Simulation in Production
and Logistics 2015
Markus Rabe & Uwe Clausen (eds.)
Fraunhofer IRB Verlag, Stuttgart 2015

Simulationsbasierte Optimierung der Reihenfolgeplanung am Beispiel eines Liniensorters in der Automobilindustrie

Simulation-based Scheduling for a FIFO Line Sorter in a Car Manufacturing Scenario

Jens Heger, Leuphana Universität Lüneburg, Lüneburg (Germany), Jens.Heger@leuphana.de

Torsten Hildebrandt, BIBA – Bremer Institut für Produktion und Logistik GmbH an der Universität Bremen, Bremen (Germany), hil@biba.uni-bremen.de

Abstract: This study analyses different scheduling methods for a FIFO multiple line buffer in a scenario with sequence-dependent setup times. A simple selection rule working after the shortest-setup-time-next principle is compared to an approach with a genetic algorithm and a MILP-Solver combination. Although the scenario is np-complete the occurring small instances can be solved in acceptable time. The scenario is implemented in two different simulation systems (Tecnomatix Plant Simulation and jasima). Results show that the rule-based scheduling is outperformed by both other algorithms, where the MILP-Solver achieves the highest improvement in mean flowtime with over 26%.

1 Einleitung und Szenariobeschreibung

Die in dieser Studie betrachtete Problemstellung ist aus der Automobilindustrie abgeleitet und betrachtet einen Liniensorter, der aus mehreren parallelen FIFO-Warteschlangen besteht und einer Arbeitsstation (hier: Lackierstation) vorgeschaltet ist (Eley 2012). Der Liniensorter wird eingesetzt, da an der Arbeitsstation reihenfolgeabhängige Rüstzeiten anfallen und diese zur Verbesserung der mittleren Durchlaufzeit reduziert werden sollen. Es werden 20 verschiedene Rüstzustände (bspw. Farben) angenommen, die je nach Vorgänger eine andere Rüstzeit benötigen.

Es gibt zwei Entscheidungspunkte, an denen unterschiedliche Strategien angewendet werden können. Einerseits werden neu eintreffende Karosserien einer Linie zugeordnet und dort eingelagert und anderseits kann die Arbeitsstation aus den fünf Linien eine der jeweils ersten Karosserien auswählen. Eley (2012) hat dazu bereits einfache Greedy-Heuristiken vorgestellt und diese mit einem Genetischen Algorithmus in einer Simulationsstudie verglichen. Die Einlastungsstrategien sind so gewählt, dass einerseits zufällig eine Verteilung vorgenommen wird und andererseits

eine zweite Strategie angewendet wird, die die Karosserien wenn möglich in eine leere Warteschlange einlagert. Sind in allen Warteschlangen bereits Karosserien vorhanden, wird die Linie ausgewählt bei der zwischen der letzten Karosserie und der einzusortierenden die geringste Rüstzeit benötigt wird.

Als erste Auslagerungsstrategien wird eine einfache Regel angewendet, die jeweils die Karosserie auswählt, die zur zuletzt bearbeiteten die geringste Rüstzeit aufweist. Die zweite Auslagerungsstrategie betrachtet das Ausgangsproblem als kombinatorisches Optimierungsproblem, das mit einem Genetischen Algorithmus gelöst wird. Ziel ist es, die Rüstzeiten und damit auch die Durchlaufzeiten zu minimieren.

