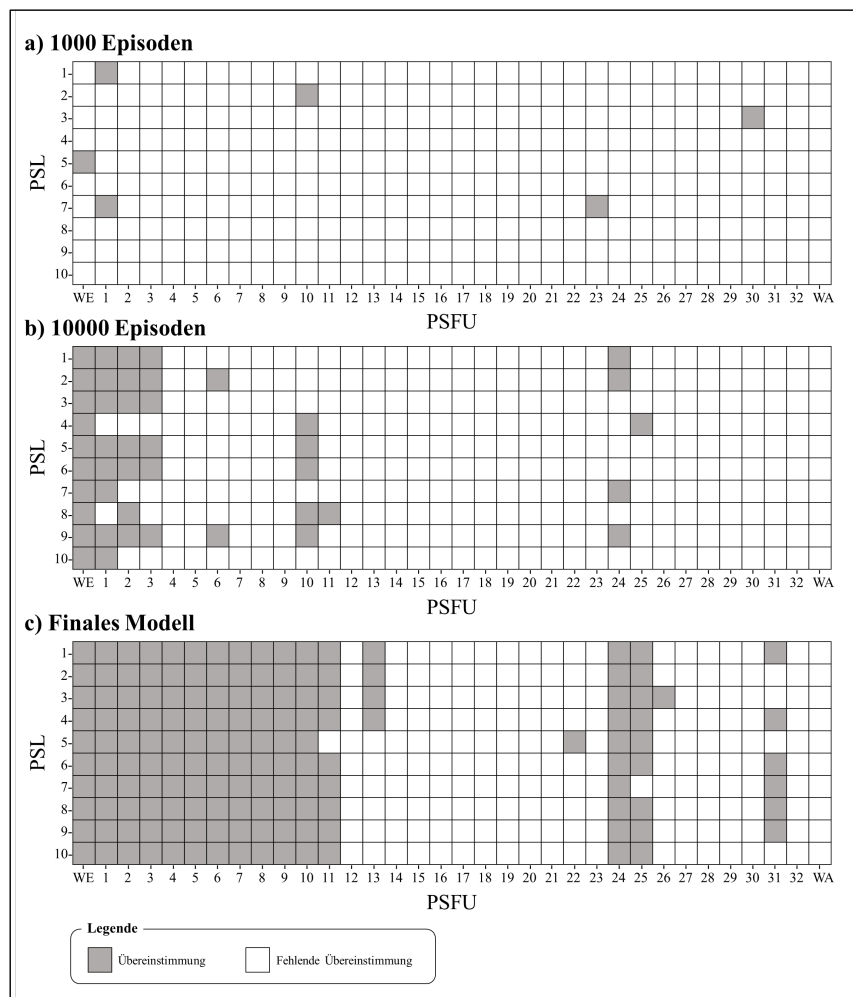


Abbildung 11-9: Übereinstimmungsanalyse nach 1000 (a), 10000 Episoden (b) sowie finales Modell (c) für den Neuplanungsfall

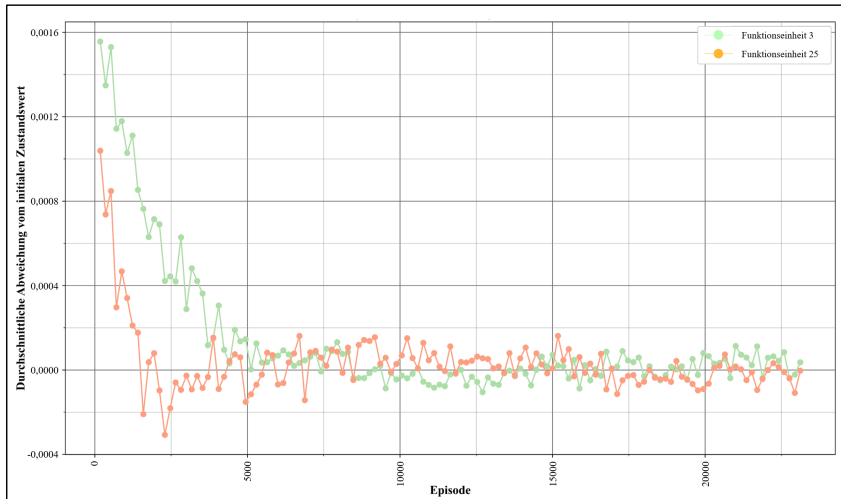


Abbildung 11-10: Ergebnisse der Occlusion-Sensitivitätsanalyse für Funktionseinheit 5 und 25 für den Neuplanungsfall

