

KI in der Kommissionierung

Optimierte Auftragszuweisung mittels Reinforcement Learning

Dominik Kuhn,
Jannis Pfister,
Matthias Jenner,
Max Eichenwald und
Rainer Müller*

Logistikprozesse, wie z.B. die manuelle Kommissionierung, fordern Unternehmen gerade in Hochlohnländern heraus, diese möglichst effizient, aber vor allem nachhaltig zu gestalten. Unternehmen müssen eine ausgewogene Balance zwischen der Optimierung der reinen Kommissionierzeit und der gleichmäßigen Auslastung der jeweiligen Lagermitarbeitenden finden. Nur so können unternehmerische Ziele langfristig mit der Mitarbeitendenzufriedenheit vereinbart werden. Dieser Beitrag stellt einen praxisnahen Ansatz zur Optimierung dieses Logistikprozesses vor, der auf den Potenzialen der Künstlichen Intelligenz (KI), genauer dem Reinforcement Learning (RL), basiert.

Einleitung und Motivation

Im globalen Wettbewerbsumfeld müssen Hochlohnländer ihre Abläufe verstärkt rationalisieren, um konkurrenzfähig zu bleiben. Dieser Druck zwingt Unternehmen zur gezielten Optimierung von Prozessen, u.a. der manuellen Kommissionierung. Die manuelle Kommissionierung wird weiterhin besonders stark in volatilen Geschäftsbereichen und bei hoher Variantenvielfalt genutzt. Mit bis zu 55 Prozent der Lagerlogistikkosten, die auf die Kommissionierung entfallen, existiert hier ein erhebliches Optimierungspotenzial [1]. Trotz dieser Möglichkeiten sehen sich viele Unternehmen weiterhin mit ineffizienten Prozessen konfrontiert, da Aufträge häufig manuell oder durch veraltete Algorithmen an die Arbeitsplätze

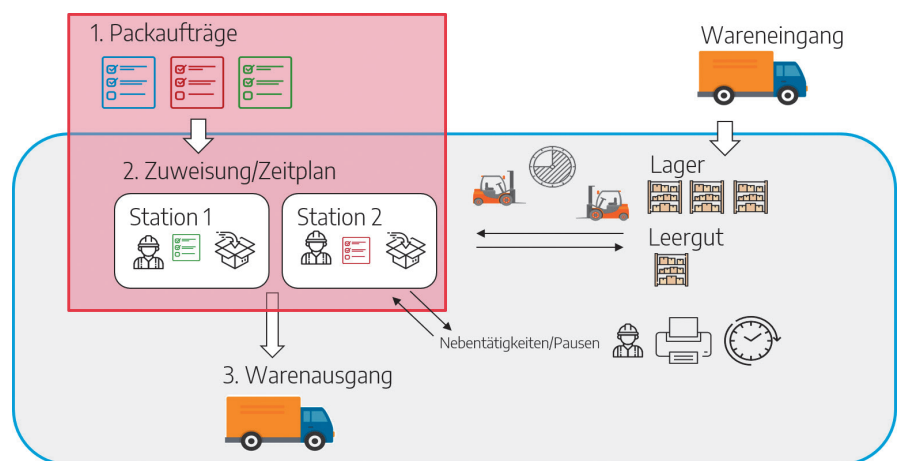


Bild 1. Ist-Situation des Anwendungsfalls

zugewiesen werden. Vor diesem Hintergrund bieten die Technologien der KI viel-

fältige Möglichkeiten, innovative Wege zu beschreiten und neue Ansätze für die Auftragszuweisung zu entwickeln [2].

* Korrespondenzautor

Dominik Kuhn, M.Sc.; ZeMA – Zentrum für Mechatronik und Automatisierungstechnik gGmbH; Eschberger Weg 46, 66121 Saarbrücken; Tel.: +49 (0) 681 85787-533, E-Mail: d.kuhn@zema.de

Weitere Autoren

*Jannis Pfister, M.Sc.; ZeMA – Zentrum für Mechatronik und Automatisierungstechnik gGmbH
Matthias Jenner, B.Sc.; ZeMA – Zentrum für Mechatronik und Automatisierungstechnik gGmbH
Max Eichenwald, M.Sc.; ZeMA – Zentrum für Mechatronik und Automatisierungstechnik gGmbH
Prof. Dr.-Ing. Rainer Müller; Lehrstuhl für Montagesysteme an der Universität des Saarlandes*

Hinweis

Bei diesem Beitrag handelt es sich um einen von den Advisory-Board-Mitgliedern des ZWF-Sonderheftes wissenschaftlich begutachteten Fachaufsatz (Peer-Review).

