

Masterarbeit

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Verfahren für die kombinierte Auftragserzeugung,
Auftragsfreigabe und Arbeitsverteilung bei
reihenfolgeabhängigen Rüstzeiten

Vorgelegt von: Philipp Bergthaler, BSc
Studienrichtung: Industrielogistik
Matrikelnummer: 0835315
Universitärer Betreuer: Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Norbert Seiffter



Verfasst am Lehrstuhl für Angewandte Mathematik an der Montanuniversität Leoben.

6. März 2017

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich diese Arbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

AFFIDAVIT

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume.

Leoben, am 6. März 2017

Philipp Bergthaler

Danksagung

Ich bedanke mich herzlich bei allen Personen, die zum Gelingen dieser Arbeit beigetragen haben.

Hervorheben möchte ich meinen Betreuer an der Montanuniversität Leoben, Professor Norbert Seiffter, der diese Arbeit von Beginn an betreut hat und mir nicht nur mit seiner fachlichen Kompetenz zur Seite gestanden ist. Besonders sein ausgesprochenes Vertrauen in meine Fähigkeiten war für die Entstehung dieser Arbeit von großem Wert. Meinen Kontaktpersonen bei dem Unternehmen, in dem das Projekt durchgeführt wurde, danke ich für die rasche und unkomplizierte Bereitstellung von Daten und Informationen.

Ein ganz besonderer Dank gilt meinen Eltern, die mir das Studium finanziell ermöglicht haben und mich bei der Erstellung dieser Arbeit mit großer Ausdauer unterstützten.

Kurzfassung

Die Themen Auftragsproduktion und Auftragsfreigabe treten in jedem produzierenden Betrieb auf. Ihre Lösung unter Miteinbeziehung aller relevanten Parameter ist häufig ein anspruchsvolles Problem. Besonders im Bereich der Serienfertigung sind oft Überlegungen zu rüstopimalen Auftragsreihenfolgen zu berücksichtigen. Wenn ein Auftrag auf mehreren Anlagen produziert werden kann, spielt auch die Arbeitsverteilung eine Rolle. Softwareunterstützung, welche über die Bereitstellung von Informationen hinausgeht, ist für diese Aufgaben kaum vorhanden, da keines der bekannten Verfahren in diesem Bereich andere Ziele als Termintreue und Auslastung berücksichtigt. Daher werden zwei Verfahren entwickelt und beschrieben, die Arbeitsverteilung und reihenfolgeabhängige Rüstzeiten miteinbeziehen. Das erste ist ein multikriterieller genetischer Algorithmus, für den eine Codierung der Individuen sowie die gezielte Einflussnahme auf die Zielgrößen durch neu entwickelte Mutationsoperatoren gezeigt werden. Das zweite Verfahren ist eine einfache Heuristik, die Probleme des genetischen Algorithmus hinsichtlich Laufzeit und Stabilität der Ergebnisse vermeidet. Die beiden Verfahren werden verglichen und es wird diskutiert, für welchen Zweck welches Verfahren besser geeignet ist.

Abstract

Order creation and order release are tasks which have to be attended to in every manufacturing company. Solving them with due regard to all relevant parameters is a difficult problem, in particular if sequence-dependent setup times and assignment to one of multiple available work systems have to be taken into consideration. Software support in this field often does little more than provide the necessary information to an experienced production-planner. This is due to the fact, that none of the known methods in this area take care of other objectives than schedule adherence and workload. Two methods are developed and described, which take into account sequence-dependent setup times and the assignment of orders to work systems. One is a multiobjective genetic algorithm, for which an encoding of individual solutions and several newly developed mutation operators are presented. The other is a simple heuristic, which avoids the genetic algorithm's problems regarding runtime and stability of the solution. Both methods are compared and it is shown which method is better suited in which context.

Inhaltsverzeichnis

Abbildungsverzeichnis	VII
Tabellenverzeichnis	VII
Algorithmenverzeichnis	VII
Abkürzungsverzeichnis	VIII
Glossar	IX
1 Einleitung	1
2 Problemstellung	5
2.1 Die LS-Komponentenfertigung	5
2.2 Die Produktionsplanung	9
2.3 Zielstellung und Potenzial	14
2.4 Anforderungen an das Verfahren	15
3 Beschreibung der Verfahren	18
3.1 Struktur des Softwarepakets	18
3.2 Bedarfsprognose durch Künstliche Neuronale Netze	21
3.3 Auftragserzeugung und Auftragsfreigabe durch Genetischen Algorithmus	27
3.4 Auftragserzeugung und Auftragsfreigabe durch Kapazitäts- und Rüstori-	
entzierte Auftragsfreigabe (KRAF)	47
3.5 Reihenfolgeoptimierung	59
4 Präsentation und Diskussion der Ergebnisse	62
4.1 Qualitative Lösungsbewertung durch den Produktionsplaner	63
4.2 Konzeptueller Vergleich zwischen Genetischer Algorithmus (GA) und KRAF	
hinsichtlich der Anwendbarkeit	66
4.3 Veränderungen im Laufzeitverhalten des GA durch gezielte Mutation und	
lokale Suche	69
5 Fazit und Ausblick	73
Literatur	76
A Benötigte Daten	79

Abbildungsverzeichnis

1	Schematischer Ablaufplan der Software	18
2	Schematische Darstellung eines künstlichen Neurons	21
3	mehrschichtige Perzeptron (MLP) mit einer verborgenen Schicht	22
4	Gezielte Mutation: Hinzufügen von Aufträgen für Schnelldreher	71
5	Gezielte Mutation: Reduktion der Rüstkosten	71
6	Gezielte Mutation: Fehlmengenreduktion durch Vorziehen von Aufträgen	72
7	Lokale Suche durch Optimierung von Einzelmaschinen	72

Tabellenverzeichnis

1	Beispiel Bestandsaufbau	31
2	Bewertung der Einzelziele	32
3	Beispiel 2 Zielwerte	34
4	Gewichtungsfaktoren	36
5	Ergebnisse Planungsbewertung	64
6	Ergebnisse Planungsbewertung (bereinigt)	65

Algorithmenverzeichnis

1	Mutation Maschinenzuordnung	38
2	Mutation Produktionsmenge	39
3	Mutation Produktionsintervall	39
4	Gezielte Mutation - Fehlmengenreduktion durch zusätzliche Fertigungsaufträge	40
5	Gezielte Mutation - Hinzufügen von Aufträgen für Schnelldreher	41
6	Gezielte Mutation - Fehlmengenreduktion durch Vorziehen von Fertigungsaufträgen	42
7	Gezielte Mutation - Rüstzeitreduktion	43
8	Kombination von Ergebnissen der lokalen Suche	46
9	Auftragserzeugung für KRAF	48
10	Vortermminierung der Aufträge für Bestandsaufbau	50
11	Maschinenzuordnung fälliger Fertigungsaufträge	54
12	Verbesserung der Arbeitsverteilung	55
13	Versuche Verschiebung auf andere Maschine	56
14	Kapazitätsauffüllung unter Berücksichtigung reihenfolgeabhängiger Rüstzeiten	58
15	Lösungsauswahl KO-Kriterium Verspätung	60

Akronyme

EDD	Early Due Date
ERP	Enterprise Resource Planning
FI	Fehlerstrom-Schutzschalter
GA	Genetischer Algorithmus
KNN	Künstliches Neuronales Netz
Korma	Kapazitätsorientierte Materialbewirtschaftung
KRAF	Kapazitäts- und Rüstorientierte Auftragsfreigabe
KW	Klemmenwinkel
LS	Leitungsschutzschalter
MLP	mehrschichtige Perzeptron
MRP	Materials Requirement Planning
MRP II	Manufacturing Resource Planning
MTO	Make to Order
MTS	Make to Stock
OEE	Gesamtanlageneffektivität
PMA	Permanentmagnet-Auslöser
PPS	Produktionsplanung und -steuerung
TPX	Two-Point-Crossover

Glossar

Bestellanforderung

Bestellungen von Kunden, deren Lieferung vom Produktionswerk noch nicht bestätigt wurde.