11.3 Anhang C: Übersicht über die wichtigsten Python-Bibliotheken

Tabelle 11-1: Übersicht über die wichtigsten Python-Bibliotheken

Name	Version	Nutzung und Kurzbeschreibung
TensorFlow	2.7.0	Nutzung: Entwicklung des Agenten
Keras	2.7.0	TensorFlow und Keras werden für die Implementierung von KNN verwendet und bieten grundlegende Bausteine, die die Modellbildung und das Training erleichtern.
SimPy	4.0.1	Nutzung: Basis des EDS-Modells SimPy ist ein Rahmenwerk, das für die Entwicklung von EDS verwendet werden kann. Es stellt grundlegende Funktionen, wie z. B. die <i>Yield</i> -Funktion, zur Verfügung, mit denen EDS beliebiger Art entwickelt werden können. Im Vergleich zu Software wie Tecnomatix Plant Simulation kann SimPy kostenlos genutzt werden, allerdings muss die Simulation komplett eigenständig entwickelt werden, da keine vorentwickelte Materialflusssimulation o. ä. zur Verfügung gestellt wird.
NetworkX	2.5.1	Nutzung: Routenfindung Förderfahrzeuge Mit NetworkX können Netzwerke analysiert, visualisiert und modelliert werden. Dazu werden Ansätze der Graphentheorie verwendet. U. a. gibt es eine Funktion, die den kürzesten Weg zwischen zwei Knoten eines Graphen mit Hilfe des Dijkstra-Algorithmus bestimmt.
Dash	2.12.1	Nutzung: Basis des Softwaredemonstrators Dash ist eine Python-Bibliothek, die es ermöglicht, interaktive Webanwendungen und Dashboards zu erstellen, indem Datenvisualisierungskomponenten mit Backend-Funktionalität kombiniert werden, was eine effektive Integration von Datenanalyse und Benutzerinteraktion ermöglicht.
NumPy	1.23.5	Nutzung: Laden, verarbeiten und transformieren von Daten
Pandas	1.2.3	NumPy kann für effiziente numerische Operationen und lineare Algebra in Python verwendet werden, während Pandas für die Transformation und Analyse von Daten in Tabellenform eingesetzt werden kann.
Matplotlib	3.5.1	Nutzung: Visualisierung von Ergebnissen
Seaborn	0.11.1	Matplotlib und Seaborn bieten die grundlegenden Funktionen, um benutzerfreundliche Grafiken wie z. B. Heatmaps zu erstellen.

11.4 Anhang D: Übersicht über die realisierbaren Ergebnisse durch Transfer Lernen

Das Training eines übertragbaren RL-Ansatzes wird anhand von drei Anwendungsfällen mit steigender Problemkomplexität hinsichtlich der Anzahl der betrachteten Funktionseinheiten, der Layoutgröße, des Optimierungsziels und der vordefinierten Bereiche untersucht. Die Charakteristika der Anwendungsfälle sind in Abbildung 11-11 dargestellt.