Ausgangssituation

Dieser Beitrag präsentiert die Umsetzung eines KI-gestützten Zuordnungssystems für Schichtleiter zur effizienten Auftragszuweisung an verfügbare Arbeitsplätze. Der Ansatz beruht auf einem realen Szenario, welches schematisch in Bild 1 zu sehen ist, und berücksichtigt automatisierte sowie manuelle Zuordnungselemente. Die Hauptoptimierungsziele sind die Minimierung der Bearbeitungszeit,

die gleichmäßige Arbeitsverteilung und die Einhaltung der Termintreue.

Der klassische Kommissionierprozess im Dienstleistungssektor der Logistik beginnt in der Regel mit der Erfassung und Einlagerung kundenspezifischer Waren und Leergüter. Die Einlagerung erfolgt dabei meist unabhängig vom konkreten späteren Kommissionierauftrag. Parallel zum Wareneingang treffen kontinuierlich neue Aufträge ein, die im Auftragspool gesammelt werden. Der Schichtleiter ordnet die Aufträge im Auftragspool den entsprechenden Arbeitsplätzen der Kommissionierung zu, wodurch ein sogenannter Zuweisungsplan entsteht. Kommt es während der Abarbeitung dieses Zuweisungsplans zu neuen Aufträgen durch den Kunden in Form von „Eilaufträgen“, muss der bestehende Zuweisungsplan regelmäßig überarbeitet und optimiert werden. Der eigentliche Kommissionierprozess beginnt, sobald das Material eines offenen Auftrags verfügbar ist. Dies umfasst die Umlagerung von Kleinladungsträgern und die Verpackung der Zielpalette mit kundenspezifischen Etiketten. Die gesamte Kommissionierzeit wird üblicherweise durch das Scannen der Auftragspapiere erfasst und durch Start- und Endzeitstempel dokumentiert.

Die Entwicklung eines digitalen und KI-gestützten Zuordnungssystems hat das Potenzial einer schnelleren Sortierung, Verteilung und Bearbeitung der Aufträge, wodurch Schichtleiter zeitlich entlastet werden, aber auch zur Erkennung von besonders zeitkritischen Terminvorgaben, um diese zu priorisieren.

Stand der Wissenschaft und Technik

Das interdisziplinäre Fachgebiet Operations Research (OR) zielt darauf ab, Entscheidungsprozesse systematisch zu unterstützen. Dabei werden häufig Methoden aus den Bereichen Mathematik, Wirtschaftswissenschaften und Informatik eingesetzt. Obwohl die methodischen Grundlagen überwiegend in der Mathematik und Informatik verankert sind, findet OR Anwendung in zahlreichen Bereichen, einschließlich der Ingenieurwissenschaften. Besonders bei der Auftragszuweisung gewinnen die Prinzipien des OR eine besondere Bedeutung, da eine effiziente

Verteilung von Aufträgen einen wesentlichen Einfluss auf den betrieblichen Erfolg ausübt [3].

Als wichtigstes Teilgebiet des OR ist die mathematische Optimierung, insbesondere die gemischt-ganzzahlige Optimierung (siehe Abschnitt Mixed Integer Programmierung), zu benennen. Darüber hinaus stehen im OR, je nach Aufgabenstellung und Modellbildung, eine breite Palette an weiteren Methoden zur Verfügung, um betriebliche Entscheidungsprozesse zu unterstützen. Zu diesen Verfahren zählen unter anderem:

- Simulationsmethoden,
- Entscheidungsbaum-Analysen,
- lineare, quadratische, nichtlineare und weitere mathematische Optimierungsverfahren,
- spieltheoretische Modelle und Methoden
- Modelle der Warteschlangentheorie sowie
- heuristische Verfahren.

Jedes dieser Verfahren bietet spezifische Ansätze zur Lösung komplexer betrieblicher Fragestellungen und trägt zur Verbesserung der Entscheidungsfindung bei.

Diesen Verfahren und Methoden des OR stehen im Hinblick zu dem aktuellen Stand der Wissenschaft und Technik der breitgefächerte Bereich der künstlichen Intelligenz (KI) gegenüber. In den folgenden beiden Abschnitten werden exemplarisch ein Verfahren der gemischten-ganzzahligen Optimierung sowie ein prominenter Ansatz des maschinellen Lernens (siehe Abschnitt Reinforcement Learning) präsentiert und erläutert. Dabei wird die Balance zwischen schnellen, effizienten Lösungen und den damit verbundenen Herausforderungen bezüglich Flexibilität und Anpassungsfähigkeit näher beleuchtet.

