

Entscheidungsbäume und bestärkendes Lernen zur dynamischen Auswahl von Reihenfolgeregeln in einem flexiblen Produktionssystem

Using Decision Trees and Reinforcement Learning for the Dynamic Selection of Sequencing Rules in a Flexible Manufacturing System

Jens Heger, Mazhar Zein El Abdine, Sughanthan Sekar, Thomas Voß,
Leuphana Universität Lüneburg, Lüneburg (Germany),
heger@leuphana.de, abdine@leuphana.de, sughanthan.sekar@stud.leuphana.de,
voss@leuphana.de

Abstract: The integration of machine learning methods can improve the performance of a manufacturing system by adjusting priority rules dynamically. In this paper, three methods for adjusting the k-factors of the ATCS sequencing rule were trained: neural networks, decision trees and reinforcement learning. They are evaluated in a static and a dynamic scenario. The required dataset was synthetically generated using a discrete event simulation, where product mix and system utilization were varied within a flow shop environment. Across all scenarios, it is shown that all three methods can improve the performance. On par, RL and NN can reduce the mean tardiness by 15% and compensate for unplanned product mix changes.

1 Einleitung und Problemstellung

Die Reihenfolgebildung von Vorgängen an einer Maschine kann unter sich ändernden Bedingungen, wie zum Beispiel Maschinenausfall schwierig sein. Da die Anwendung von zentralen und statischen Lösungsverfahren in komplexen und unsicheren Szenarien nicht geeignet ist, stellen dezentrale Reihenfolge-Regeln eine praktikable Option dar. Diese nutzen lokal verfügbare Informationen zur schnellen Entscheidungsfindung. Es existiert jedoch keine Regel, die alle anderen bei unterschiedlicher Systemleistung übertrifft.

Aus diesem Grund wird eine Hyperheuristik entwickelt, um Gewichtungsfaktoren eine Prioritätsregel zur Reihenfolgebildung dynamisch, auf Basis des Systemstatus, auszuwählen und anzupassen. Basierend auf mehreren Trainingsszenarien unter Berücksichtigung stochastischer Einflüsse, wie z.B. variierende Ankunftszeiten oder wechselnde Anteile im Produktmix, werden die Vorteile von dynamischer Anpassung der Regel vorgestellt. Für die dafür benötigte Schätzung des

Systemverhaltens zur Auswahl und Anpassung der Gewichte der Regel, passend zum aktuellen Zustand, haben unterschiedlichste Modelle des maschinellen Lernens sehr gute Ergebnisse geliefert (Mouelhi-Chibani und Pierreval, 2010; Heger, 2014; Shiue et al., 2018). Bei der Verwendung dieser Methoden stellt sich aber nicht zuletzt die Frage nach der Menge an Trainingspunkten, dem Nachlernen von Wissen in neuen Szenarien und der Tauglichkeit, Generalisierbarkeit und Nachvollziehbarkeit der verwendeten Methoden (Priore et al., 2006; Priore et al., 2018; Usuga Cadavid et al., 2020). Das dabei generierte Wissen und Verständnis der Handlungen wird anstelle von einfacher Vorhersage und Blackbox Optimierung zunehmend bevorzugt (Nunes und Jannach, 2017; Rehse et al., 2019). An dieser Stelle kann der verwendete Ansatz der Hyperheuristik zur Auswahl unterschiedlicher Reihenfolgeregeln in Kombination mit nachvollziehbaren Lernmethoden (z.B. Entscheidungsbäumen) seine Tauglichkeit beweisen.