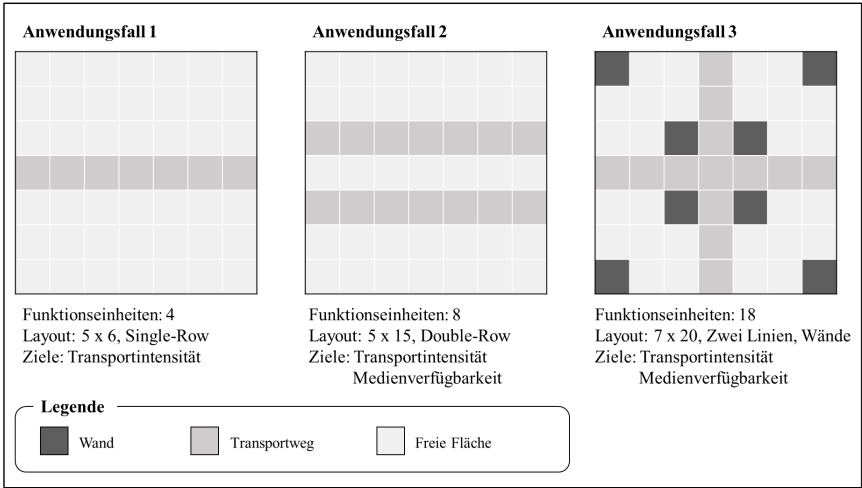


Abbildung 11-11: Anwendungsfälle für die Transferuntersuchungen

Zur Bewertung des Leistungspotentials wird ein vortrainierter RL-Ansatz der 5. Stufe der Übertragbarkeit (siehe Kapitel 9) mit einem neu trainierten RL-Ansatz für jeden Anwendungsfall verglichen. Die Ergebnisse sind in Tabelle 11-2 bis Tabelle 11-4 zusammengefasst.

Tabelle 11-2: Zusammenfassung der besten Lösungen gemessen an der Episodenbelohnung

RL-Ansatz	Anwendungsfall		
	1	2	3
Transfer Lernen	-0,498	-0,396	-1,260
Neu trainiert	-0,498	-0,587	-1,525

Tabelle 11-3: Trainingsepisoden bis zur Erreichung der besten Lösungen

RL-Ansatz	Anwendungsfall		
	1	2	3
Transfer Lernen	109	175632	61493
Neu trainiert	11896	100619	107323

Tabelle 11-4: Trainingsepisoden bis zur Erreichung der besten Lösung des neu trainierten Ansatzes

RL-Ansatz	Anwendungsfall		
	1	2	3
Transfer Lernen	109	9001	19700
Neu trainiert	11896	100619	107323

Lebenslauf

Persönliche Daten:

Name: Matthias Klar

Geburtsort: Mainz

Berufstätigkeit:

10/2016 – 10/2018	Wissenschaftliche Hilfskraft Lehrstuhl für Mikroökonomie und Spieltheorie Lehrstuhl für Business Information Systems & Operations Research TU Kaiserslautern
04/2018 – 10/2018	Praktikant Audi AG
10/2018 – 04/2020	Wissenschaftliche Hilfskraft Lehrstuhl für Werkstoffkunde TU Kaiserslautern
seit 09/2020	Wissenschaftlicher Mitarbeiter Lehrstuhl für Fertigungstechnik und Betriebsorganisation TU Kaiserslautern
seit 10/2023	Oberingenieur Produktionssysteme Lehrstuhl für Fertigungstechnik und Betriebsorganisation TU Kaiserslautern