Fertigungsauftrag

Ein Element der Fertigungssteuerung. Wird auch als Produktionsauftrag bezeichnet und ist ein Auftrag an die Fertigung eine gewisse Menge eines bestimmten Produkts auf einer definierten Maschine zu produzieren.

Langsamdreher

Artikel mit niedrigem Lagerumschlag pro Periode [29]

Materialreservierung für Fertigungsauftrag

Für jeden Fertigungsauftrag berechnet das System im Zuge eines MRP-Laufs den Sekundärbedarf. Dieser scheint im System als Reservierung auf.

NB-Datum

Nullbestands-Datum; Das Datum zu dem ein Fertigungsauftrag fertiggestellt sein muss, damit der Bestand des entsprechende Artikel nicht auf 0 sinkt.

Planauftrag

Vom Enterprise Resource Planning (ERP)-System vorgeschlagene Fertigungsaufträge, welche noch nicht vom Disponenten freigegeben wurden.

Produktionszyklus

Der Produktionszyklus für einen Artikel a ist der über einen längeren Zeitraum gebildete Mittelwert der Zeitspanne zwischen zwei aufeinanderfolgenden Zeitpunkten, zu denen die Rüstnachbarschaft von a betreten wird, d.h. Zeitpunkten zu denen auf einen Artikel $r \in R_{a,m}$ gerüstet wird, wenn zuvor ein Artikel $p \notin R_{a,m}$ produziert wurde. Für Artikel welche auf mehreren Maschinen hergestellt werden können ergeben sich daraus theoretisch verschiedene Produktionszyklen. In der Praxis ist aber meist nur interessant, wann ein Artikel das nächste Mal mit geringem Rüstaufwand produziert werden kann, daher wird nur der kürzeste Produktionszyklus berücksichtigt.

Rüstfall

Die Umrüstung einer Maschine zwischen zwei verschiedenen Artikeln und alle dafür notwendigen Arbeiten.

Rüstmachbarschaft

Die Rüstmachbarschaft $R_{a,m}$ von Artikel a auf Maschine m besteht aus jener Teilmenge der auf m herstellbaren Artikel A_m für die gilt, dass der Aufwand für die Umrüstung von $r \in R_{a,m}$ auf a relativ niedrig ist. Meist ist diese Beziehung wechselseitig, wenn $x \in R_{y,m}$ gilt, dann gilt auch $y \in R_{x,m}$.

SB-Datum

Sicherheitsbestands-Datum; Das Datum zu dem ein Fertigungsauftrag fertiggestellt sein muss, damit der Bestand des entsprechende Artikel nicht den Sicherheitsbestand unterschreitet

Schnelldreher

Artikel mit hohem Lagerumschlag pro Periode [29]

SIOP

Sales Inventory and Operations Planning ist ein Management-Prozess zum Abgleich von Bedarfs- und Produktionsmengen unter Berücksichtigung von Lagerständen.

Werksverbund

Als Werksverbund werden die Montagewerke bezeichnet in welchen der Zusammenbau der Schalter erfolgt. Diese Werke sind mit dem Produktionswerk direkt über das ERP-System verbunden.

1 Einleitung

Die wirtschaftlichen Rahmenbedingungen, unter denen produzierende Betriebe tätig sind, haben sich in den letzten Jahrzehnten grundlegend gewandelt. Der durch die Globalisierung verstärkte Wettbewerb führte zu einem Übergang der Verkäufermärkte in Käufermärkte, in denen Kundenwunsch und Kundennutzen im Zentrum stehen [27]. Daraus folgt, dass die alleinige Fokussierung auf geringe Kosten durch hohe Auslastung für den wirtschaftlichen Erfolg nicht mehr genügt. Viele Unternehmen setzen auf einen Wettbewerbsvorteil durch hohe Produktqualität, doch die Möglichkeit dazu hängt vom Produkt ab und so gibt es heute nur noch sehr wenige und spezialisierte Branchen, in denen die Produktqualität alleine ausreicht, um sich gegenüber den Mitbewerbern abzugrenzen. Verstärkt wird heute versucht, den Kundennutzen durch zusätzliche Serviceleistungen zu erhöhen. Insbesondere logistische Leistungen stehen im Fokus, da die Erwartungen der Kunden in diesem Bereich stark gestiegen sind. Die Forderung nach kürzeren Lieferzeiten, größerer Lieferflexibilität und Termintreue, bei gleichzeitig steigender Variantenvielfalt stellen hohe Ansprüche an die Produktionslogistik. Der Grund dafür ist der Kostendruck und die damit einhergehenden Forderung, Kapitalbindungskosten zu reduzieren. Deshalb kann die hohe logistische Zielerreichung nicht durch einen vergrößerten Bestand an Enderzeugnissen erreicht werden [11]. Um unter dieser Restriktion den genannten Anforderungen gerecht zu werden, ist es notwendig, den Ausstoß der Produktion möglichst genau mit dem aktuellen Bedarf in Deckung zu bringen. Das Potenzial dafür muss in so unterschiedlichen Unternehmensfunktionen wie der Fabrikplanung, Produktentwicklung, Technologieentwicklung, Marketing und Vertrieb oder der strategischen Produktionsplanung geschaffen werden, die Umsetzung des Potenzials erfolgt durch die operative Produktionsplanung und -steuerung (PPS). Eine herausragende logistische Zielerreichung zur Erzielung eines Wettbewerbsvorteils ist daher nur durch die Koordination aller betroffenen Bereiche möglich [18].

In dieser Arbeit werden zwei Verfahren zur kombinierten, zentral durchgeführten Auftragsherzeugung, Auftragsfreigabe und Arbeitsverteilung unter Berücksichtigung reihenfolgeabhängiger Rüstzeiten beschrieben. Diese Aufgaben fallen in das Gebiet der operativen PPS und werden nachfolgend beschrieben.

Auftragserzeugung

Die Auftragserzeugung verarbeitet Bedarfsdaten, wie Kundenaufträge, Materialentnahmen oder ein Produktionsprogramm und generiert daraus Fertigungsaufträge. Nach dem Modell der Fertigungssteuerung von Lödding [18] ist sie damit für die Festlegung des Plan-Zugangs und des Plan-Abgangs der Fertigung sowie der Plan-Reihenfolge zuständig. Ein vollständiges und realisierbares Ergebnis ist eine Voraussetzung für hohe Termintreue.