Mixed Integer Programming

Mixed Integer Programming (MIP), auch bekannt als gemischt-ganzzahliges lineares Programmieren, ist eine leistungsstarke Optimierungsmethode. Ziel der Methode ist die Maximierung oder Minimierung einer linearen Zielfunktion, unter Einhaltung von linearen Nebenbedingungen. Hierzu wird das eigentliche Problem zu Beginn mathematisch formuliert, wobei sogenannte Entscheidungsvariablen, Zielfunktion und Nebenbedin-

gungen definiert werden. Die Zielfunktion ist linear und hängt von den oben genannten Entscheidungsvariablen ab.

Nebenbedingungen, die die zulässigen Werte für die Variablen einschränken, werden ebenfalls formuliert und in das Modell integriert. Sie können Ressourcenbeschränkungen, Kapazitätsgrenzen oder logische Verknüpfungen darstellen und sind typischerweise ebenfalls linear. MIP wird in der Logistik häufig zur Lösung verschiedener Optimierungsprobleme eingesetzt, so beispielsweise bei der Optimierung der Routenplanung, der Lagerhaltung oder der Standortplanung [4].

Der eigentliche Ablauf eines MIP-Verfahrens zur Lösung des Modells erfolgt in mehreren Schritten:

■ Relaxation

Zunächst werden die ganzzahligen Variablen als kontinuierlich behandelt, wodurch das Problem leichter zu lösen ist und erste Näherungslösungen bietet.

■ Branch-and-Bound-Verfahren

Der Algorithmus teilt den Lösungsraum in Teilprobleme (Branching), berechnet für jedes Teilproblem die beste Lösung und schließt nicht zulässige oder suboptimale Teilräume aus (Bounding). Der Lösungsraum wird dadurch systematisch und effizient durchsucht [5].

■ Branch-and-Cut-Verfahren

Um den Prozess weiter zu beschleunigen, wird oft das Branch-and-Cut-Verfahren verwendet. Hierbei werden zusätzlich Schnittebenen (Cuts) eingeführt, die den Lösungsraum ohne Verlust der optimalen Lösung einschränken und so den Rechenaufwand reduzieren [6].

■ Iterationen und Konvergenz

Am Ende iteriert ein Algorithmus über alle Teilprobleme bis alle Ganzzahligkeitsanforderungen erfüllt sind und die Zielfunktion nicht weiter verbessert werden kann [7].

Ein wesentlicher Punkt bei MIP ist, dass es in der strukturierten Optimierung mit festen Randbedingungen sehr gute Ergebnisse liefert. Allerdings kann MIP keine Lösung bieten, wenn bestimmte Bedingungen nicht erfüllt sind, was seine Abhängigkeit von der strikten Einhaltung der Randbedingungen zeigt. In sol-

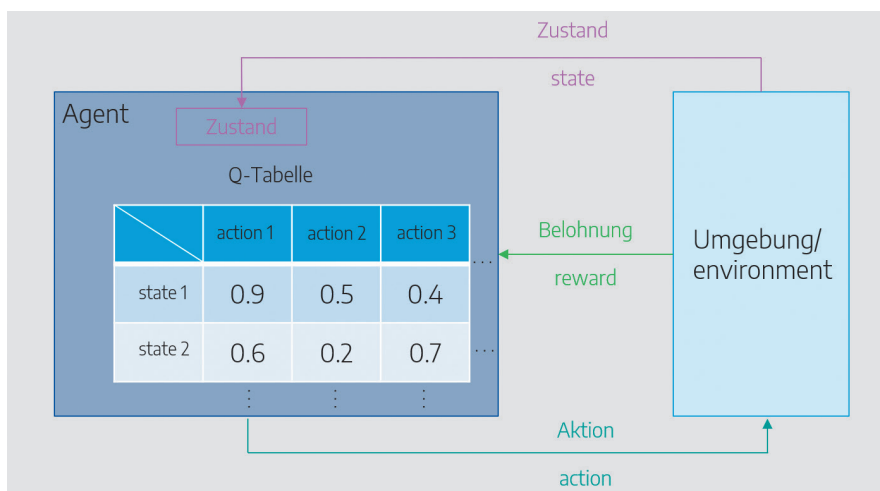


Bild 2. Prinzip des Reinforcement Learnings

chen Fällen sind ggf. alternative Ansätze notwendig, wie etwa RL, das flexibel auf veränderliche Anforderungen reagiert und selbst unter unvollständigen Bedingungen eine Lösung bieten kann.