Produktionstechnische Berichte aus dem FBK

bereits veröffentlicht wurden

- 01/09 Aurich, J.C.: Radio Frequency Identification (RFID) in der Fertigung – Handbuch zur Einführung in neue Anwendungsfelder.
2009, ISBN 978-3-941438-08-8 € 30,-
- 02/09 Stürenburg, B.: Optimierung der Spanbildung und Minimierung des Späneeintrages in das Werkstück für das Bohren von Al-Legierungen.
2009, ISBN 978-3-941438-13-2 € 30,-
- 03/09 Eyrisch, T.: Optimierung der Herstellung von Vollhartmetallwerkzeugen – Strategie zur Vermeidung von Oberflächenschädigungen.
2009, ISBN 978-3-941438-12-5 € 30,-
- 04/09 von Killisch-Horn, G.: Erfahrungsbasierte Implementierung von APS-Systemen in der Einzel- und Kleinserienproduktion.
2009, ISBN 978-3-941438-17-0 € 30,-
- 05/09 Naab, C.: Auslaufmanagement – Konzept zur Abwicklung des Serienauslaufs in der Produktion.
2009, ISBN 978-3-941438-26-2 € 35,-
- 06/09 Herzenstiel, P.: Hochleistungsplanschleifen mit einer definiert gesetzten CBN-Schleifscheibe.
2009, ISBN 978-3-941438-28-6 € 35,-
- 07/09 Aurich, J.C.: Lebenszyklusorientierte Konfiguration von Produkt-Service Systemen – Ein Leitfaden für die Praxis.
2009, ISBN 978-3-941438-27-9 € 30,-
- 01/10 Aurich, J.C.: Organisationsveranstaltung und Kompetenzaufbau bei der Gestaltung und Realisierung investiver Produkt-Service Systeme – Ein Leitfaden für die Praxis.
2010, ISBN 978-3-941438-37-8 € 35,-
- 02/10 Aurich, J.C.: Gestaltung investiver Produkt-Service Systeme – Ein Leitfaden für die Praxis.
2010, ISBN 978-3-941438-38-5 € 35,-
- 03/10 Aurich, J.C.: Realisierung investiver Produkt-Service Systeme – Ein Leitfaden für die Praxis.
2010, ISBN 978-3-941438-39-2 € 35,-
- 04/10 Franke, V.: Einfluss der Werkzeugschneidkante auf die Bohrungsqualität bei der spanenden Bearbeitung langfaserverstärkter Kunststoffe.
2010, ISBN 978-3-941438-53-8 € 35,-

05/10	Sudermann, H.: Gratbildung beim Schleifen – Experimentelle Analyse und Modellierung der Gratbildung sowie Ansätze zur Gratminimierung. 2010, ISBN-978-3-941438-54-5	€ 35,-
06/10	Shahzad, M.A.: Analysis of the Machinability of an Aluminum Matrix Composite Material. 2010, ISBN-978-3-941438-56-9	€ 35,-
07/10	Gómez Kempf, F.A.: Flexibilitätsorientierte Produktionssysteme – Modulare Gestaltung, Einführung und Nutzung. 2010, ISBN-978-3-941438-58-3	€ 35,-
01/11	Engmann, J.: Galvanisch gebundene Mikroschleifstifte – Entwicklung, Herstellung und Einsatz. 2011, ISBN-978-3-941438-59-0	€ 35,-
02/11	Leitz, L.: Beitrag zur Beherrschung der Gratbildung bei Bohrungsverschneidungen. 2011, ISBN-978-3-941438-72-9	€ 35,-
03/11	Siener, M.: Sicherung der Qualitätsfähigkeit von Produktionsprozessen. 2011, ISBN-978-3-941438-79-8	€ 35,-
04/11	Wolf, N.: Modell zur Unterstützung der Analyse von Ausfallursachen bei komplexen mechatronischen Systemen. 2011, ISBN-978-3-941438-89-7	€ 35,-
01/12	Faltin, M.: RFID-Einsatz in fertigungstechnischen Prozessketten – Identifikation und Beurteilung von Einsatzpotentialen. 2012, ISBN-978-3-941438-97-2	€ 35,-
01/13	Lauer, C.: Integriertes Modell zur Materialflusssimulation und zur Visualisierung in der Virtuellen Realität. 2013, ISBN 978-3-943995-19-0	€ 35,-
02/13	Aurich, J.C.: Qualitätsorientierte Instandhaltungsplanung und -steuerung in produktionstechnischen Prozessketten Handbuch für die Praxis. 2013, ISBN 978-3-943995-21-3	€ 35,-
03/13	Kirsch, B.: Wirkmechanismen der Kühlschmierstoff-Zufuhr beim Schleifen – Analyse externer und interner Kühlschmierstoff-Zufuhrmethoden. 2013, ISBN 978-3-943995-25-1	€ 35,-
04/13	Schweitzer, E.J.: Prozessorientiertes Dienstleistungsproduktionssystem. 2013, ISBN 978-3-943995-30-2	€ 35,-