2 Stand der Technik

Auf Grund ihrer einfachen Nachvollziehbarkeit und der sehr kurzen Rechendauer ist die Verwendung von Prioritätsregeln zur Reihenfolgebildung, d.h. zur Auswahl des nächsten Auftrags zur Bearbeitung durch die Maschinen, in der Industrie sehr beliebt. Dabei ist zu beachten, dass über 100 Regeln bekannt sind, die je nach Szenario unterschiedliche Leistungen bringen (Panwalkar und Iskander, 1977). Im Verlauf der Jahre wurden kombinierte Prioritätsregeln entwickelt, die mehrere Auftragsattribute gleichzeitig betrachten, um die Leistung des Systems zu verbessern. So auch die Regel „Apparent Tardiness Cost“, die neben der gewichteten Prozesszeit auch den geplanten Fertigstellungszeitpunkt und einen Gewichtungsfaktor (k_1) betrachtet (Vepsalainen und Morton, 1987). Mit Blick auf die Rüstzeiten wurde die Regel dann um einen Rüstzeitterm erweitert und wird seither als „Apparent Tardiness Cost with Setups“ (ATCS) beschrieben. Der zusätzliche Term bezeichnet das Verhältnis zwischen reihenfolgeabhängiger Rüstzeit und durchschnittlicher Rüstzeit, multipliziert mit dem zweiten Gewichtungsfaktor (k_2). Die Regel wird in der in Gleichung (1) gezeigten Form für die folgende Studie verwendet. Die Kombination von drei Attributen und die Verwendung von zwei Gewichtungsfaktoren ermöglichen es, bei einer korrekten Dimensionierung über eine breite Masse an Szenarien hinweg gute Leistung zu erzielen (Lee et al., 2002).

$$Z_i^t = \frac{w_i}{p_i} \exp\left(-\frac{(d_i - t - p_i)^+}{k_1 \bar{p}}\right) \exp\left(-\frac{s_{i,l}}{k_2 \bar{s}}\right) \quad (1)$$

Wissend, dass die Systemleistung stark von der korrekten Auswahl der k-Faktoren passend zur Systemauslastung abhängt, werden diese dynamisch an die Situation in der Fertigung angepasst. Es handelt sich bei der dynamischen Anpassung folglich um eine Hyperheuristik. Zum Aufbau der Wissensbasis über den Zusammenhang der k-Faktoren und der resultierenden Leistung müssten allerdings alle möglichen Kombinationen aus k-Faktoren, Produktmix und Systemzustand simuliert werden. Da auf Grund der Komplexität in realen Systemen nicht alle möglichen

Kombinationen an Einflussfaktoren simuliert werden können, werden häufig definierte Kombinationen aus Systemzuständen simuliert und die unbekanntes Situationen durch ein Regressionsverfahren geschätzt.

Dabei stellt die Verwendung von Neuronalen Netzen (NN) die Basiskennlinie dar. Die Kombination von NN zur Vorhersage der Systemleistung wurde ausführlich hinsichtlich der Abhängigkeit von den k -Faktoren und der Systemauslastung betrachtet (Mönch et al., 2006; Mouelhi-Chibani und Pierreval, 2010; Heger, 2014; Heger et al., 2016). Trotz guter Ergebnisse ist hierbei jedoch festzustellen, dass die NN grundsätzlich als Blackbox verwendet werden und keine Aussage über den Einfluss bestimmter Faktoren ermöglichen. Aus diesem Grund wird die Verwendung von NN als Basislinie mit der Verwendung von Entscheidungsbäumen und bestärkendem Lernen verglichen (Rai, 2020).

Die Verwendung von Entscheidungsbäumen erhielt im Kontext von erklärbaren Modellen (engl.: Explainable AI) des maschinellen Lernens deutliche Aufmerksamkeit (Rai, 2020; Puiutta; Veith, 2020).

Im Gegensatz zu komplexen Modellen, wie tiefe NNs, die nicht-interpretierbare Blackbox-Modelle erzeugen, sind Entscheidungsbäume regelbasierte Modelle, die dem Benutzer eine intuitive Darstellung von Regeln und Abläufen bieten. An jedem Knoten eines Entscheidungsbaums wird eine bestimmte Zielfunktion geprüft. Das Ergebnis liefert den Pfad zum neuen Knoten. Die Struktur wiederholt sich, bis eine bestimmte Voraussetzung getroffen wird.

Regeln können von Pfaden durch den Entscheidungsbaum abgeleitet und Entscheidungsbäume sowohl für Klassifikations- als auch für Regressionsaufgaben verwendet werden. Somit eignen sie sich beispielsweise zur dynamischen Auswahl von Prioritätsregeln, jedoch mit relativ niedriger Leistung in unbekanntes Szenarien (Shahzad und Mebarki, 2016). Zudem sind sie, zusammen mit weiteren baumbasierten Verfahren, wie Random Forest und XGBoost, in deren Anwendungen weit verbreitet. Diese Verfahren, die auf Entscheidungsbäume basieren, haben allerdings ein gewisses Maß an Interpretierbarkeit eingebüßt, um bessere Schätzungen und Generalisierbarkeit dieser zu erreichen. Dennoch werden sie immer häufiger mit weiteren Funktionalitäten ausgestattet, um die Interpretation zu verbessern (Lundberg et al., 2020).