Reinforcement Learning

RL ist ein Ansatz des maschinellen Lernens, bei dem ein Agent mit seiner Umwelt interagiert, um bestimmte Ziele zu erreichen. Durch Beobachtungen, eigenständige Entscheidungen und Aktionen verändert der Agent den Zustand seiner Umgebung und erhält Rückmeldungen in Form von Belohnungen. Diese unterstützen die Entwicklung einer geeigneten Strategie zur Zielerreichung. Der Lernprozess umfasst Exploration für neue Erfahrungen und Exploitation für bereits erlerntes Wissen. Das Ziel besteht darin, die kumulative Belohnung zu maximieren. Wie in Bild 2 zu erkennen, handelt es sich um ein iteratives Verfahren. RL findet in vielen Bereichen Anwendung, wobei es sich besonders in komplexen und dynamischen Umgebungen, wie z. B. der Auftragszuweisung, durch die Entwicklung effizienter Entscheidungsstrategien auszeichnet [8, 9].

Q-Learning-Algorithmus

Q-Learning beschreibt einen speziellen Lernalgorithmus des RL, der verwendet wird, um optimale Aktionswerte in einem Umfeld zu berechnen. Er basiert auf der Vorhersage des Werts aus Zustands-Aktions-Paaren. Der Algorithmus

lernt schrittweise die Vorteile bestimmter Aktionen in bestimmten Zuständen auszuführen, indem er die aktuellen Belohnungen und zukünftigen Werte berücksichtigt. Dies wird durch die Q-Funktion dargestellt, die mit einer Lernrate aktualisiert wird. Die Q-Tabelle (Tabelle 1), welche Zustände und Aktionen miteinander verknüpft und entsprechende Werte der Q-Funktion speichert, wird während des Lernprozesses kontinuierlich aktualisiert [10].

Entwicklung eines Reinforcement-Learning-Ansatzes zur optimierten Auftragszuweisung

Im Folgenden wird ein Ansatz zur Anwendung von RL als Lösungsverfahren für die Auftragszuweisung vorgestellt.

Der nachfolgende Ansatz steht dabei im Kontrast zu den klassischen Verfahren. Diese bieten zwar schnelle und gute Lösungen, jedoch sind sie in der Regel aufgrund ihrer Limitierungen an Flexibilität und Anpassungsfähigkeit eingeschränkt. Durch die Entwicklung eines RL-Ansatzes wird das Ziel verfolgt, Limitierungen, z. B. des MIP, zu überwinden und das System agiler zu gestalten. Im konkreten Anwendungsfall liegt der Fokus auf:

- Optimierung bzw. Minimierung der Packzeiten,
- gleichverteilte Packzeiten bzw. gleichmäßige Auslastung an den Packplätzen sowie
- Einhaltung der Termintreue.

Der RL-Ansatz soll dem System ermöglichen, aus Erfahrungen zu lernen und seine Entscheidungen kontinuierlich zu verbessern. Dadurch kann es sich an sich ändernde Anforderungen anpassen und optimale Auftragszuweisungen vornehmen. Gleichzeitig sollte sichergestellt sein, dass die Mitarbeitenden in den Entscheidungsprozess eingebunden werden können. Das bedeutet, dass ihre individuellen Erfahrungen und Rückmeldungen mit in die Entscheidungsfindung einfließen.

Implementierung des Ansatzes

Dieser Abschnitt beschäftigt sich neben dem Grobkonzept des ganzen Systems mit der Strukturierung und softwaretechnischen Implementierung des RL-Ansatzes unter Verwendung objektorientierter Programmierungskonzepte. Im Kontext des RL-Ansatzes können so bspw. Klassen verwendet werden, um die verschiedenen Komponenten des Systems zu mo-

Q-Learning-Algorithmus

1	Initialisiere Q-Tabelle mit zufälligen Werten
2	for jede Lernepisode do
3	Setze den aktuellen Zustand s auf den Startzustand
4	while das Lernziel nicht erreicht ist do
5	Wähle Aktion a basierend auf einer Strategie (z. B. Epsilon-greedy)
6	Führe Aktion a aus und erhalte Belohnung r und neuen Zustand 's'
7	Aktualisiere den Q-Wert mit der Q-Lerngleichung: $Q(s,a) \leftarrow Q(s,a) + \text{Lernrate} \times (r + \text{Discount} \times \max_{a'}(Q(s',a')) - Q(s,a))$
8	Setze den aktuellen Zustand s auf s'
9	end while
10	end for