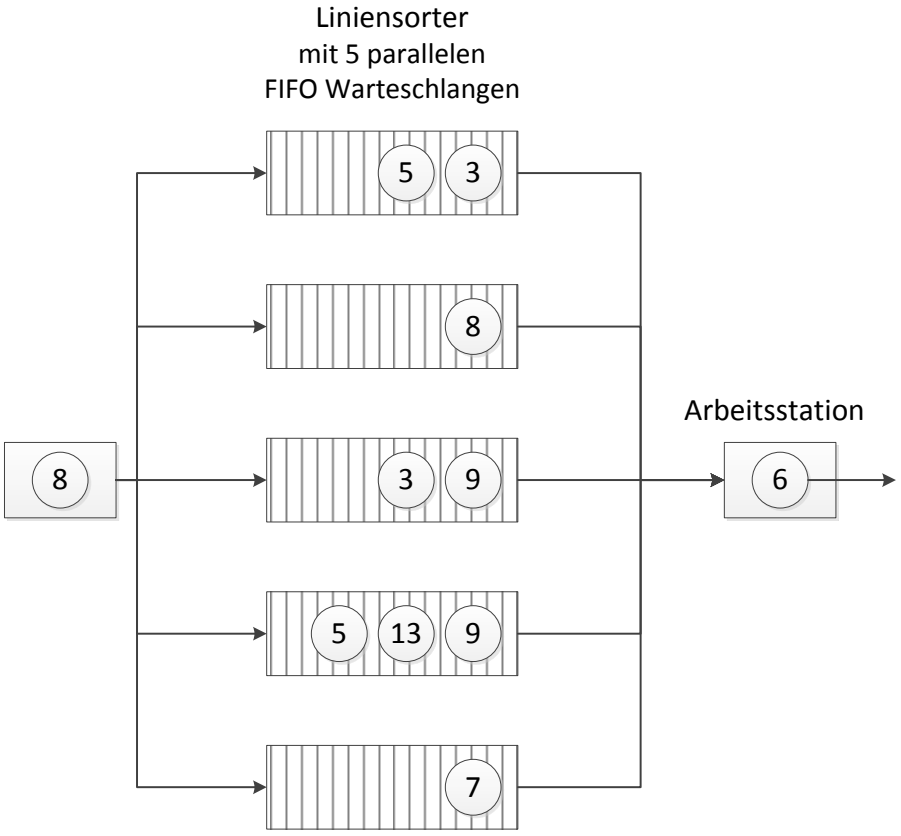

Abbildung 1: *Paralleler Liniensorter bestehend aus FIFO-Warteschlangen; Zahlen stellen den Rüstzustand dar (Eley 2012)*

Das betrachtete Szenario ist in Abbildung 1 dargestellt. Beispielhaft sind einige Aufträge in die Warteschlagen eingereiht worden. Die Arbeitsstation kann, nachdem sie den Auftrag mit Rüstzustand 6 abgearbeitet hat, zwischen den jeweils vorderen Aufträgen einen als nächsten auswählen (3, 8, 9, 9, 7). Die dafür anfallenden Rüstkosten sind in der nachfolgenden Tabelle 1 aufgeführt.

Tabelle 1: *Rüstzeiten in Sekunden*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	**237**	353	**324**	91	**274**	434	**311**	102	**214**	228	**64**	253	**103**	170	**162**	196	**108**	167	**371**
2	126	**0**	181	**234**	125	**64**	355	**146**	227	**187**	99	**98**	143	**228**	205	**268**	282	**142**	173	**134**
3	259	**197**	0	**53**	94	**235**	395	**316**	62	**369**	296	**178**	95	**59**	284	**122**	205	**222**	128	**317**
4	346	**284**	185	**0**	151	**322**	470	**396**	149	**275**	383	**124**	42	**244**	231	**210**	335	**168**	163	**418**
5	169	**307**	151	**204**	0	**217**	334	**298**	213	**165**	245	**233**	174	**210**	339	**143**	184	**277**	148	**441**
6	61	**298**	159	**112**	152	**0**	372	**82**	163	**276**	289	**126**	79	**164**	232	**223**	257	**85**	200	**432**
7	294	**303**	204	**19**	170	**325**	0	**328**	168	**253**	321	**143**	60	**170**	160	**228**	316	**187**	182	**200**
8	420	**210**	219	**118**	270	**275**	363	**0**	217	**393**	309	**243**	85	**275**	349	**100**	70	**3**	106	**345**
9	197	**135**	259	**285**	260	**135**	252	**217**	0	**83**	234	**233**	192	**299**	340	**60**	286	**220**	66	**269**
10	65	**95**	163	**72**	156	**4**	376	**85**	167	**0**	193	**129**	38	**168**	236	**186**	155	**89**	160	**229**
11	371	**308**	101	**155**	195	**150**	329	**124**	163	**247**	0	**276**	134	**96**	380	**224**	193	**127**	75	**329**
12	195	**245**	178	**222**	27	**154**	361	**236**	215	**150**	272	**0**	189	**116**	106	**141**	110	**44**	146	**379**
13	379	**317**	144	**33**	185	**197**	376	**279**	183	**266**	416	**158**	0	**143**	264	**243**	252	**202**	122	**452**
14	239	**176**	251	**221**	205	**176**	293	**258**	42	**125**	275	**178**	134	**0**	284	**102**	146	**214**	107	**311**
15	249	**186**	261	**231**	215	**186**	303	**268**	51	**135**	285	**188**	144	**10**	0	**112**	156	**224**	117	**321**
16	88	**118**	32	**65**	126	**27**	191	**108**	94	**23**	216	**152**	32	**26**	258	**0**	136	**112**	5	**252**
17	193	**123**	217	**233**	232	**187**	320	**115**	68	**151**	222	**205**	160	**27**	310	**129**	0	**118**	134	**257**
18	185	**189**	129	**162**	223	**124**	288	**181**	135	**120**	288	**249**	129	**93**	355	**97**	66	**0**	102	**324**
19	137	**198**	27	**80**	121	**75**	254	**157**	63	**146**	296	**199**	121	**21**	305	**123**	130	**160**	0	**332**
20	118	**148**	47	**100**	141	**57**	221	**138**	109	**53**	120	**182**	71	**106**	288	**169**	208	**142**	175	**0**