Die Verwendung von bestärkendem Lernen hat als Hyperheuristik bei der dynamischen Auswahl von Reihenfolgeregeln bereits gute Ergebnisse erzielt. Studien zeigen, dass die inhärenten Vorteile des bestärkenden Lernens, im Gegensatz zu den Methoden des überwachten Lernens, bei der direkten Interaktion mit dem System liegen. Der Agent lernt die richtigen Verhaltensweisen basierend auf dem beobachteten Verhalten und der erhaltenen Rückmeldung. Konkret handelt es sich bei dem Anwendungsfall um die Auswahl von Prioritätsregeln für alle Maschinen im System (Heger und Voss, 2020, 2021). Die Autoren zeigen, dass ausgehend von der beobachteten Systemauslastung und Warteschlangen die Leistung durch dynamische Auswahl von Reihenfolgeregeln verbessert werden kann. Analog dazu zeigen andere Autoren in ihrer Studie, dass die dynamische Anpassung von maschinenspezifischen Regeln in verschiedenen Szenarien signifikante Leistungsverbesserungen ermöglicht (Shiue et al., 2018, 2020).

In diesem Beitrag wird geprüft, inwieweit die drei beschriebenen Verfahren zur Unterstützung der Auswahl von geeigneten k -Faktoren taugen. Dabei ist zu prüfen,

inwieweit die Verwendung von nachvollziehbaren Handlungen zu einer Reduktion der Leistung führt. Das Zusammenspiel zwischen Leistung und Erklärbarkeit wird im Kontext des vorgestellten Szenarios genauer untersucht. Zusätzlich ist zu prüfen, unter Verwendung der trainierten Hyperheuristiken, ob diese in einem unbekanntem Szenario immer noch in der Lage sind, die k-Faktoren entsprechend dem Systemstatus auszuwählen und dynamisch zu ändern, und dadurch eine robuste Leistung zu erzielen.

3 Simulation und Szenario

Die Studie wird im Rahmen eines Fertigungssystems mit reihenfolgeabhängigen Rüstzeiten eingesetzt und durchgeführt. Dabei werden ungeplante und unbekannte Änderungen, wie Produktmix-Wechsel und Auslastungsschwankungen, hinzugefügt, um ein Verhalten der verschiedenen Methoden in unbekanntem Szenarien betrachten zu können. Im Folgenden ist das Szenario detailliert beschrieben:

Tabelle 1: Detaillierte Szenariobeschreibung

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

Der besondere Fokus liegt dabei auf den reihenfolgeabhängigen Rüstzeiten, die im Folgenden dargestellt sind. Je nach Produktmix sind die Anteile der vier Produkttypen unterschiedlich, was zu mehr oder weniger großen Rüstzeitanteilen führt. So ist davon auszugehen, dass ein Produktmix mit den ersten drei Produktfamilien deutlich weniger Rüstzeit benötigt als ein Produktmix, der alle vier Produktfamilien enthält. Dabei kann die Matrix in (2) wie folgt gelesen werden: die Rüstzeit von Familie 1 auf Familie 2 beträgt 5 Minuten; die Rüstzeit von Familie 1 auf Familie 4 beträgt 25 Minuten. Bei einer durchschnittlichen Bearbeitungszeit von 50 Minuten kann die Rüstzeit massiven Einfluss auf die Leistung des Systems haben.

$$\begin{pmatrix} 0 & 5 & 10 & 25 \\ 5 & 0 & 10 & 25 \\ 5 & 5 & 0 & 25 \\ 5 & 5 & 10 & 0 \end{pmatrix} \quad (2)$$

Zur Messung der Leistung werden die durchschnittliche Verspätung sowie die durchschnittliche Durchlaufzeit dokumentiert. Die durchschnittliche Verspätung ergibt sich aus dem geplanten Fertigstellungszeitpunkt und der tatsächlichen Fertigstellung. Dabei ist zu beachten, dass zu früh gefertigte Aufträge mit einer Verspätung von 0 bewertet werden. Der Fertigstellungszeitpunkt berechnet sich aus der Summe der Startzeit und der durchschnittlichen geplanten Bearbeitungszeit multipliziert mit einem Verspätungsfaktor (engl. Due Date Factor). Dieser ist so gewählt, dass eine gewisse Rüst-, Wart- und Transportzeit zwischen den Maschinen akzeptabel ist.