Auftragsfreigabe

Die Auftragsfreigabe legt fest, ab welchem Zeitpunkt ein Fertigungsauftrag in der Produktion bearbeitet werden darf und bestimmt damit den Ist-Zugang zur Fertigung. Daher hat die Auftragsfreigabe großen Einfluss auf die Höhe des Fertigungsbestandes. Den Grundgesetzen der Produktionslogistik zufolge beeinflusst die Bestandshöhe die Durchlaufzeit, die Streuung der Durchlaufzeit und damit die logistische Prozesssicherheit negativ [20]. Hohe Termintreue kann ohne logistische Prozesssicherheit nicht erreicht werden. Daraus folgt, dass sowohl Auftragserzeugung als auch Auftragsfreigabe großen Einfluss auf die logistische Zielerreichung haben.

Arbeitsverteilung

Die Funktion der Arbeitsverteilung ist die Zuteilung eines Auftrages zu einem Arbeitssystem, falls für die Bearbeitung verschiedene Arbeitssysteme zur Verfügung stehen. Sie kann als eigenständige Aufgabe oder als Teil der Auftragserzeugung und Auftragsfreigabe begriffen werden.

Rüsto Optimierung

Rüsto Optimierung ist keine Aufgabe, sondern ein Ziel, auf dessen Erreichung die drei zuvor genannten Aufgaben einwirken. Die Berücksichtigung reihenfolgeabhängiger Rüstzeiten bedeutet in diesem Kontext, dass es sinnvoll sein kann, Fertigungsaufträge vorzeitig zu erzeugen und freizugeben. Dieser Fall tritt ein, wenn vorzuziehende Aufträge, in Hinblick auf die Rüstzeiten, gut zu den aktuell freigegebenen Fertigungsaufträgen passen und die vorgezogene Produktion daher eine Reduktion der Gesamtrüstzeit bedeutet. Um die Termintreue nicht zu gefährden, dürfen Aufträge nur unter Berücksichtigung der Kapazitätsauslastung der infrage kommenden Arbeitssysteme vorgezogen werden.

Dem weitverbreiteten Aachener PPS-Modell [26] zufolge, ist die Auftragserzeugung eine Aufgabe der Produktionsplanung, während die Auftragsfreigabe in den Bereich der Produktionssteuerung fällt. Lödding [18] differenziert zwischen Produktionssteuerung und Fertigungssteuerung insofern, dass die Produktionssteuerung auch für die Steuerung des Fremdbezuges verantwortlich ist, während die Fertigungssteuerung nur die Eigenfertigung betrachtet. Da die in dieser Arbeit beschriebenen Verfahren für die Eigenproduktion entwickelt wurden, wird diese Terminologie übernommen. Aus dem gleichen Grund werden auch die Begriffe Auftragserzeugung und die Auftragsfreigabe im Rahmen der Eigenfertigung gesehen, obwohl die gleichen Aufgaben auch in der Fremdbeschaffung anfallen. Weiter stellt Lödding fest, dass die exakte Verteilung der Aufgaben zwischen Produktionsplanung und Fertigung in der Praxis nicht immer gegeben ist. Er beschreibt, dass die Fertigungssteuerung teilweise die Auftragserzeugung, bspw. nach dem Kanban-Prinzip, übernimmt. In dem Fall, welcher dieser Arbeit zugrunde liegt, ist es umgekehrt, die Produktionsplanung übernimmt die Auftragsfreigabe. Es ist anzunehmen, dass, besonders bei kurzen Lieferfristen, diese Aufgaben häufig kombiniert werden, da ihre Trennung Mehrarbeit ohne Mehrwert bedeuten würde.

Ein weiterer Vorteil der Kombination von Auftragserzeugung und Auftragsfreigabe kann sich aus der Verknüpfung der Verfahren ergeben. Die meisten Verfahren zur Auftragserzeugung führen keinen Kapazitätsabgleich durch. Daraus können nicht realisierbare Produktionspläne resultieren. Durch Kombination mit einem Freigabeverfahren, welches das Kapazitätsangebot berücksichtigt, kann diesem Problem vorgebeugt werden. Es ist festzustellen, dass im Bereich der Auftragserzeugung und Auftragsfreigabe nur eine überschaubare Anzahl von bekannten Verfahren existiert. Während beispielsweise für die Reihenfolgeoptimierung unzählige Methoden des Operations Research, ausdifferenziert nach Zielvorgaben und Randbedingungen, publiziert wurden, beschreibt Lödding in seinem Übersichtswerk lediglich acht Verfahren zur Auftragsfreigabe [18]. Zusätzlich ist noch das Manufacturing Resource Planning (MRP II)-Verfahren zu nennen, aber es muss auch angemerkt werden, dass sich manche der Verfahren überschneiden und in Teilen große Ähnlichkeiten aufweisen. Anscheinend wird der Reihung von Fertigungsaufträgen ein größeres Interesse gewidmet, als der Frage danach, welche Fertigungsaufträge überhaupt gereicht werden müssen. Von den bekannten Steuerungsverfahren kommen

Bestellbestandsverfahren [18], Kapazitätsorientierte Materialbewirtschaftung (Korma) [25] und MRP II [1] für die zentrale Auftragssteuerung in Frage, während Kanban [33], Synchro MRP [12], Hybride Kanban-Conwip-Steuerung [4] und Basestock [10] dezentrale Steuerungskonzepte darstellen. Die Fortschrittszahlensteuerung [30] und Production Authorization Cards [5] können als generische Verfahren für verschiedene Zwecke konfiguriert werden. In der Literatur wird jedoch nur beschrieben, wie sie so ausgestaltet werden können, dass sie die bereits genannten Verfahren darstellen. Keine bekannte Methodik zur Auftragserzeugung berücksichtigt reihenfolgeabhängige Rüstzeiten oder die Arbeitsverteilung. Bei Korma und, je nach Ausgestaltung, bei MRP II wird aber das Vorziehen von Aufträgen zur Kapazitätsauslastung berücksichtigt.

Zusammengefasst decken die beschriebenen Verfahren nicht alle Vorgaben ab, die in einer Fertigung auftreten können. Die vorliegende Arbeit entstand im Rahmen eines Projekts, bei dem ein kombiniertes Verfahren für die Auftragserzeugung, Auftragsfreigabe und Arbeitsverteilung unter Berücksichtigung reihenfolgeabhängiger Rüstzeiten entwickelt wurde. Als Lösungsansätze wurden ein GA und eine Heuristik, welche Anleihen bei Korma nimmt, gewählt. In dieser Arbeit soll die Frage beantwortet werden, ob diese Ansätze für die Lösung des Problems geeignet sind. Dabei wird sowohl die Fähigkeit gute Lösungen zu finden, als auch die praktische Anwendbarkeit überprüft. Dazu wurden der GA und die Heuristik, sowie notwendige vorgelagerte Verfahren zur automatisierten Aufbereitung der Eingangsdaten, in Java implementiert und so ein Softwarepaket für die Bearbeitung des Problems geschaffen. Der Aufbau dieser Arbeit orientiert sich am Entstehungsprozess dieses Programms. In Kapitel 2 werden zuerst die Fertigung und die Produktionsplanung beschrieben, für welche die Software entwickelt wurde. Daraus werden anschließend die Anforderungen an das Verfahren abgeleitet. Kapitel 3 beginnt mit einer Beschreibung des grundsätzlichen Aufbaus der Software. Da die Verfahren auf Daten über die zukünftigen Bedarfe basieren, diese aber nur für einen sehr kurzen Zeithorizont verlässlich verfügbar sind, wurde eine Bedarfsprognose durch neuronale Netze durchgeführt. Die angewendeten Techniken werden im Abschnitt 3.2 kurz zusammengefasst dargestellt. Eine ausführliche Darstellung der Gestaltung des GA (3.3) und der Heuristik (3.4) folgen. In Kapitel 4 werden die Ergebnisse präsentiert und analysiert. Eine Zusammenfassung der wichtigsten Resultate sowie der Ausblick auf weitere Betätigungsmöglichkeiten in diesem Forschungsfeld bilden in Kapitel 5 den Abschluss.