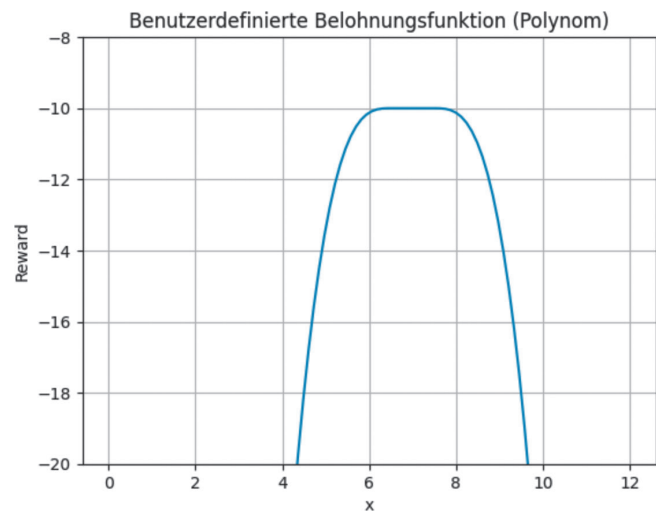
Tabelle 1. Pseudo-Implementierung von Q-Learning

modellieren. Da bei RL-Algorithmen ein Agent (Bild 2) mit einer Umgebung (Environment) interagiert und Entscheidungen trifft, um ein bestimmtes Ziel zu erreichen, spielt dabei die Modellbildung der Umgebung eine entscheidende Rolle. Sie ermöglicht dem Agenten, Vorhersagen über seine Umgebung zu treffen und zukünftige Aktionen abzuschätzen. Die Repräsentation des aktuellen Zustands ist hierbei sehr wichtig, da sie die relevanten Informationen enthält, die der Agent für seine Entscheidungsfindung benötigt. Darüber hinaus enthält sie Informationen wie mögliche Aktionen sowie ein sorgfältig gestaltetes Belohnungsmodell. Die eigentliche Agentenklasse repräsentiert den handelnden Agenten und enthält Funktionen zur Aktionsauswahl, Zustandsaktualisierung und Gewichts Anpassung, inklusive RL-Algorithmen, wozu beispielsweise das Q-Learning für das Lernen und die Optimierung zählen. Man kann sich die Agentenklasse beziehungsweise den Agenten wie eine Spielfigur vorstellen, die sich in einer virtuellen Welt bewegt und verschiedene Aktionen ausführt. Der Agent „erkennt“ den aktuellen Zustand seiner Umgebung und entscheidet darauf basierend, welche Aktion er als nächstes ausführen sollte, ähnlich wie ein Charakter in einem Video- oder Brettspiel. Im Folgenden werden anwendungsspezifische Implementierungen der beiden oben genannten Klassen näher betrachtet sowie die Besonderheiten von gleichverteilten Zeiten als Lernansatz diskutiert.

Umgebungs-klasse

Die Umgebungs-klasse modelliert die Agenteninteraktion, verwaltet – wie oben beschrieben – den aktuellen Zustand, ermöglicht Aktionen und gibt Rückmeldungen in Form von Belohnungen und Zustandsübergängen. Membervariablen implementieren in der Umgebungs-klasse eine Auftragsliste, die Zuweisungstabelle sowie die möglichen Aktionen (Zuweisung eines Auftrags zu einem Kommissionierarbeitsplatz). Ungültige Aktionen werden nach jedem Schritt entfernt, um somit eine doppelte Auswahl von Aufträgen zu verhindern und das Lernverhalten zu beschleunigen. Die Klasse enthält Methoden wie reset für den Anfangszustand und step für Aktionen mit Informationen über den nächsten Zustand und die Belohnung.

Bild 3. Belohnungsfunktion für gleichmäßige Auslastung



Agentenklasse

Die Agentenklasse repräsentiert einen künstlichen Agenten, der durch das Q-Learning-Verfahren in der Umgebung trainiert wird. Entscheidend sind dabei die select-action- und train-Methoden, welche die Exploration und Exploitation des Agenten steuern. Die select-action-Methode nutzt eine angepasste epsilon-greedy-Strategie, um eine effiziente Balance zwischen Erkundung (Exploration) und Ausbeutung (Exploitation) für optimale Entscheidungen zu gewährleisten. Hierbei wird nur der maximale Q-Wert berücksichtigt, wenn der Agent erkundet, was seine Effizienz und Effektivität im Lernprozess verbessert. Durch dieses Zusammenspiel von Erkundung und Ausbeutung kann der Agent flexibel auf verschiedene Situationen reagieren und unter Anwendung des Q-Learning-Verfahrens optimale Entscheidungen treffen.