05/13	Malak, R. C.: Methode zur softwarebasierten Planung technischer Änderungen in der Produktion. 2013, ISBN 978-3-943995-34-3	€ 35,-
06/13	Yang, X.: Sound Simulation and Visualization in virtual Manufacturing Systems. 2013, ISBN 978-3-943995-39-8	€ 35,-
07/13	Kranz, J.-N.: Konzept zur Vermeidung von Produkt-Piraterie Risiken – Ein Ansatz für die Ersatzteile der Investitionsgüterindustrie – 2013, ISBN 978-3-943995-46-6	€ 35,-
01/14	Mannweiler, C.: Konfiguration investiver Produkt-Service Systeme. 2014, ISBN 978-3-943995-61-9	€ 35,-
02/14	Aurich J. C. (Hrsg.): Produktivität und Qualität von Serviceprodukten – Ein Leitfaden für die Praxis. 2014, ISBN 978-3-943995-65-7	€ 35,-
03/14	Waltemode S.: Qualitätsbewertung technischer Produkt-Service Systeme. 2014, ISBN 978-3-943995-68-8	€ 35,-
04/14	Bohr C. L.: Vermeidung struktureigener Produktpiraterie bei Ersatzteilen. 2014, ISBN 978-3-943995-69-5	€ 35,-
05/14	Olenburg A.: Tribologisches Prozessverhalten beim Bearbeiten von Feinsteinzeug-Fliesen. 2014, ISBN 978-3-943995-76-3	€ 35,-
01/15	Zimmermann M.: Drehen von Aluminium - Matrix - Verbundwerkstoffen – Analyse thermischer und mechanischer Einflüsse auf die Fertigungs- genauigkeit bei der Trockenbearbeitung. 2015, ISBN 978-3-943995-94-7	€ 35,-
01/16	Adam M.: Konzept zum Umgang mit scheinbar zufälligen Fehlern an technischen Systemen – eine Systematik. 2016, ISBN 978-3-95974-008-1	€ 35,-
02/16	Walk M.: Integriertes Desktopmaschinensystem für die Herstellung und Anwendung ultrakleiner Mikroschleifwerkzeuge. 2016, ISBN 978-3-95974-021-0	€ 35,-
03/16	Carrella M.: Zerspanungsmechanismen beim Mikroschleifen von einkristallinem Silizium. 2016, ISBN 978-3-95974-032-6	€ 35,-
01/17	Reichenbach I. G.: Beitrag zur Beherrschung der Mikrofräsbearbeitung von Polymethylmethacrylat. 2017, ISBN 978-3-95974-042-5	€ 35,-

- 02/17 Fallenstein F.: Kühlkanalaustrittsbedingungen bei VHM-Wendelbohrern
- Analyse der Einflüsse von Kühlkanalaustrittsposition und
KSS-Volumenstrom auf die Temperatur im Schneidkeil.
2017, ISBN 978-3-95974-056-2 € 35,-
- 03/17 Winkes P.: Vorgehensmodell für den systematischen Einsatz
der Virtuellen Realität zur Unterstützung des Montageplanungsprozesses.
2017, ISBN 978-3-95974-057-9 € 35,-
- 04/17 Schneider F.: Analyse der Spanbildung und Oberflächenmorphologie
beim Mikrospanen von Titanwerkstoffen
2017, ISBN 978-3-95974-059-3 € 35,-
- 05/17 Mert G.: Kumulierter Energieaufwand technischer Produkt-Service Systeme
2017, ISBN 978-3-95974-074-6 € 35,-
- 06/17 Weidig C.: Konzept zum prozessgetriebenen Anforderungsmanagement
für die Entwicklung mobiler Fabrikplanungs-Apps
2017, ISBN 978-3-95974-075-3 € 35,-
- 07/17 Cichos D.: Physikalische Modellierung technischer Änderungen
in der Produktion
2017, ISBN 978-3-95974-073-9 € 35,-
- 01/18 Effgen C.: Schneidkantenpräparation mit elastisch gebundenen
Schleifwerkzeugen – Konzeptionierung, Entwicklung und Bewertung
am Beispiel von Vollhartmetallfräswerkzeugen
2018, ISBN 978-3-95974-089-0 € 35,-
- 02/18 Kasakow G.: Entwicklung einer ereignisdiskreten Produktionssteuerung
2018, ISBN 978-3-95974-090-6 € 35,-
- 03/18 Mayer P.: Verformungsinduzierte Martensitbildung beim kryogenen
Drehen von metastabilem austenitischem Stahl
2018, ISBN 978-3-95974-098-2 € 35,-
- 04/18 Herder C.: Konzept zur impliziten Identifikation und Planung
von Technologien für die Produktion
2018, ISBN 978-3-95974-101-9 € 35,-
- 01/19 Müller C.: Kompakte Luftlagerspindeln für die spanende Mikrobearbeitung
auf Desktop-Werkzeugmaschinen
2019, ISBN 978-3-95974-105-7 € 35,-
- 01/20 Meissner H. M.: Integrierte Arbeitsplanung und Produktionssteuerung
cyber-physischer Produktionssysteme
2020, ISBN 978-3-95974-125-5 € 35,-