Im Folgenden wird kurz der Stand der Technik dargestellt und auf Vorarbeiten speziell zu Szenarien mit Rüstzeiten und Prioritätsregelsteuerung eingegangen. Anschließend werden der hier untersuchte Ansatz und die Simulationsstudie beschrieben. Dabei wird auch auf die unterschiedliche Simulationssoftware eingegangen. Nachfolgend werden die Ergebnisse der Studie aufgezeigt und diskutiert. Der Beitrag schließt mit einer Zusammenfassung und einem Ausblick zu zukünftigen Forschungsfragen.

2 Bekannte Ansätze für die Reihenfolgeplanung mit reihenfolgeabhängigen Rüstzeiten

In diesem Abschnitt werden zunächst allgemeine Ansätze zur Reihenfolgeplanung mit reihenfolgeabhängigen Rüstzeiten dargestellt. Anschließend wird auf Vorarbeiten und Implementierungsdetails zu dem hier untersuchten Szenario eingegangen.

2.1 Allgemeine Ansätze zur Reihenfolgeplanung mit reihenfolgeabhängigen Rüstzeiten

Das Problem der Reihenfolgeplanung mit reihenfolgeabhängigen Rüstzeiten wird schon seit langer Zeit aufgrund seiner hohen Praxisrelevanz untersucht. Eine Übersicht über vorgestellte Ansätze auf Basis von Prioritätsregeln geben etwa (Pickardt und Branke 2011). Je nach Zielkriterium (bspw. mittlere Durchlaufzeit, Verspätung etc.) führen verschiedene Regel zum besten Ergebnis. Weiterhin spielt auch das Verhältnis der Rüstzeit zur Bearbeitungszeit des Szenarios eine entscheidende Rolle. Beste Ergebnisse lieferten die Regeln FE[MMS; SPT] (van der Zee 2010), SNSPT (Kochhar und Morris 1987), ACTS (Lee und Pinedo 1997) und ECR (Chiang und Fu 2009). Eine weitere Verbesserung kann durch die dynamische Anpassung der ACTS-Regel auf Basis vorgelagerter Simulationen und der Leistungsprognose mithilfe von Gaußscher Prozesse Regression erreicht werden (Heger et al. 2015; Heger 2014).

Im Gegensatz zu den vorgestellten dezentralen regelbasierten Steuerungsverfahren existieren zentrale Ansätze, die sämtliche Informationen berücksichtigen. Da es sich aber bereits beim Ein-Maschinen-Problem mit reihenfolgeabhängigen Rüstzeiten um ein NP-vollständiges Problem handelt (Monma und Potts 1989), können optimale Verfahren, wie beispielsweise MILPs beziehungsweise mathematische Solver, nur von kleinen Instanzen Lösungen berechnen. Daher muss genau untersucht werden, ob ihr Einsatz in einem solchen Szenario in der Praxis möglich ist. Alternativ werden diese Verfahren häufig eingesetzt, um die Leistungsfähigkeit von nicht optimalen Heuristiken zu beurteilen (Kurz und Askin 2004; Naderi und Salmasi 2012). Als Heuristik bietet sich beispielsweise das Simulated Annealing an (Tan und Narasimhan 1997).