Für das Erstellen von Regressionsmodellen werden in einer umfangreichen Parameterstudie Trainingsdaten mit Hilfe von ereignisdiskreter Simulation generiert. Die Verwendung von Simulation ermöglicht es, die Länge und Breite des Datensatzes selbst zu bestimmen. Diese generierten Daten bilden die Grundlage für die wissensbasierten Ansätze der dynamischen Anpassung. Das Simulationsmodell steht in diesem Fall als Trainingsumgebung für das bestärkende Lernen zur Verfügung.

In der Abbildung 1 ist das Verfahren beschrieben. Beginnend unten in der Mitte steht das Simulationsmodell. Durch die Parameterstudie zur Betrachtung der Leistung über die unterschiedlichen Systemzustandskombinationen ist es möglich, Trainingsdaten für die unterschiedlichen Regressionsmodelle zu erstellen.

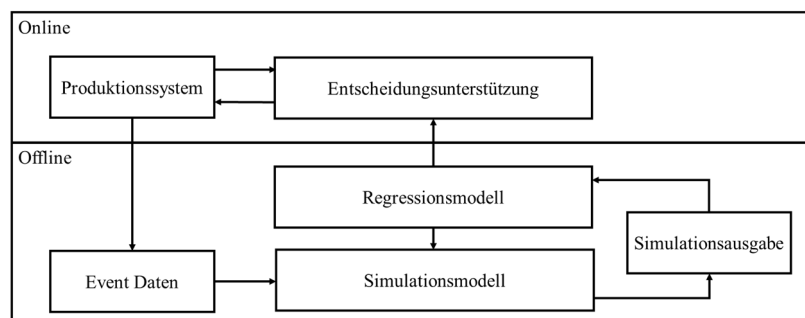


Abbildung 1: Konzept zur Verwendung von Regressionsmodellen zur Entscheidungsunterstützung

Diese werden dann als Entscheidungsunterstützung im realen Produktionssystem genutzt. Ausgehend von dem aktuellen Systemzustand (z.B. Maschinenauslastung und Produktmix) schlägt die Entscheidungsunterstützung einen neuen k -Werte-Paar vor. Eine Simulation aller k -Werte würde in der Onlineanwendung viel Zeit in Anspruch nehmen.

In der Abbildung 2 ist das Verfahren zur dynamischen Anpassung schematisch dargestellt. Dabei wird die Leistung des Systems über die Zeit und unter sich verändernden Zuständen sowie einer Auswahl an k -Werten betrachtet. Je nach Ausprägung des Leistungsindikators gilt es, den Wert zu minimieren oder zu maximieren; in diesem Beitrag wird die Minimierung der durchschnittlichen Verspätung betrachtet. Zu definierten Zeitpunkten wird, unabhängig vom Verfahren, durch die Entscheidungsunterstützung eine Aussage über die für die Situation am

besten geeigneten k -Werte zur Verbesserung der Leistung getätigt, welche dann für einen definierten Zeitraum konstant gehalten werden. Es ist zu evaluieren, ob und bis zu welchem Grad die Schwankung durch die inhärenten stochastischen Unsicherheiten des Simulationsmodells oder durch die Auswahl der k -Faktoren geschieht. Im Rahmen der Simulationsstudie kann weiterhin evaluiert werden, wie die Häufigkeit der Anpassung Einfluss auf die Leistung nimmt.