2 Problemstellung

Das Projekt, in dessen Rahmen diese Arbeit entstand, wurde in Kooperation mit einem global agierenden Unternehmen aus dem Sektor der Energietechnik durchgeführt. Gefertigt werden hauptsächlich elektrotechnische Komponenten für die Herstellung von Schutzschaltgeräten. Betrachtet werden in dieser Arbeit die Komponenten für Artikel aus der Produktgruppe Leitungsschutzschalter (LS). Diese werden in einem räumlich und organisatorisch zusammengefassten Bereich der Fertigung hergestellt, der als LS-Komponentenfertigung bezeichnet wird. Ziel des Projektes war es, eine Software zur Unterstützung der Produktionsplanung für diese Fertigung zu erstellen. Für das bessere Verständnis der Problemstellung werden in diesem Abschnitt zunächst die LS-Komponentenfertigung sowie der Produktionsplanungsprozess für diese Abteilung beschrieben. Anschließend werden aus diesen Rahmenbedingungen die Anforderungen an das Verfahren abgeleitet. Um den Geheimhaltungsrichtlinien des Unternehmens zu entsprechen, sind einige unternehmensspezifische Daten in diesem Kapitel leicht verändert. Die Größenordnung wurde jedoch beibehalten, sodass ohne weiteres Rückschlüsse auf die Anwendbarkeit der präsentierten Verfahren getroffen werden können.

2.1 Die LS-Komponentenfertigung

Die LS-Komponentenfertigung produziert die beiden zentralen Baugruppen für verschiedene Modelle von LS. Nachfolgend werden die charakteristischen Eigenschaften der Fertigung angeführt, sowie die Produkte und die für ihre Herstellung eingesetzten Maschinen beschrieben.

Fertigungsprinzip

Die Fertigung am Standort ist nach dem Werkstattprinzip organisiert [1]. Für die LS-Komponentenfertigung von Bedeutung sind Stanzerei und Kunststofffertigung, da diese benötigte Vorprodukte und weitere Komponenten für LS herstellen. Mit den weiteren Werkstätten gibt es wenige direkte Berührungspunkte.

Fertigungsart

Die LS-Komponentenfertigung ist eine klassische Serienfertigung.

Materialflusskomplexität

Da die einzelnen Werkstätten durch Zwischenlager sehr stark entkoppelt sind ist die Materialflusskomplexität aus Sicht der LS-Komponentenfertigung sehr niedrig. In Normalfall kann davon ausgegangen werden, dass die benötigten Vorprodukte vorhanden sind, die Produkte werden nur auf einer Maschine bearbeitet und danach eingelagert. Es handelt sich daher um mehrere parallele, einstufige, lineare Materialflüsse.

Teilefluss

Grundsätzlich wird immer ein ganzes Fertigungslos auf einer Maschine produziert und danach ins Lager gebracht. Bei sehr großen Produktionslosen werden allerdings manchmal auch Teilmengen eingelagert, um Platz zu schaffen.

Variantenanzahl

In der LS-Komponentenfertigung werden die Baugruppen Joch und Mechanismus hergestellt. Eine Beschreibung dieser Produktgruppen folgt im Abschnitt 2.1.1. Derzeit existieren etwa 70 Varianten der Baugruppe Mechanismus und etwa 260 Varianten der Baugruppe Joch. Die Variantenanzahl liegt damit in einem mittleren Bereich.

2.1.1 Die Produkte

Leitungsschutzschalter werden eingesetzt, um Leitungen vor Beschädigung durch Überströme zu schützen. Tritt ein Überstrom auf, wird der Stromkreis vom LS unterbrochen. Es sind zwei Arten von Überstrom zu berücksichtigen [15] [3].

Überlaststrom

Tritt im fehlerfreien Betrieb auf, beispielsweise wenn eine zu hohe Belastung eines Motors, oder der gleichzeitige Betrieb mehrerer Verbraucher zu einer Überlastung der Leitung führen. Überlaststrom ist höher als der Bemessungsstrom, auf den die Leitung ausgelegt ist, überschreitet diesen jedoch oft nur geringfügig.

Kurzschlussstrom

Wird durch einen Kurzschluss im elektrischen Netz hervorgerufen. Die Stromstöße steigen sehr schnell an und sind deutlich stärker als der Bemessungsstrom.

Verschiedene Ausführungen von LS unterscheiden sich durch die Höhe des Bemessungsstroms und das Ausschaltverhalten. Welche Ausführung benötigt wird, hängt vom Einsatzzweck ab. Kleine Unterschiede zwischen den Ausführungen werden durch Veränderungen der Einstellungen am LS realisiert, größere durch den Einsatz von Komponenten, welche sich durch das Material oder die Bauteilstärke unterscheiden.

Die Komponenten für LS werden am Standort hergestellt und dann für den Zusammenbau in die Montagewerke geliefert. Diese Montagewerke werden als Werksverbund bezeichnet. Nur die, maßgeblich für die Eigenschaften des LS verantwortlichen Baugruppen, Joch und Mechanismus werden in der LS-Komponentenfertigung produziert. Die weiteren Komponenten, mit deutlich geringerer Variantenanzahl, werden in Stanzerei und Kunststofffertigung hergestellt. Diese werden teilweise auch in anderen Enderzeugnissen, wie Fehlerstrom-Schutzschalter (FI) oder Permanentmagnet-Auslöser (PMA), verbaut.

Joch

Das auffälligste Bauteil eines Jochs ist eine Spule, die ein Magnetfeld induziert. Bei ausreichend starken Strömen, insbesondere bei Kurzschlussströmen, wird dieses Magnetfeld so stark, dass der sogenannte Stößel aus seiner Ausgangsposition bewegt wird und eine Abschaltung des Stromkreises verursacht.

Mechanismus

In dieser Baugruppe ist ein Bimetallstreifen verbaut, der sich durch den Stromfluss erwärmt und dadurch krümmt. Wird der Bemessungsstrom für eine ausreichend lange Zeitspanne überschritten, erfolgt eine Unterbrechung des Stromkreises. Je größer die Überschreitung des Bemessungsstromes, umso schneller erfolgt die Erwärmung und damit die Abschaltung.

Klemmenwinkel (KW)

Verbindung des Schalters zum Stromkreis. LS für hohe Bemessungsströme benötigen KW aus Kupfer, sonst genügt Eisen. KW werden in der Stanzerei gefertigt und als Vorprodukt an die LS-Komponentenfertigung geliefert, wo sie mit den Jochen und Mechanismen verbunden werden.

Deionkammer

Bei der Unterbrechung des Stromkreises, besonders bei Kurzschlüssen, tritt ein Schaltlichtbogen auf, welcher in der Deionkammer (vgl. Deionisations-Kammer, Lichtbogenlöschkammer) durch Deionisierung gelöscht wird. Diese Komponente wird in der Stanzerei produziert.