- 02/20 Sinnwell C.: Methode zur Produktionssystemkonzipierung auf Basis früher Produktinformationen – Ein Beitrag zur Integration von Produktionssystemplanung und Produktentwicklung unter Einsatz des MBSE
2020, ISBN 978-3-95974-127-9 € 35,-
- 03/20 Bohley M.: Prozess-Maschine-Interaktion beim Mikrofräsen von cp-Titan
2020, ISBN 978-3-95974-130-9 € 35,-
- 01/21 Hotz H.: Kryogenes Drehen metastabiler austenitischer Stähle – Analyse der Wirkzusammenhänge zwischen Einstellgrößen, Prozessgrößen und Oberflächenmorphologie
2021, ISBN 978-3-95974-146-0
- 02/21 Arrabiyeh P. A.: Electroless Plated Micro Pencil Grinding Tools: Conception, Manufacturing, and Application
2021, ISBN 978-3-95974-158-3
- 03/21 Yi L.: Eco-Design for Additive Manufacturing Using Energy Performance Quantification and Assessment
2021, ISBN 978-3-95974-157-6
- 01/22 Kölsch P.: Agile Planung von Produkt-Service Systemen durch den Einsatz von Design Thinking
2022, ISBN 978-3-95974-170-5
- 02/22 Klauer K.: Mikrofräsen flächenhafter Kalibriernormale auf Basis realer Bauteiloberflächen
2022, ISBN 978-3-95974-177-4
- 03/22 Greco S.: Additiv-Subtraktive Prozesskette in der Mikrobearbeitung – Analyse der Einflüsse auf die Prozess- und Prozessergebnisgrößen beim selektiven Laserschmelzen und Mikrofräsen von Edelstahl 1.4404
2022, ISBN 978-3-95974-182-8
- 04/22 Siedler C.: Entwicklung eines Transformationskonzepts zur Digitalisierung von Produktionssystemen in KMU
2022, ISBN 978-3-95974-185-9
- 05/22 Kirsch B.: Spezialanwendungen von Schleifscheiben – Technologische Möglichkeiten und Anforderungen sowie deren Nachhaltigkeit
2022, ISBN 978-3-95974-191-0
- 01/23 Basten S.: Sub-zero Kühlschmierstoffe: Wirkmechanismen und Einsatzverhalten am Beispiel der Drehbearbeitung
2023, ISBN 978-3-95974-202-3

- 02/23 Lange A.: Interaktion zwischen Prozess, Werkzeug, Spindel und Maschine bei der spanenden Mikrobearbeitung. Simulationsgestützte Untersuchung des Mikrofräsens und Mikroschleifens
2023, ISBN 978-3-95974-205-4
- 03/23 Glatt M. F.: Untersuchung des Einsatzes eines Digitalen Zwillings zur Physiksimulation von Materialflüssen in Cyber-Physischen Produktionssystemen
2023, ISBN 978-3-95974-207-8
- 01/24 Weber D.: Simulationsgestützte Vorhersage und Minimierung des eigenspannungsbedingten Verzugs beim Fräsen dünnwandiger Aluminium-Strukturbauteile
2024, ISBN 978-3-95974-227-6
- 02/24 Klar M.: Automatisierte und simulationsbasierte Fabriklayoutplanung mittels Reinforcement Learning
2024, ISBN 978-3-95974-226-9

ISSN 0937-9061
ISBN 978-3-95974-226-9