Alle diese beschrieben Ansätze und Untersuchungen haben gemein, dass sie von nur einem Puffer ausgehen. Es werden zwar teilweise parallele Maschinen untersucht, aber keine parallelen Puffer. Weiterhin wird davon ausgegangen, dass sämtliche Aufträge innerhalb des Puffers ausgewählt werden können.

2.2 Untersuchung einer Lackieranlage in der Automobilindustrie

Das in dieser Studie ausgewählte Szenario wurde bereits von Eley (2012) untersucht. In seiner Simulationsstudie hat er dazu bereits zwei Einlagerungsstrategien und zwei Auslagerungsstrategien miteinander verglichen. Die Einlagerungsstrategie teilt die ankommenden Karosserien auf die Linienpuffer auf, wohingegen die Auslagerungsstrategien eine Karosserie für die Bearbeitung aus einem der Linienpuffer auswählt. Bei der Einlagerungsstrategie stellte sich heraus, dass es am günstigsten ist, die ankommende Karosserie in einen freien Puffer einzulagern, alternativ den auszuwählen, bei dem die geringste Rüstzeit zu der vorherigen Karosserie anfällt. Die zufällige Auswahl eines Linienpuffers führt zu schlechteren Ergebnissen bezüglich der Durchlaufzeit und der benötigten Rüstzeiten.

Weiterhin wurden zwei Auslagerungsstrategien untersucht. Die erste Regel wählt die Karosserie aus, die die geringste Rüstzeit zur zuletzt bearbeiteten Karosserie aufweist. Die zweite Auslagerungsstrategie reduziert das Entscheidungsproblem auf das des Handlungsreisenden (TSP: Travelling Salesman Problem), indem die Rüstzeiten als Entfernungen zwischen den verschiedenen Rüstzuständen interpretiert

werden. Das TSP wird mit einem genetischen Algorithmus (GA) gelöst, der bei jeder Auswahl einer Karosserie aufgerufen wird, also sobald die Lackieranlange wieder zur Verfügung steht.

Die Implementierung dieser Studie hat Eley (2012) in *Tecnomatix Plant Simulation* der Firma Siemens vorgenommen, und auch den dort integrierten Baustein des genetischen Algorithmus verwendet. Es zeigte sich, dass die Einlagerung nach Rüstzeitendifferenz und die Auslagerung mithilfe des genetischen Algorithmus die besten Ergebnisse lieferten. Für diese Studie haben wir die Einstellungen übernommen, die zu besten Ergebnissen geführt haben (Crossover-Parameter = Partially Matched Crossover (PMX); siehe Eley 2012, Experiment 4).

3 Simulationsstudie und untersuchter Ansatz

Ziel dieser Studie ist es einen verbesserten Ansatz zur Reihenfolgeplanung für das hier dargestellte Szenario zu finden. Dazu haben wir ein mathematisches Modell (MILP) entwickelt, das wir in Kombination mit einem Solver als Alternative zum genetischen Algorithmus einsetzen wollen, um so mit optimalen Lösungen des Teilproblems bessere Ergebnisse zu erzielen. Dies erfordert die Kopplung des Simulationsmodells beziehungsweise der Simulationssoftware mit dem Solver, da regelmäßig bei der Fertigstellung eines Auftrags ein neuer Reihenfolgeplan berechnet werden muss. Es gilt zu untersuchen, ob der Solver die auftretenden Instanzen in vertretbarerer Zeit, also innerhalb von Sekunden oder wenigen Minuten, lösen kann.