Schalen

Schaltergehäuse aus Kunststoff, welches in der Kunststofffertigung hergestellt wird.

Zeiger

Dient der Anzeige ob der Schalter gespannt ist und stammt ebenfalls aus der Kunststofffertigung.

2.1.2 Die Maschinen

Zum Zeitpunkt des Entstehens der vorliegenden Arbeit besteht der Anlagenpark der LS-Komponentenfertigung aus zehn Maschinen. Sechs davon dienen der Produktion von Mechanismusbaugruppen, die anderen vier der von Jochbaugruppen. Die zweite wichtige Unterscheidung zwischen den Maschinen erfolgt anhand der eingesetzten Technologie. Es werden drei verschiedene Maschinentypen eingesetzt, diese werden als Tech 1, Tech 2 und Tech 3 bezeichnet. Es folgt eine kurze Erläuterung dieser Maschinentypen, sowie die Angabe, wie viele davon Mechanismen bzw. Joche herstellen.

Tech 1

4 x Mechanismus / 0 x Joch; Die Baugruppen werden auf diesen Maschinen aus vorgefertigten Einzelteilen zusammengebaut. Diese Einzelteile stammen aus der Stanzerei. Die Stückzahl, welche auf diesen Maschinen pro Schicht durchschnittlich gefertigt werden kann, ist geringer als bei den anderen Maschinentypen.

Tech 2

1 x Mechanismus / 2 x Joch; Im Unterschied zu den Tech 1-Maschinen können diese Anlagen einige Einzelteile direkt aus Draht beziehungsweise Blech herstellen. Weitere benötigte Teile stammen aus der Stanzerei. Die Stückzahl pro Schicht liegt zwischen denen von Tech 1 und Tech 3.

Tech 3

1 x Mechanismus / 2 x Joch; Auf diesen Anlagen werden alle benötigten Teile aus Draht oder Blech hergestellt, eine Vorfertigung von Einzelkomponenten ist daher nicht notwendig. Auf den Tech 3-Maschinen wird der höchste Ausstoß pro Schicht erreicht.

Die meisten der ca. 330 Artikel sind eindeutig einem Maschinentyp zugewiesen, 60 können jedoch mit verschiedenen Maschinentypen hergestellt werden. Diese Varianten werden als Fertigungsversionen bezeichnet. Durch die Taktzeit und weil, je nach Maschinentyp, andere Vorleistungen notwendig sind, ergeben sich Unterschiede zwischen den Produktionskosten der Fertigungsversionen. Am teuersten ist die Fertigung mit Tech 1, danach folgt Tech 2 und am günstigsten ist die Fertigung mit Tech 3. Bei Mechanismen, die auf Tech 1-Maschinen produziert werden, ist eine Auswahl zwischen den Anlagen dieses Typs zu treffen. Wegen technologischer Einschränkungen kann jedoch nicht jeder Artikel auf jeder Tech 1-Maschine gefertigt werden. Das gleiche gilt für die Produktion von Jochen auf den Tech 2-Maschinen. Zwischen verschiedenen Anlagen des gleichen Maschinentyps besteht kein Unterschied in den Produktionskosten eines Artikels.

2.2 Die Produktionsplanung

Die Produktionsplanung für die verschiedenen Fertigungsabteilungen erfolgt zentral. Durch die Zusammenführung der Planungsarbeiten in einer Abteilung ist eine bessere Abstimmung der Fertigungen, welche teilweise in einer Lieferant - Kunde Beziehung stehen, möglich. Jeder Planer ist für einen funktional zusammengehörigen Bereich der Fertigung verantwortlich. Nachfolgend werden zuerst jene Eigenschaften der Produktgruppe LS angeführt, welche die Produktionsplanung für diesen Bereich prägen, anschließend wird der Produktionsplanungsprozess beschrieben.

Auslösung der Fertigungsaufträge

Etwa 80% der Artikel werden auf Lager produziert, 20% nur auf Kundenauftrag. Es handelt sich daher um eine Mischfertigung. Der Anteil, der auf Kundenauftrag produzierten Artikel, am Produktionsvolumen beträgt jedoch weniger als 1%.

Lieferzeit

Bei den meisten Artikeln gilt für den Werksverbund eine Lieferzeit von drei Tagen. Kurzfristigere Bestellungen werden nur in Ausnahmefällen akzeptiert. Das Gros der Bestellungen des Werksverbundes trifft zwischen fünf und drei Tagen vor dem gewünschten Auslieferungsdatum ein. Diese kurze Lieferzeit wird durch Lagerhaltung der Enderzeugnisse erreicht. Für externe Kunden gelten längere Lieferzeiten. Bestellungen von diesen können daher zur Bedarfsglättung herangezogen werden.

Schwankung der Nachfrage

Die nachgefragten Wochenmengen je Artikel sind starken Schwankungen unterworfen. Durch die flexible Produktion, in der jede Anlage verschiedene Produkte fertigen kann, können viele dieser Ausschläge ausgeglichen werden. Allerdings unterliegt auch die Gesamtbedarfsmenge Schwankungen, welche berücksichtigt werden müssen. Im Rahmen der Kapazitätsbedarfsplanung wird das Kapazitätsangebot an den prognostizierten Kapazitätsbedarf angepasst. Die Unterschreitung der Kapazitätsbedarfsprognose wird bis zu einem gewissen Grad durch verstärkte Lagerproduktion von Schnelldrehern ausgeglichen. Dadurch kann auch ein Überschreiten der Prognose teilweise ausgeglichen werden.

Flexibilität

Um den Lagerbestand an Enderzeugnissen gering zu halten, wird versucht, die Produktionsmengen auf den tatsächlichen Bedarf abzustimmen. Die benötigte Flexibilität wird durch die Entkoppelung der Fertigung von den internen und externen Lieferanten, mittels Halbfabrikatlager, erreicht. Im Normalfall wird die Flexibilität der Produktionsplanung nicht durch einen Mangel an Halbfertigprodukten eingeschränkt.

Kundenstruktur und ungefähre Verteilung der Produktionsmenge

- 70% werden an den Werksverbund geliefert, in denen die Endfertigung der Schutzschaltgeräte erfolgt.
- 30% gehen an verschiedene externe Kunden.

Auftragserzeugung und Auftragsfreigabe

Diese Aufgaben werden für die LS-Komponentenfertigung zusammengefasst vom Produktionsplaner übernommen. Dadurch stehen mehr Informationen zur Verfügung, deren Berücksichtigung grundsätzlich die Qualität der zu treffenden Entscheidungen positiv beeinflussen kann [18]. Die wichtigsten Entscheidungen, die getroffen werden müssen sind:

- Welche Artikel müssen aufgrund der Bedarfssituation produziert werden?
- Auf welchen Maschinen werden die Artikel gefertigt?
- Die Produktion welcher Artikel soll aufgrund der absehbaren Bedarfsentwicklung vorgezogen werden, um eine Verbesserung der Rüstreihenfolge zu erreichen?
- Sollen Schnelldreher, in größerer Menge, als zur Deckung des aktuellen Bedarfs, produziert werden, um in der nahen Zukunft mehr Kapazität für die Produktion von Artikeln mit unregelmäßigem Bedarfsverlauf zur Verfügung zu haben?
- Wie soll die Produktion für große Bestellungen externer Kunden, die eine längere Lieferzeit haben, verteilt werden um, Kapazitätsschwankungen auszugleichen und gleichzeitig die Termintreue sicherzustellen?