Mithilfe von Q-Learning lernt das System eigenständig und verbessert kontinuierlich seine Entscheidungen, was zu einer höheren Effizienz bei der Auftragszuweisung und einer optimierten Packzeit führt. Durch die Wahl geeigneter Parameter und eine strukturierte Belohnungsfunktion reagiert das System flexibel auf unterschiedliche Szenarien und passt die Arbeitsabläufe entsprechend an.

Gleichverteilte Zeiten als Lernansatz

Effiziente Arbeitsabläufe und eine gleichmäßige Auslastung der Ressourcen sind entscheidend für die Maximierung von Produktivität und Leistungsfähigkeit in der Kommissionierung. Die Einbeziehung dieser Aspekte in die Auftragszuweisung des RL-Systems ermöglicht die gleichmäßige Verteilung der Zeiten und optimale Ressourcennutzung. Die Implementierung dieses Lernziels erfordert die Anpassung der Belohnungsfunktion (R_x) des Systems. Um ein solches Belohnungssystem zu entwickeln, wurde eine Arbeitszeit x von 8 Zeitstunden zugrunde gelegt, was einer Nettoarbeitszeit von 6,5 bis 7,5 Stunden entspricht. Grafisch zeigt sich die Funktion (siehe Formel 1) dann wie in Bild 3.

Innerhalb des Bereiches der Nettoarbeitszeit (6,5 bis 7,5 h) bleibt die Belohnung konstant bei -10, um stabile Lernbedingungen zu schaffen und eine übermäßige oder unzureichende Auslastung gleichwertig zu vermeiden. Der gewählte konstante Wert unterstützt das RL-System dabei, eine konsistente Auslastung anzustreben, ohne extreme Abweichungen zu bevorzugen. Außerhalb des definierten Arbeitsbereichs sinkt die Belohnung polynomial, um größere Abweichungen entsprechend zu bestrafen. Obwohl der Exponent einen Einfluss auf die Bestrafung

$$\text{Formel 1. } R_x = \begin{cases} -10 & \text{für } 6.5 \leq x \leq 7.5 \\ -|x - 6.5|^n - 10 & \text{für } x < 6.5 \text{ mit } n \in \mathbb{N} \\ -|x - 7.5|^n - 10 & \text{für } x > 7.5 \text{ mit } n \in \mathbb{N} \end{cases}$$

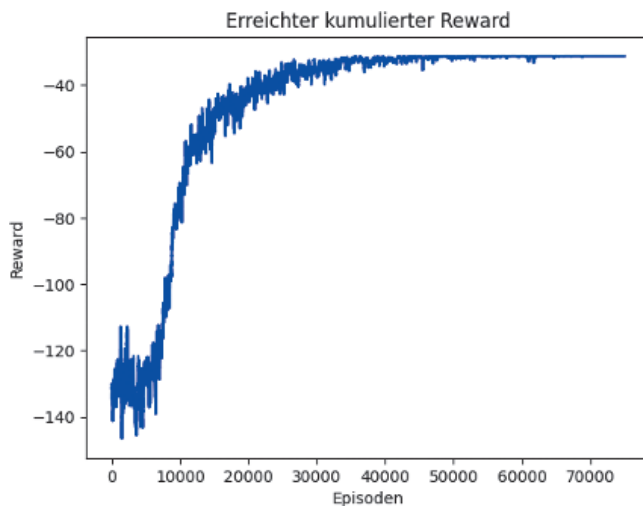


Bild 4. Kumulatives Lernverhalten des Systems

von Abweichungen hat, zeigten Tests, dass dessen Erhöhung nur geringe Effekte auf die Verteilung der Arbeitszeiten hatte. Diese Ergebnisse verdeutlichen, dass eine konstante Belohnung innerhalb des Arbeitsbereichs wichtiger für das Lernverhalten des Systems ist als die Stärke der Bestrafung außerhalb dieses Bereichs.

Dieser Ansatz sorgt für eine ausgewogene Nutzung der Ressourcen, da das System lernt, die Arbeitszeiten optimal zu verteilen. So entsteht ein dynamisches Auftragszuweisungssystem, das flexibel auf veränderte Anforderungen reagieren und gleichzeitig eine konsistente Arbeitsbelastung aufrechterhalten kann.