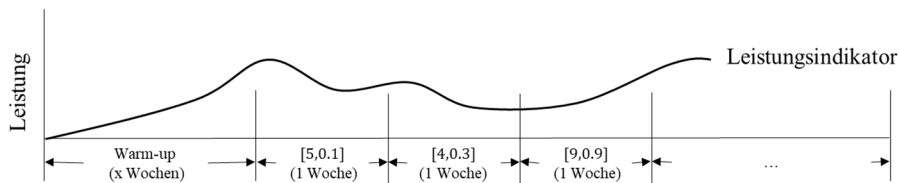


Abbildung 2: Systematische Abbildung des Vorgehens über den Simulationsverlauf

Eine Parameteranalyse wird in dem Zusammenhang durchgeführt, um herauszufinden, welche Beobachtungen signifikanten Einfluss haben und welche nicht. Die Datenvorverarbeitung wie Standardisierung, One-Hot-Encoding und einer Kombination daraus werden unabhängig von der verwendeten Methode durchgeführt, auf Grund des Simulationsfokus werden die Auswirkungen allerdings nicht detailliert evaluiert.

4 Auswertung

Als Trainingsdatensatz wurde eine Parameterstudie mit allen möglichen Kombinationen aus k_1 -Werten von 1 bis 10 und k_2 -Werten von 0.01 bis 1.01 unter 7 verschiedenen Einlastungen von 85 % bis 95 % ebenso wie 12 verschiedenen Produktmischen mit unterschiedlichen Rüstanteilen durchgeführt. Für jede der 9240 individuellen Parameterkombinationen wurden 5 Replikationen durchgeführt. Der verwendete Datensatz umfasst damit 46 200 Samples. Die Beobachtungen aus dem System waren die durchschnittliche Verspätung, die durchschnittliche Durchlaufzeit sowie die durchschnittliche Maschinenauslastung des Szenarios. Dabei sind die k -Faktoren, die Auslastung sowie die Leistungsindikatoren kontinuierliche Variablen. Der Produktmix wird als kategorische Variable betrachtet.

In den Rohdaten ist klar zu erkennen, dass unterschiedliche Produktmische bei derselben Einlastung und der Verwendung von statischen k -Werten unterschiedliche Leistungen bringen. Exemplarisch ist dies für zwei Produktmische in Abbildung 3 zu erkennen. Ein kleiner k_2 -Wert, welcher vorteilhaft für Produktmix 1 ist, würde bei Produktmix 6 zu einer Verschlechterung der Leistung um 30 % führen. Weiterhin zeigen die Rohdaten, dass auf Grund der unterschiedlichen Rüstzeitanteile pro Produktmix die Auslastung des Systems bei konstanter Einlastung bis zu 5 % variieren kann. In dieser Studie ist eine Einlastung von 85 % vorgesehen. Es ist festzuhalten, dass eine zu geringe Einlastung dazu führt, dass kein Verbesserungspotenzial möglich ist. Diesen Sachverhalt gilt es über mehrere Produktmische zu erkennen und zur Verbesserung der Leistung zu nutzen.

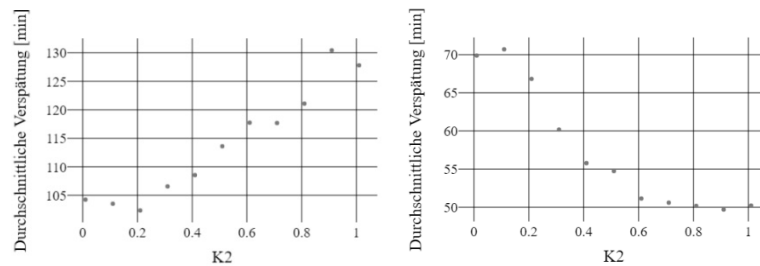


Abbildung 3: Leistung von Produktmix 1 (links) und Produktmix 6 (rechts) bei 85 % Einlastung unter unterschiedlichen k_2 -Werten.

Zusätzlich zum ersten Datensatz mit 46200 Datenpunkte wird ein zweiter Datensatz mit 13860 Datenpunkte (entspricht 30 % des ersten Datensatzes) generiert. Im Anschluss wurden die NN ebenso wie die Entscheidungsbäume (DT) auf beiden Sätzen trainiert, um eine Aussage über die Verbesserung der Präzision mit mehr Datenpunkten machen zu können. Eine gute Konfiguration für die NN sowie die DT wurde mit Hilfe eines vorgelagerten Grid-Search-Verfahrens ermittelt. Das NN wurde mit scikit learn als Multi Layer Perceptron in Python implementiert. Die Parametrierung wurde mit Hilfe eines Grid-Search-Verfahren durchgeführt. Das resultierende, zweilagige Netz mit 10 Neuronen in der ersten und 30 Neuronen in der zweiten Schicht hatte die Aktivierungsfunktion „relu“. In Kombination mit dem Solver „adam“ zeigte eine Minibatchgröße von 500 Samples gute Ergebnisse. Die initiale Lernrate wurde mit 0.01 festgelegt. Eine L2-Regulierung wurde durchgeführt.