Da es in unserer Fragestellung weniger auf die visuelle Darstellung der Abläufe ankommt, sondern vielmehr um die Entscheidungsfindung und die Steuerung der Anlage geht, haben wir uns für eine alternative Simulationssoftware mit hoher Geschwindigkeit und geeigneten Schnittstellen entschieden. Die java-basierte Simulationssoftware *jasima*® (JAva SImulator for MAnufacturing and logistics ¡jasima solutions UG) ist effektiv in der Ausführung und bietet eine leichte Integration insbesondere in Java-Projekte. Als Solver wurde *Gurobi* von Gurobi Optimization eingesetzt, der sich durch seine hohe Leistungsstärke auszeichnet und ebenfalls eine Java-Schnittstelle besitzt. So ist es leicht möglich das entwickelte Modell in der Praxis einzusetzen und in bereits bestehende Steuerungssoftware zu integrieren.

Das hier verwendete MILP stellt eine Erweiterung der Modellierung des TSP dar, indem zusätzliche Reihenfolgebedingungen ergänzt wurden, die die verwendeten FIFO Puffer-Bedingungen sicherstellen.

3.1 Analyse des Simulationsmodells und alternative Implementierung

Für das Simulationsmodell werden einige Annahmen getroffen: die Karosserien treffen negativ exponentialverteilt mit einem Erwartungswert von $1/\lambda = 300$ Sekunden ein, der Abstand beträgt dabei mindestens 60 und höchstens 12000 Sekunden. Es werden 20 verschiedene Rüstzustände, hier Farben, angenommen, die gleichverteilt auftreten. Die Rüsttabelle ist in Tabelle 1 dargestellt. Die Bearbeitungszeit auf der Lackierstation beträgt jeweils 5 Minuten. Eley (2012) hat in seiner Untersuchung einen Zeitraum von 8 Stunden betrachtet.

Bei der Implementierung des Szenarios in *jasima* stellte sich heraus, dass die in *Plant Simulation* berechneten Werte beispielsweise für die Durchlaufzeit nicht mit

denen des *jasima* Modells übereinstimmten, wenn man die Simulation über einen längeren Zeitraum laufen ließ. Erste Vermutungen betreffend unterschiedlicher Implementierungsdetails beispielsweise bei den Verteilungen konnten nicht der Grund für die deutlichen Abweichungen sein. Nach genauer Betrachtung stellte sich heraus, dass das System mit den gewählten Parametern überlastet ist und langsam vollläuft. Betrachtet man also unterschiedlich lange Zeiträume erhält man entsprechend unterschiedliche Durchlaufzeiten, da diese bei den gewählten Parametern ansteigen. Bei der gewählten Einlastung ist dies unabhängig von der gewählten Steuerungsmethode.

Nach diesen Ergebnissen wurde die Einlastung der Aufträge verringert, sodass der Erwartungswert der negativen Exponentialverteilung $1/\lambda = 360$ Sekunden entspricht. So wird sichergestellt, dass das System auch mit den einfachen Regeln zur Einbeziehungsweise Auslagerung nicht vollläuft. Diese Erkenntnis unterstreicht weiterhin die Bedeutung der Auswahl eines geeigneten Steuerungsalgorithmus möglichst bereits bei der Dimensionierung von Anlagen, da dieser mitunter nicht nur zu leicht verbesserten Durchlaufzeiten führt, sondern auch andernfalls nötige Kapazitätserweiterungen vermeiden kann.

3.2 Durchführung der Simulationsstudie

Mit beiden Simulationssystemen wurde das beschriebene Szenario implementiert und mit der verringerten Einlastung untersucht. Als Steuerungsverfahren kamen dazu als Auslagerungsstrategie insgesamt drei Verfahren zum Einsatz. In beiden Systemen wurde die einfache Regel implementiert, die den nächsten Auftrag anhand der kürzesten Rüstzeit zum aktuellen Rüstzustand auswählt. In *Plant Simulation* wurde zusätzlich das Verfahren basierend auf dem genetischen Algorithmus eingesetzt. In der *jasima* Implementierung wurde eine Kopplung zu *Gurobi* umgesetzt, der jedes Einzelproblem optimal löst. Aufgrund der regelmäßig nach der ausgewählten Verteilung neu in das System eintreffenden Aufträge kann nicht davon ausgegangen werden, dass eine insgesamt optimale Strategie berechnet wird. Da allerdings aufgrund der FIFO-Linienpuffer Eigenschaften jeweils nur der erste Auftrag ausgewählt werden kann, kann man von sehr guten Ergebnissen ausgehen, denn eingehende Aufträge können in der Regel ohnehin erst später berücksichtigt werden. Weiterhin wurde das *jasima* Modell in einer zweiten Untersuchung auf 10 Linienpuffer erweitert und es wurden ebenfalls beide Steuerungsverfahren miteinander verglichen. Die Simulation wurde für jeweils 1000 Aufträge durchgeführt und bildet damit je nach Verfahren etwas weniger als 5 Tage ab. Um statistisch signifikannte Aussagen treffen zu können, wurden etwa 100 bis 200 einzelne Experimente durchgeführt.