Zusammenfassung und Ausblick

RL präsentiert sich als entscheidende Komponente zur Optimierung von Logistikprozessen. Der vorgestellte RL-Ansatz zur Optimierung hebt sich im Vergleich zu herkömmlichen Methoden wie dem MIP hervor. Während MIP sehr schnelle Lösungen bietet, stößt es auf Limitationen hinsichtlich Flexibilität und Anpassungsfähigkeit. Im Gegensatz dazu ermöglicht RL eine flexible Gestaltung des Systems. Die Kombination von objektorientierter Programmierung sowie zielgerichteter Belohnung legt einen längerfristigen Grundstein für agile und innovative Prozesse in der Logistiko-optimierung. Durch die Implementierung zeigt das System die Fähigkeit zur adaptiven Anpassung an wandelnde Prioritäten und

dynamische Anforderungen. Dies wird auch durch das stabile Lernverhalten des Systems verdeutlicht (Bild 4).

Für Nutzende des Systems liegt der Mehrwert der RL-Variante in ihrer Fähigkeit, Arbeitsabläufe zu vereinfachen und Prozesse flexibel an wechselnde Bedingungen anzupassen. Die kontinuierliche Verbesserung und Anpassungsfähigkeit des Systems entlasten die Mitarbeitenden, da das System selbstständig lernt und optimale Entscheidungen trifft, wodurch Fehlbelastungen und Überlastungen minimiert werden. Die Mitarbeiter können sich stärker auf wertschöpfende Tätigkeiten konzentrieren und profitieren von einer besseren Planbarkeit und Zuverlässigkeit des Systems. Gleichzeitig wird die Einbindung ihrer Rückmeldungen ermöglicht, was den Entscheidungsprozess verfeinert und die Akzeptanz der Technologie erhöht.

Darüber hinaus bietet in Zukunft die Anwendung einer Pareto-Optimierung die Möglichkeit, unterschiedliche Belohnungsziele geschickt zu kombinieren und somit eine Optimierung in Bezug auf mehrere Kriterien zu erreichen. Eine einfache Möglichkeit, dies zu realisieren, ist es, eine gewichtete Summe mehrerer Belohnungen zu bilden. Für zwei Belohnungen $reward_1$ und $reward_2$ würde dies in einer Gesamtbelohnung $reward_{total} = a \cdot reward_1 + (1-a) \cdot reward_2$ resultieren, wobei $a \in [0;1]$. Dies könnte ebenfalls analog auf mehrere Ziele ausgeweitet werden. Für diese Addition ist es wichtig, darauf zu achten, dass die einzel-

nen Belohnungen sich in einem ähnlichen Werteintervall bewegen, da sonst einzelne Ziele automatisch ausschlaggebender sind als andere und die Bewertung weiterer Kriterien somit hinfällig wird.

Die Integration von Techniken wie Heuristiken, Multi-Agenten-Systemen und Deep Reinforcement Learning verspricht zusätzlich eine Steigerung der Leistungsfähigkeit des Systems und performantere Ergebnisse in dynamischen Umgebungen.

Literatur

1. Franzke, T.: Der Mensch als Faktor in der manuellen Kommissionierung. Springer, Wiesbaden 2018
DOI:10.1007/978-3-658-20469-3
2. Straßer, T.; Axmann, B.: Analyse und Bewertung von KI-Anwendungen in der Logistik. Logistics Journal 8 (2021)
DOI:10.2195/lj_NotRev_strasser_de_202108_01
3. Hompel, M. ten; Sadowsky, V.; Beck, M.: Optimierung von Kommissioniersystemen. In: Hompel, M. ten; Sadowsky, V.; Beck, M. (Hrsg.): Kommissionierung. Springer, Berlin, Heidelberg 2011, S. 91–125
DOI: DOI:10.1007/978-3-540-29940-0_6
4. Koop, A.; Moock, H.: Lineare Optimierung – eine anwendungsorientierte Einführung in Operations Research. Springer Spektrum, Berlin, Heidelberg 2018
DOI:10.1007/978-3-662-56141-6
5. Floudas, C. A.: Nonlinear and Mixed-Integer Optimization. Fundamentals and Applications. Oxford University Press, New York 1995
DOI:10.1093/oso/9780195100563.003.0011
6. Achterberg, T.; Koch, T.; Martin, A.: Branching Rules Revisited. Operations Research Letters 33 (2005) 1, S. 42–54
DOI: DOI:10.1016/j.orl.2004.04.002
7. Wolse, L. A.: Integer Programming. Wiley & Sons, New York et al. 1998
8. Sutton, R. S.; Barto, A. G.: Reinforcement Learning: An Introduction. MIT Press. 2018
9. Zai, A.; Brown, B.: Einstieg in Deep Reinforcement Learning. Carl Hanser Verlag, München 2020
DOI:10.3139/9783446466081.fm
10. Lorenz, U.: Reinforcement Learning. Springer Vieweg, Berlin 2020
DOI:10.1007/978-3-662-61651-2