Der Produktionsplanungsprozess

Alle 4 Wochen wird, im Rahmen eines SIOP-Prozesses, jeweils für die nächsten 18 Monate, ein Supply-Plan erstellt, der prognostizierte Bedarfe, Bestandsentwicklungen in den Distributionszentren und vorhandene Produktionskapazitäten berücksichtigt. Der SIOP-Prozess wird auf Ebene der Produktgruppen Joch und Mechanismus durchgeführt, daher können daraus keine Informationen über den Bedarf auf Artikelebenen gewonnen werden. Die Mengen, die im Supply-Plan für das nächste Monat festgelegt werden, gelten als fixiert. Auf Basis dieser Informationen wird unter anderem der Schichtplan erstellt.

Kennzeichnend für die operative Produktionsplanung der LS-Komponentenfertigung ist der knappe Planungshorizont aufgrund der kurzen vereinbarten Lieferzeiten. Um den Ausstoß der Produktion qualitativ und quantitativ möglichst genau an die aktuelle Bedarfssituation anzupassen, wird der Prozess, welcher zu Auftragserzeugung und Auf-

tragsfreigabe führt, täglich zwei- bis dreimal durchgeführt. Der Disponent prüft grob, ob die vom Werksverbund als Bestellanforderung eingetroffenen Bestellungen erfüllt werden können, ändert, falls nötig, das Lieferdatum oder die Menge und gibt sie dann frei. Durch diese Freigabe wird aus der Bestellanforderung eine Bestellung. Damit wird erstens dem Kunden signalisiert, dass er die bestätigte Menge zum bestätigten Datum erhalten wird und zweitens die Bestellmenge bestandswirksam. Das bedeutet, dass im Zuge des Materials Requirement Planning (MRP)-Laufs die Bestellmenge zum Bestelltermin vom verfügbaren Bestand abgezogen wird. Zusätzlich zu den Bestellungen, welche durch Bestätigung von Bestellanforderungen des Werksverbundes entstehen, werden Bestellungen von externen Kunden durch den Vertrieb direkt in das System eingetragen.

Bei Make to Order (MTO)-Artikeln wird für jede Bestellung automatisch ein Fertigungsauftrag über die gleiche Menge und mit einer eingestellten Vorlaufzeit erstellt. Komplizierter ist die Situation bei Make to Stock (MTS)-Artikeln. Für diese berechnet das System auf Basis aller Bestellungen, bereits freigegebener Fertigungsaufträge, des aktuellen Lagerbestandes und des gewählten Dispositionsverfahrens [32], im Rahmen eines MRP-Laufs, zu welchem Zeitpunkt und welche Menge des jeweiligen Artikels produziert werden soll. Diese Daten werden als Planaufträge im System gespeichert. Auf dieser Basis führt der Disponent die Auftragserzeugung und Auftragsfreigabe durch, indem er entscheidet, welche Planaufträge in Fertigungsaufträge umgewandelt werden. Dazu werden in erster Linie jene Planaufträge herangezogen, welche in naher Zukunft produziert werden müssen, um alle Bedarfe bedienen zu können. Da die Terminierung der Planaufträge keinerlei Rücksicht auf Rüstreihenfolgen oder Kapazitätsauslastung nimmt, besteht die Aufgabe des Disponenten darin, Planaufträge, deren Produktion das System erst in mittlerer Zukunft vorschlägt, vorzuziehen, wenn diese gut in die Rüstreihenfolge passen. Über einen längeren Zeitraum hinweg wird so die Rüstzeit reduziert und damit die Gesamtanlageneffektivität (OEE) gesteigert. Da bei den meisten Artikeln die vereinbarte Lieferzeit deutlich kürzer ist, als der Produktionszyklus, ist es, vor allem bei Langsamdrehern, für den Disponenten regelmäßig absehbar, dass in nächster Zeit die Produktion eines Artikels notwendig sein wird, obwohl das System wegen fehlender Bestellungen noch keinen Planauftrag erstellt hat. Neben der Erfahrung des Planers dienen hier die Materialreservierungen für Fertigungsaufträge in den Montagewerken als Indikator. Auf diese besteht über das ERP-System direkter Zugriff, eine regelmäßige und vollständige Einbeziehung dieser Daten wäre jedoch zu zeitaufwendig und kann daher nicht durch-

geführt werden. Wenn ein solcher Artikel, im Hinblick auf kurze Rüstzeit, gut in die aktuelle Produktion passen würde, erstellt der Disponent manuell einen entsprechenden Fertigungsauftrag. Kapazitätsbedarfsschwankungen, die nicht durch Kapazitätsanpassungen ausgeglichen werden können, werden soweit wie möglich durch eine entsprechende Verteilung der Produktionsmengen für externe Kunden und die Vorproduktion von Schnelldrehern geglättet. Bei zu hohem Bedarf wird dadurch die Termintreue, bei zu niedrigem die Auslastung hochgehalten. Neben diesen positiven Effekten führt das Vorgehen aber auch zu einer unerwünschten Erhöhung des Lagerbestandes und damit der Kapitalbindungskosten.

Jedem Artikel ist eine Maschine als Standard für die Produktion zugewiesen. Wie in 2.1.2 beschrieben, gibt es für einige Artikel aber auch Fertigungsversionen für andere Anlagen. Diese werden vom Disponenten, im Zuge der Auftragserzeugung und Auftragsfreigabe, in folgenden Fällen gewählt:

- Bessere Rüstreihenfolge durch Verschiebung auf andere Maschine
- Kapazitätsmangel auf der Standardmaschine
- Zu geringe Auslastung der Maschine der Fertigungsversion

Eine grobe Reihenfolgeplanung für die einzelnen Maschinen wird vom Disponenten über die Terminierung der Fertigungsaufträge durchgeführt. Die endgültige Reihenfolge wird jedoch erst an den Anlagen, durch die Fertigungsleitung, festgelegt. Dabei werden mehrere Aspekte berücksichtigt:

- Rüstreihenfolge
- Verfügbarkeit benötigter Vorprodukte
- Dringlichkeit der Produktion in Absprache mit dem Disponenten
- Personalverfügbarkeit, bspw. für kompliziertere Umrüstvorgänge

Der Planer ermöglicht also durch die Auftragsfreigabe und Maschinenzuordnung eine gute Rüstreihenfolge, umgesetzt wird diese jedoch erst durch die Fertigungsleitung.

2.3 Zielstellung und Potenzial

Ziel des Projekts war die Erstellung einer Software zur Unterstützung des Produktionsplaners der LS-Komponentenfertigung. Der erste Ansatz, ein Verfahren zur Optimierung der Arbeitsverteilung und Reihenfolgebildung zu entwickeln, wurde schnell verworfen. Damit diese Art von Verfahren ihr Potenzial zur Geltung bringen kann, muss eine gewisse Anzahl an freigegebenen Aufträgen bekannt sein, um Spielraum bei der Lösungsfindung zu haben. In der LS-Komponentenfertigung sind aber im Normalfall deutlich weniger als zehn Aufträge je Maschine freigegeben. Die Arbeitsverteilung bietet nicht allzu viele Varianten, da nur relativ wenige Artikel auf mehr als einem Maschinentyp (vgl. 2.1.2) hergestellt werden können und nach erfolgter Arbeitsverteilung ist die Reihenfolgefindung, bei meist weniger als fünf verschiedenen Artikeln, weder für den erfahrenen Disponenten noch für die Fertigungsleitung ein anspruchsvolles oder zeitraubendes Problem.