Die DT wurden mit scikit learn als Entscheidungsbaum Regressor in Python implementiert. Die endgültige Konfiguration wurde mit Hilfe eines Grid-Search-Verfahren ermittelt. In diesem Rahmen zeigte sich, dass die Verwendung einer maximalen Tiefe von 5, mit mindestens 4 Samples pro Blatt zu guten Ergebnisse führte. Der RL-Agent wurde mit Hilfe der Pathmind Software-as-a-Service-Plattform mit 12000 Simulationsläufen trainiert. In dem Zuge wurde das ereignisdiskrete Simulationsmodell als alleinstehendes Java-File exportiert und auf der Plattform, unabhängig von lokalen Ressourcen, über 12 Stunden trainiert. Dabei wurden verschiedene Hyperparameterkonfigurationen im Rahmen von populationsbasiertem Training automatisch evaluiert und die beste Konfiguration für das Szenario gefunden. Pathmind verwendet Ray und RLlib für das Training des Agenten. Die Strategie des Agenten wurde durch Proximal Policy Optimization trainiert.

Im Rahmen der Evaluation werden die drei Methoden in der Onlineverwendung getestet. Dabei wird für ein bekanntes Szenario bei statischer Einlastung und bekanntem Produktmix die Leistung dokumentiert. Im Folgenden handelt es sich um eine Einlastung von 85 % und den Produktmix 1 von oben. In der Abbildung 4 ist exemplarisch das Verhalten des Agenten, trainiert mit bestärkendem Lernen, abgebildet. Wie bereits oben zu sehen war, ist für den Produktmix 1 ein niedriger k_2 -Wert von Vorteil. Es ist klar zu erkennen, dass bei Schwankungen der durchschnittlichen Verspätung (linke Y-Achse) im System über die Zeit (X-Achse) die verwendeten k_2 -Werte der ATCS-Regel (rechte Y-Achse) angepasst wird.

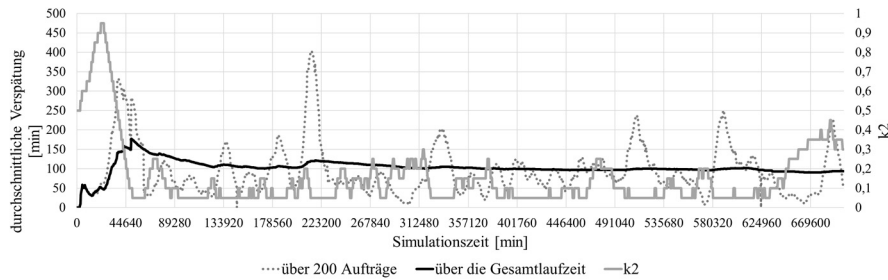


Abbildung 4: Änderung der k_2 -Werte durch den RL-Agenten zur dynamischen Anpassung an die Systemauslastung

Über 30 Replikationen zeigt sich, dass bei einer statischen Einlastung und bei einem bekannten Produktmix die dynamische Anpassung der k -Werte mit RL einen positiven Einfluss hat, sich aber nicht signifikant zur statischen Auswahl von k -Werten unterscheidet. Besonders interessant (siehe Abbildung 5) ist an dieser Stelle die schlechte Leistung der NNs; es ist davon auszugehen, dass das dynamische Umschalten und Anpassen durch das NN einen negativen Einfluss auf die Leistung im statischen Szenario hat (Priore et al., 2006).