Die Autoren dieses Beitrags

Prof. Dr.-Ing. Rainer Müller, geb. 1963, studierte Maschinenbau in Paderborn und Fahrzeugtechnik an der RWTH Aachen. 1996 promovierte er im Bereich der Robotik und Automatisierung. Nach verschiedenen Führungspositionen bei der Schenk AG und Dürr AG nahm er eine Pro-

fessur an der RWTH Aachen an und war von 2012 bis 2020 Wissenschaftlicher Geschäftsführer der ZeMA gGmbH in Saarbrücken. Er übernimmt als Professor Aufgaben in der Lehre an der Universität des Saarlandes am Lehrstuhl für Montagesysteme M:sys.

Dominik Kuhn, M.Sc., geb. 1990, studierte Mechatronik mit dem Schwerpunkt Sensortechnik an der Hochschule für Technik und Wirtschaft des Saarlandes, seit 2019 als Wissenschaftlicher Mitarbeiter und Doktorand am ZeMA – Zentrum für Mechatronik und Automatisierungstechnik – in Saarbrücken im Bereich Montageverfahren und -automatisierung tätig.

Jannis Pfister, M.Sc., geb. 1996, studierte Elektrotechnik mit dem Schwerpunkt Automatisierungstechnik an der Hochschule für Technik und Wirtschaft des Saarlandes, seit 2023 als Wissenschaftlicher Mitarbeiter am ZeMA – Zentrum für Mechatronik und Automatisierungstechnik – in Saarbrücken im Bereich Montageverfahren und -automatisierung tätig.

Matthias Jenner, B.Sc., geb. 1990, studierte mechatronik mit dem Schwerpunkt Sensortechnik an der Hochschule für Technik und Wirtschaft des Saarlandes, seit 2022 als Wissenschaftlicher Mitarbeiter am ZeMA – Zentrum für Mechatronik und Automatisierungstechnik – in Saarbrücken im Bereich Montageverfahren und -automatisierung tätig.

Max Eichenwald, M. Sc., geb. 1994, studierte Mechatronik an der Universität des Saarlandes. Er ist seit 2019 als Wissenschaftlicher Mitarbeiter und Doktorand am ZeMA – Zentrum für Mechatronik und Automatisierungstechnik – in Saarbrücken im Bereich Montageverfahren und -automatisierung tätig.

Abstract

AI in Picking – Optimized Order Allocation Using Reinforcement Learning. Companies in high-wage countries are faced with the challenge to make logistic processes as efficient and sustainable as possible, especially manual order picking. Companies must find a balanced approach between optimizing pure picking time and ensuring the even utilization of warehouse workers. Only in this way long-term business goals can be reconciled with worker satisfaction. This article presents a practical approach to optimizing these logistic processes based on the potentials of artificial intelligence, specifically Reinforcement Learning.

Förderhinweis

Diese Arbeit entstand im Rahmen des vom Bundesministerium für Bildung und Forschung geförderten Projektes „Virtuelle Arbeitsgestaltung & Technologien für Innovationen im

Strukturwandel“ (ViSAAR; Förderkennzeichen 02L20B088). Ziel des Forschungsprojekts ViSAAR ist es, kleine und mittelständische Unternehmen (KMU) in strukturschwachen Regionen durch innovative organisatorische und digitale Lösungen im Bereich des ortsunabhängigen Arbeitens zukunftsfähig aufzustellen. Weitere Informationen unter www.visaar.de.

Schlüsselwörter

Maschinelles Lernen, Reinforcement Learning, Optimierte Auftragszuweisung, Kommissionierung, Logistik

Keywords

Reinforcement Learning, Optimized Order Assignment, Commissioning, Logistics

Bibliography

DOI:10.1515/zwf-2024-0132

ZWF 120 (2025) Special Issue; page 274 – 279

Open Access. © 2025 bei den Autoren,

publiziert von De Gruyter. 

Dieses Werk ist lizenziert unter der Creative Commons Namensnennung 4.0 International Lizenz.

ISSN 0947-0085 · e-ISSN 2511-0896