Es hat sich gezeigt, dass sich in der Regel weniger als 20 Aufträge im System befanden und diese Menge in wenigen Sekunden, in Ausnahmen in wenigen Minuten, vom Solver gelöst werden konnte. Damit ist der Einsatz eines Solver in diesem Szenario sehr gut möglich. Sollten schwierige Instanzen auftreten oder deutlich mehr Aufträge im System vorhanden sein, könnte man beispielsweise die Anzahl der berücksichtigen Aufträge reduzieren und nur eine gewisse Anzahl an Aufträgen pro Warteschlange betrachten. Weiterhin könnte man die maximale Zeit, die dem Solver zur Berechnung gegeben wird, begrenzen und sich mit der bis dahin besten Lösung zufrieden geben oder alternativ auf eine Regelauswahl zurückgreifen.

4 Ergebnisse

Die Ergebnisse der Studie zeigen, dass der Genetische Algorithmus genauso wie der Solver im Vergleich zu der einfachen Steuerungsregel signifikant bessere Ergebnisse in Bezug auf die Durchlaufzeit erzielen. Die mittlere Durchlaufzeit beträgt bei den Experimenten die mit *jasima* durchgeführt wurden 2872 Sekunden mit der Regelsteuerung und 2114 Sekunden, wenn der Solver die Reihenfolge berechnet hat. Die Unterschiede sind signifikant, wie der Standardfehler zeigt; ein paarweiser Vergleich (paired t-test) hat dies ebenfalls bestätigt. Die Verbesserung beträgt 26,39 % und liegt damit sehr hoch (Tabelle 2).

Tabelle 2: *Simulationsergebnisse: jasima - Regel und MILP*

Verfahren	Durchlaufzeit	Standardfehler
Regel	2872 s	79,3 s
MILP	2114 s	63,1 s
Differenz	26,4 %	

Der Vergleich zwischen der Entscheidungsregel und dem genetischen Algorithmus zeigt, dass die Regel eine mittlere Durchlaufzeit von 2829 Sekunden liefert, die damit signifikant über der des genetischen Algorithmus mit 2180 Sekunden liegt. Der Unterschied beträgt 22,9 % und ist signifikant (Tabelle 3). Die Ergebnisse bestätigen weiterhin, dass die unterschiedlichen Implementierungen, im Rahmen der auftretenden Schwankungen, die gleichen Ergebnisse liefern.

Tabelle 3: *Simulationsergebnisse: Plant Simulation - Regel und GA*

Verfahren	Durchlaufzeit	Standardfehler
Regel	2829 s	42,8 s
GA	2181 s	24,5 s
Differenz	22,9 %	

Weiterhin ist zwar anzunehmen, dass der Solver mit seinen optimalen Lösungen bessere Ergebnisse liefert als der genetische Algorithmus, da seine Verbesserung gegenüber der Regel etwa 3,5 % höher ausfällt als die des genetischen Algorithmus; allerdings lässt sich dies nicht signifikant zeigen. Der Standardfehler ist dafür zu groß und ein paarweiser Vergleich ist aufgrund der unterschiedlichen Simulationssysteme hier nicht möglich.

In Tabelle 4 sind die Ergebnisse der Simulationsstudie mit 10 statt 5 Linienpuffern aufgeführt, die ähnliche Vorteile für den Solver aufzeigen, allerdings fällt die Verkürzung der mittleren Durchlaufzeit mit 18,3 % kleiner aus. Dies geht einher mit insgesamt kürzeren Durchlaufzeiten (2476 Sekunden und 2023 Sekunden).