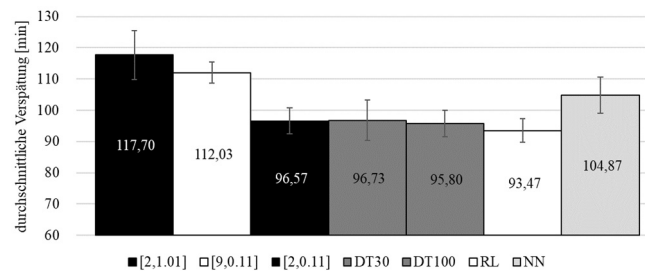


Abbildung 5: Vergleich der Leistung bei statischen k -Werten und der dynamischen Anpassung in einem bekannten Szenario

Im zweiten Evaluationsszenario wird die dynamische Anpassung bei einer statischen Einlastung unter wechselnden Produktmixen getestet. Im Szenario wird über $\frac{1}{4}$ der Simulationszeit ein neuer Produktmix (in diesem Fall Produktmix 6) im System betrachtet. Dieses Verhalten ist exemplarisch an das Weihnachtsgeschäft angelehnt. Es zeigt sich in der Abbildung 6, dass die Auswahl von guten statischen k -Werten bereits zu guter Leistung führt. Im Vergleich dazu bringen die DTs, die NNs ebenso wie der RL-Agent eine zusätzliche signifikante Verbesserung von bis zu 15 %. Dabei ist der RL-Agent noch 3 % besser als die Verwendung der DTs. Im Gegensatz zum statischen Szenario kann das NN seine Vorteile bzgl. der Generalisierbarkeit von Verhalten zeigen. Die vergleichbare Leistung von RL und NN ist nachvollziehbar, da RL zur Schätzung der Belohnung NNs verwendet.

Über die Evaluation in beiden Szenarien zeigt sich, dass DTs bekanntes Systemverhalten sehr gut wiedergeben können und bis zu einem gewissen Grad dynamische Verhalten beschreiben können. Die Verwendung von NN und RL ist besonders in Szenarien mit unbekanntem Verhalten im Vorteil und kann zu einer Verbesserung der Leistung von bis zu 15 % führen.

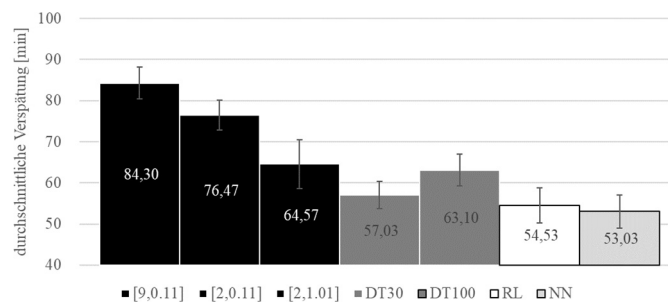


Abbildung 6: Vergleich der Leistung bei statischen k -Werten und der dynamischen Anpassung in einem unbekanntem Szenario

5 Zusammenfassung und Ausblick

Die dynamische Anpassung von Prioritätsregeln mit Hilfe von verschiedenen Methoden des maschinellen Lernens kann zu einer Verbesserung der Leistung führen. Im Rahmen des Beitrags wurden drei Verfahren zur Anpassung der k -Faktoren der ATCS-Regel trainiert und über zwei Szenarien evaluiert. Am Beispiel einer Fließfertigung wurde mit Hilfe einer ausführlichen Parameterstudie ein Datensatz über die Zusammenhänge zwischen Produktmix, k -Faktoren und Systemauslastung erstellt. Dieses wurde anschließend als Trainingsgrundlage für DT und NN genutzt. Das ereignisdiskrete Modell wurde als Trainingsumgebung für den RL-Agenten verwendet. Der Vergleich im statischen Szenario zeigt, dass DT und RL in der Lage sind, die Leistung der statischen k -Faktoren abzubilden. Im Rahmen des Trainings zeigte sich, dass die Verwendung von DTs bei qualitativen Aussagen helfen kann. Im dynamischen Szenario zeigt sich, dass alle drei Verfahren die Leistung verbessern können. Gleichauf können RL und NN die durchschnittliche Verspätung um 15 % reduzieren und ungeplante Produktmixwechsel kompensieren. Im nächsten Schritt soll eine tiefe und detaillierte Analyse der dynamischen Anpassung über mehrere Produktmixe und unbekanntem Szenarien durchgeführt werden.