Es zeigt sich, dass unter den regelmäßig durchzuführenden Aufgaben, die Bestimmung der zu erzeugenden Fertigungsaufträge den größten Aufwand und die größte Schwierigkeit darstellt. Wie schon in Abschnitt 2.2 beschrieben, kann der Disponent aus Zeitgründen auch nicht immer alle zur Verfügung stehenden Informationen miteinbeziehen und über das gesamte Artikelspektrum ergeben sich potenziell sehr viele Möglichkeiten für die Lösung des Problems der Auftragserzeugung und Auftragsfreigabe. Außerdem basieren viele Entscheidungen, die hierbei getroffen werden müssen, auf der Erfahrung des Disponenten und die hinter den Entscheidungen stehenden Regeln sind schwer vollständig zu erfassen. Daher ist es auch nicht einfach, das Wissen über diese Regeln zwischen den Mitarbeitern zu transferieren. Wenn der erfahrene Disponent aus verschiedenen Gründen nicht zur Verfügung steht, ist eine gleichbleibend hohe Qualität der Produktionsplanung daher schwierig zu gewährleisten und der Zeitaufwand für auf diesem Gebiet weniger erfahrene Mitarbeiter liegt noch höher. Die derzeit häufig gewählte Lösung, bei geplanter, kurzfristiger Abwesenheit Fertigungsaufträge für einen längeren Zeitraum auszuschreiben, reduziert die Möglichkeit, flexibel auf Änderungen der Nachfrage zu reagieren. Daraus ergeben sich folgende Potenziale für ein computergestütztes Verfahren zur Unterstützung der Produktionsplanung:

- Der Zeitaufwand für den Disponenten wird reduziert, weil nur noch das Ergebnis der Software geprüft und gegebenenfalls leicht angepasst werden muss.

- Die Planungsqualität wird erhöht, da
 - mehr Informationen berücksichtigt werden können.
 - mehr mögliche Lösungen geprüft werden können.
- Wenn der verantwortliche Disponent nicht zur Verfügung steht, wird der Zeitaufwand für die Vertretung reduziert, die Planungsqualität wird erhöht und die Planungsflexibilität bleibt erhalten, da nicht mehr Aufträge im Voraus freigegeben werden.

2.4 Anforderungen an das Verfahren

Aus den zuvor beschriebenen Rahmenbedingungen und Zielen lassen sich die Anforderungen an das Verfahren ableiten. Zusätzlich wurden vom Unternehmen Vorgaben gemacht, um die praktische Anwendbarkeit zu gewährleisten. Die Anforderungen können nach funktionalen Kriterien gegliedert werden. Diese Gliederung wird auch durch die Struktur der Software widerspiegelt.

Unabhängig vom gewählten Ansatz werden gewisse Daten benötigt. Diese können in Stammdaten und Bewegungsdaten unterteilt werden. Stammdaten sind auftragsneutral und bleiben über einen längeren Zeitraum gleich. Typische Beispiele dafür sind Artikelstammdaten oder Rüstzeiten [8]. Als Bewegungsdaten werden typischerweise geschäftsfallbezogene Daten wie Auftragsmenge, Lieferdatum oder Auftragsnummer bezeichnet [2]. In dieser Arbeit wird der Begriff etwas weiter gefasst und beschreibt alle Daten, welche sich typischerweise zwischen zwei Programmdurchläufen ändern können. Die klassischen Bewegungsdaten werden daher um Bestandsdaten und Daten zum Kapazitätsangebot, in Form von Schichtplänen erweitert. Die Software muss all diese Daten einlesen und verarbeiten können. Um die praktische Anwendbarkeit zu gewährleisten, darf kein hoher manueller Erhebungs- und Aufbereitungsaufwand anfallen. Dies gilt insbesondere für die Bewegungsdaten, welche bei jedem Programmdurchlauf aktualisiert zur Verfügung gestellt werden müssen.

Wie in Abschnitt 2.3 beschrieben, soll das Potenzial durch die Beachtung aller verfügbaren Daten genützt werden. Besonders relevant sind die Daten bezüglich der Materialreservierungen für Fertigungsaufträge in den Montagewerken, welche in der manuellen Planung nur teilweise berücksichtigt werden können. In dieser kombiniert der Disponent die bestätigten Bestellungen im Produktionswerk mit den Reservierungen und Beständen in den Montagewerken und seinen Erfahrungen über den durchschnittlichen Bedarf und die Bedarfsentwicklung. So kann er eine bessere Abschätzung über die Nachfrageentwicklung für einen längeren Zeitraum treffen, als bei alleiniger Berücksichtigung der bestätigten Bestellungen. Um diese Fähigkeit auf eine Software zu übertragen, ist ein Prognoseverfahren notwendig. Zur Verfügung stehen Daten über die Bestellungen, Reservierungen und Lagerbestände aus den letzten Jahren.

Basierend auf der Bedarfsprognose muss ein Vorschlag ermittelt werden, für welche Artikel Fertigungsaufträge erzeugt und freigegeben werden sollen. Der Zeitraum, für den die vorhandene Fertigungskapazität mit Aufträgen eingedeckt werden soll, muss vom Anwender bestimmbar sein. Es kann aber angenommen werden, dass es sich um etwa 5 bis 10 Schichten handelt, wobei der Wert von 10 Schichten im Normalfall nur für die Planung der Wochenendproduktion erreicht wird und über das Wochenende keine Bestellungen hinzukommen. Die freigegebenen Fertigungsaufträge müssen in erste Linie den Bedarf decken und es darf zu keiner Verschlechterung des Servicegrades im Vergleich zur manuellen Produktionsplanung kommen. Weiter müssen die Aufträge so ausgewählt werden, dass durch entsprechende Reihung eine Rüstzeitreduktion möglich ist. Dazu sind, in Abhängigkeit von der Kapazitätssituation, gegebenenfalls erst später notwendige Fertigungsaufträge vorzuziehen. Die Anlagen mit niedrigen Produktionskosten sind bevorzugt auszulasten. Das bedeutet auch, dass auf diesen Anlagen die Rüstzeitreduktion noch wichtiger ist und bevorzugt größere Mengen des gleichen Artikels produziert werden sollen. Bei niedrigerer Nachfrage ist die freie Kapazität mit Schnelldrehern aufzufüllen, bis ein maximaler Lagerstand erreicht ist. Ein weiteres Ziel sind niedrige Bestände. Dieses Ziel kann durch steigende Planungsqualität erreicht werden, wenn weniger nicht benötigte Artikel produziert werden. Allerdings muss dann auch die Kapazität reduziert werden, da sonst ein Konflikt mit dem Ziel der hohen Auslastung eintritt. Andererseits kann auch das Vorziehen von Aufträgen bewusst eingeschränkt werden. Damit steigt die

Rüstzeit und die OEE sinkt, aber auch die Bestände gehen nach unten. Die entsprechende Einflussmöglichkeit auf das Verfahren soll gegeben sein. Der Produktionsplaner soll außerdem die Möglichkeit haben, die Gewichtung der verschiedenen Optimierungsziele zu beeinflussen.