Tabelle 4: *Simulationsergebnisse: jasima - Regel und MILP mit 10 Linienpuffern*

Verfahren	Durchlaufzeit]	Standardfehler
Regel	2476 s	47,3 s
MILP	2023 s	25,6 s
Differenz	18,3 %	

Sämtliche Ergebnisse sind in Abbildung 2 mit dem zweifachen Standardfehler grafisch dargestellt.

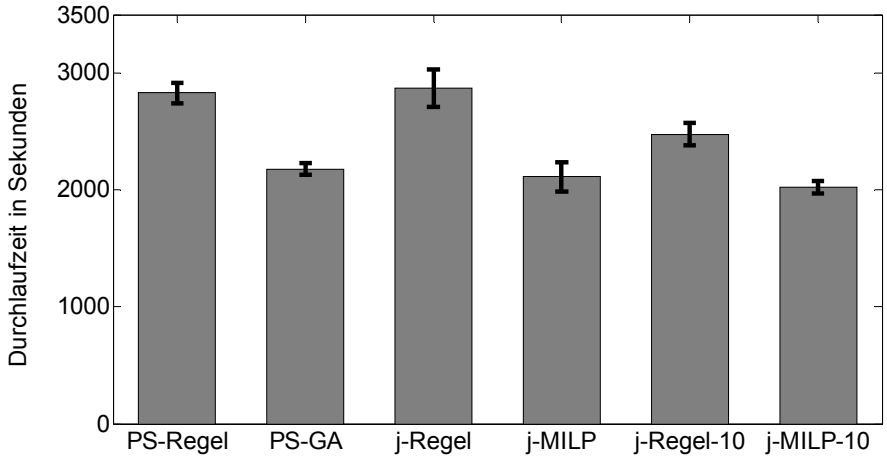

Abbildung 2: *Simulationsergebnisse (mit zweifachem Standardfehler) in der Übersicht*

Ein Geschwindigkeitsvergleich zwischen den beiden Simulationssystemen bietet sich bei dieser Studie nur bedingt an, da die Länge der Simulationsläufe hauptsächlich von den verwendeten Steuerungsverfahren abhängt und deutlich weniger von der Simulationssoftware an sich. Auf einem aktuellen Intel i7 4510 System benötigen die Simulationsläufe mit hunderten Replikationen mehrere Stunden. Prinzipiell liegen die Stärken von *Plant Simulation* sicherlich in der Visualisierung und der relativ einfachen Bedienbarkeit. Bei der Geschwindigkeit und bei der Integration in weitere Systeme spielt *jasima* seine Vorteile aus. Speziell bei der simulationsbasierten Optimierung oder der Integration in Entscheidungsunterstützungssysteme bietet sich daher *jasima* an.

5 Zusammenfassung und Ausblick

In dieser Studie wurden verschiedene Steuerungsverfahren zur Reihenfolgeplanung in einem Szenario mit mehreren parallelen Linienpuffern und reihenfolgeabhängigen Rüstzeiten untersucht. Obwohl es sich dabei um ein NP-vollständiges Problem handelt, sind die auftretenden Problemgrößen in diesem Szenario in vertretbarer Zeit

lösbar. Ein Ausweichen auf eine nichtoptimale Heuristik kann in diesem Fall vermieden werden. Die Kopplung mit einem Simulationssystem bietet dabei vielversprechende Möglichkeiten, beispielsweise die Verknüpfung von dynamischen Abläufen mit (teil)-optimierten Lösungen.

Bei der Durchführung dieser Studie ist weiterhin bestätigt worden, dass es wichtig ist, Annahmen des Simulationsmodels, wie hier die Einlastung der Aufträge, zu überprüfen, um vergleichbare Ergebnisse zu erhalten.

In weiteren Studien wäre es interessant zu untersuchen, ob alternative Einlagerungsstrategien zu weiteren Verbesserungen führen können. Es könnte beispielsweise sinnvoll sein, einen Linienpuffer freizuhalten, um gut passende Aufträge direkt verfügbar zu machen. Dieses Vorgehen könnte je nach Anzahl der Linienpuffer mehr oder weniger vorteilhaft sein. Treten die Aufträge nicht gleichverteilt auf, da einige Farben in diesem Beispiel häufiger nachgefragt werden, könnte dies ebenfalls in der Einlagerungsstrategie berücksichtigt werden.