Literaturverzeichnis

- Heger, J.: Dynamische Regelselektion in der Reihenfolgeplanung. Wiesbaden: Springer Fachmedien Wiesbaden 2014.
- Heger, J.; Branke, J.; Hildebrandt, T.; Scholz-Reiter, B.: Dynamic adjustment of dispatching rule parameters in flow shops with sequence-dependent set-up times. *International Journal of Production Research* 54 (2016) 22, S. 6812–6824.
- Heger, J.; Voss, T.: Dynamically Changing Sequencing Rules with Reinforcement Learning in a Job Shop System with Stochastic Influences. In: Bae, K.-H.; Feng, B.; Kim, S.; Lazarova-Molnar, S.; Zheng, Z.; Roeder, T.; Thiesing, R. (Hrsg.): *Proceedings of the Winter Simulation Conference 2020*, S. 1608–1618.
- Heger, J.; Voss, T.: Dynamically adjusting the k -values of the ATCS rule in a flexible flow shop scenario with reinforcement learning. *International Journal of Production Research* (2021).

- Lee, Y.H.; Jeong, C.S.; Moon, C.: Advanced planning and scheduling with outsourcing in manufacturing supply chain. *Computers & Industrial Engineering* 43 (2002) 1, S. 351–374.
- Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.-I.: From Local Explanations to Global Understanding with Explainable AI for Trees. *Nature machine intelligence* 2 (2020) 1, S. 56–67.
- Mönch, L.; Zimmermann, J.; Otto, P.: Machine learning techniques for scheduling jobs with incompatible families and unequal ready times on parallel batch machines. *Engineering Applications of Artificial Intelligence* 19 (2006) 3, S. 235–245.
- Mouelhi-Chibani, W.; Pierreval, H.: Training a neural network to select dispatching rules in real time. *Computers & Industrial Engineering* 58 (2010) 2, S. 249–256.
- Nunes, I.; Jannach, D.: A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction* 27 (2017) 3-5, S. 393–444.
- Panwalkar, S.S.; Iskander, W.: A Survey of Scheduling Rules. *Operations Research* 25 (1977) 1, S. 45–61.
- Priore, P.; La Fuente, D. de; Puente, J.; Parreno, J.: A comparison of machine-learning algorithms for dynamic scheduling of flexible manufacturing systems. *Engineering Applications of Artificial Intelligence* 19 (2006) 3, S. 247–255.
- Priore, P.; Ponte, B.; Puente, J.; Gómez, A.: Learning-based scheduling of flexible manufacturing systems using ensemble methods. *Computers & Industrial Engineering* 126 (2018), S. 282–291.
- Puiutta, E.; Veith, E.M.: *Explainable Reinforcement Learning: A Survey*, 2020.
- Rai, A.: Explainable AI: from black box to glass box. *Journal of the Academy of Marketing Science* 48 (2020) 1, S. 137–141.
- Rehse, J.-R.; Mehdiyev, N.; Fettke, P.: Towards Explainable Process Predictions for Industry 4.0 in the DFKI-Smart-Lego-Factory. *KI - Künstliche Intelligenz* 33 (2019) 2, S. 181–187.
- Shahzad, A.; Mebarki, N.: Learning Dispatching Rules for Scheduling: A Synergistic View Comprising Decision Trees, Tabu Search and Simulation. *Computers* 5 (2016) 1, S. 3.
- Shiue, Y.-R.; Lee, K.-C.; Su, C.-T.: Real-time scheduling for a smart factory using a reinforcement learning approach. *Computers & Industrial Engineering* 125 (2018), S. 604–614.
- Shiue, Y.-R.; Lee, K.-C.; Su, C.-T.: A Reinforcement Learning Approach to Dynamic Scheduling in a Product-Mix Flexibility Environment. *IEEE Access* 8 (2020), S. 106542–106553.
- Usuga Cadavid, J.P.; Lamouri, S.; Grabot, B.; Pellerin, R.; Fortin, A.: Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0. *Journal of Intelligent Manufacturing* 31 (2020) 6, S. 1531–1558.
- Vepsäläinen, A.P.; Morton, T.E.: Priority rules for job shops with weighted tardiness costs. *Management Science* 33 (1987) 8, S. 1035–1047.