Eine interessante Fragestellung ist darüber hinaus, wieweit die Steuerungsentscheidungen verbessert werden können, wenn Informationen über zukünftige Aufträge bekannt sind. Gerade im hier betrachteten Beispiel der Automobilproduktion sind die in näherer Zukunft zu erwartenden Karosserien mit geringer Unsicherheit bekannt. Diese Information lässt sich wahrscheinlich für bessere Ergebnisse nutzen, insbesondere für verbesserte Einlagerungsstrategien in die Puffer.

Nicht zuletzt ist ein Vergleich der optimierenden Verfahren (seien es heuristische Optimierungsverfahren wie GAs oder MILPs) mit optimierten Prioritätsregeln sehr interessant (ein aktueller Überblick über deren Erstellung findet sich in (Branke et al. 2015). Diese Regeln konnten manuell entwickelte Regeln in verschiedenen Szenarien zum Teil deutlich schlagen (Hildebrandt et al. 2014). Sie haben gerade unter praktischen Gesichtspunkten den Vorteil, dass sie sich, einmal erstellt, über einen längeren Zeitraum einsetzen lassen und dabei nur sehr geringe Rechenzeitanforderungen stellen. Somit eignen sie sich sehr gut zum Treffen von Planungs- und –Steuerungsentscheidungen in Echtzeit. Für das in diesem Paper betrachtete Szenario ließe sich dieser Ansatz einsetzen, um sowohl optimierte Strategien für die Einlagerung als auch die Auslagerung zu entwickeln.

Literaturverzeichnis

Branke, J.; Nguyen, S.; Pickardt, C.; Zhang, M.: Automated design of production scheduling heuristics: A review. IEEE Transactions on Evolutionary Computation, PP 99 (2015, early access).

Chiang, T.-C.; Fu, L.-C.: Using a family of critical ratio-based approaches to minimize the number of tardy jobs in the job shop with sequence dependent setup times. European Journal of Operational Research 196 (2009) 1, S. 78–92.

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

Heger, J.; Hildebrandt, T.; Scholz-Reiter, B.: Dynamic adjustment of dispatching rule parameters in flow shops with sequence dependent setup times. International Journal of Production Research 2015 (accepted with minor revisions)

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

Hildebrandt, T.; Goswami, D.; Freitag, M.: Large-scale simulation-based optimization of semiconductor dispatching rules. In: Winter Simulation Conference, Savannah 2014, S. 2580-2590.

jasima solutions UG: jasima. Online verfügbar unter http://code.google.com/p/jasima/.

Kochhar, S.; Morris, R. J. T.: Heuristic methods for flexible flow line scheduling. Journal of Manufacturing Systems 6 (1987) 4, S. 299–314.

Kurz, M. E.; Askin, R. G.: Scheduling flexible flow lines with sequence-dependent setup times. European Journal of Operational Research 159 (2004) 1, S. 66–82.

Lee, Y. H.; Pinedo, M.: Scheduling jobs on parallel machines with sequence-dependent setup times. European Journal of Operational Research 100 (1997) 3, S. 464–474.

Monma, C. L.; Potts, C. N.: On the complexity of scheduling with batch setup times. Operations Research 37 (1989) 5, S. 798–804.

Naderi, B.; Salmasi, N.: Permutation flowshops in group scheduling with sequence-dependent setup times. European Journal of Industrial Engineering 6 (2012) 2, S. 177–198.

Pickardt, C. W.; Branke, J.: Setup-oriented dispatching rules - a survey. In: International Journal of Production Research 50 (2011) 20, S. 1–20.

Tan, K. C.; Narasimhan, R.: Minimizing tardiness on a single processor with sequence-dependent setup times: a simulated annealing approach. Omega 25 (1997) 6, S. 619–634.

van der Zee, D.-J.: Non-exhaustive family based dispatching heuristics – exploiting variances of processing and set-up times. International Journal of Production Research 48 (2010) 13, S. 3783–3802.