Die nach der Auftragserzeugung und Auftragsfreigabe eindeutig einer Maschine zugewiesenen Fertigungsaufträge müssen in einer Reihenfolge produziert werden, die einerseits die termingerechte Fertigstellung sicherstellt und andererseits die Rüstzeit durch eine optimierte Rüstreihenfolge reduziert. Weitere Vorgaben des Unternehmens waren eine Laufzeit von weniger als zehn Minuten ohne den Datendownload aus dem ERP-System und ein möglichst modularer Aufbau, sodass das Verfahren mit vertretbarem Aufwand auch für andere Abteilungen angepasst werden kann. Die Funktionen welche die Softwarelösung unter diesen Bedingungen umfassen soll sind im Anschluss zusammengefasst dargestellt.

Geforderter Funktionsumfang des Softwarepakets:

- Die Eingangsdaten müssen in einer Form eingelesen werden, die ohne großen Aufbereitungsaufwand zur Verfügung gestellt werden kann.
- Auf Basis der bekannten Daten zu bestätigten Bestellungen, Materialreservierungen für Fertigungsaufträge in den Montagewerken und den Lagerständen in den Montagewerken muss eine Bedarfsprognose erstellt werden.
- Ein Vorschlag für zu erzeugende und freizugebende Fertigungsaufträge muss erstellt werden. Dabei müssen folgende Ziele berücksichtigt werden, wobei die Gewichtung der Ziele veränderbar sein muss.
 - Termintreue
 - Rüstoptimierung
 - Produktionskosten
 - Kapazitätsauslastung
 - Bestand
- Für die einzelnen Maschinen ist eine Auftragsreihenfolge zu erstellen, die Termintreue und Rüstoptimierung berücksichtigt.

3 Beschreibung der Verfahren

Anhand der in Abschnitt 2.4 beschriebenen Anforderungen wurde ein Softwarepaket entwickelt. Das Paket ist nach funktionalen Kriterien in Module geteilt, welche die notwendigen Funktionen umfassen, um jeweils einen Teil der Anforderungen zu erfüllen. Dabei gibt es auch Überschneidungen zwischen den Modulen. In Abschnitt 3.1 wird erläutert, welches Modul welche Anforderungen erfüllt und wie die einzelnen Module zusammenhängen. Im Anschluss werden diese Module beschrieben, wobei der Detaillierungsgrad von der Relevanz abhängt, die das Modul für die Lösung des Problems der Auftragserzeugung und Auftragsfreigabe hat.

3.1 Struktur des Softwarepakets

Das Softwarepaket ist in vier Module gegliedert. Innerhalb eines Programmdurchlaufs werden diese Module zum Teil sequenziell, zum Teil iterativ durchlaufen (siehe Abb. 1). Implementiert wurde das gesamte Softwarepaket in der objektorientierten Programmiersprache Java.

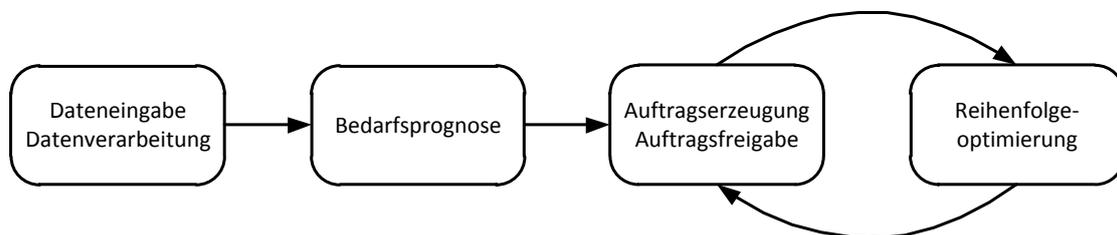


Abbildung 1: Schematischer Ablaufplan der Software

Im ersten Schritt müssen die Stamm- und Bewegungsdaten eingelesen und in eine für das Programm verwendbare Form gebracht werden. Dies ist relativ einfach, daher wird diesem Punkt kein eigenes Kapitel gewidmet, sondern die relevanten Punkte an dieser Stelle zusammengefasst. Die benötigten Daten stammen aus dem ERP-System des Unternehmens. Aus Gründen der Informationssicherheit ist kein direkter Zugriff von Fremdsoftware auf dieses System gestattet, daher werden die Daten manuell exportiert

und in MS-Excel Dateien gespeichert. Durch Vorgaben beim Datenexport wird gewährleistet, dass diese Dateien immer die gleiche Struktur aufweisen. Nach dem Einlesen sämtlicher Informationen generiert das Programm daraus die benötigten Objekte wie Maschinen, Artikel, bereits bekannte Fertigungsaufträge oder Bestellungen. Wichtig ist, dass sämtliche Bewegungsdaten über nur zwei Abfragen aus dem ERP-System gewonnen werden können. Dies reduziert den Aufwand für den Produktionsplaner und erhöht damit sowohl den Nutzen als auch die Anwendbarkeit des Programms. Aufgrund der besonderen Art der Abbildung der Bedarfe im System, muss nach dem Einlesen eine Bearbeitung erfolgen, um die tatsächlichen Bedarfsdaten zu erhalten. Da manche ungewöhnliche Bedarfszustände nach Abbildung im ERP-System gleich aussehen, entsteht bei diesem Schritt eine Unschärfe. Diese kann jedoch ohne Zusatzaufwand seitens des Anwenders nicht vermieden werden. Eine Auflistung der Daten, die benötigt werden, findet sich im Anhang A.

Nach dem Einlesen der Daten werden diese durch die drei Module bearbeitet, die in den folgenden Unterkapiteln eingehend beschrieben werden. Zuerst wird aus den bekannten Bedarfsdaten eine Bedarfsprognose für den Betrachtungszeitraum erstellt. Die Länge dieses Zeitraums ist frei wählbar, jedoch basieren alle präsentierten Ergebnisse auf einem Betrachtungszeitraum von 28 Tagen. In der weiteren Darstellung wird daher diese Dauer angenommen. Zur Erstellung der Bedarfsprognose wird für jeden Artikel ein Künstliches Neuronales Netz (KNN) erstellt und anhand von Vergangenheitsdaten trainiert. Mit den KNN werden nur Bedarfe des Werksverbundes für MTS-Artikel prognostiziert. Bestellungen von MTO-Artikeln und Bestellungen externer Kunden schwanken in Frequenz und Menge deutlich stärker und sind deshalb kaum zu prognostizieren. Außerdem sind diese Bestellungen aufgrund der höheren vereinbarten Lieferzeit länger im Voraus bekannt. Da die Genauigkeit der Bedarfsprognose abnimmt, je weiter in der Zukunft ein zu prognostizierender Bedarf liegt, wird angenommen, dass die Gesamtqualität der Bedarfsvorschau durch die Nichtbeachtung von weit in der Zukunft auftretenden, noch nicht bekannten, Bestellungen dieser Art nicht stark abnimmt. Diese Annahme ist auch damit zu begründen, dass weniger als 1% der Produktionsmenge auf MTO-Artikel entfällt und ein Großteil der Bestellungen externer Kunden etwa einen Monat im Voraus eintrifft. Die Bedarfe, mit denen im weiteren Verlauf gerechnet wird, ergeben sich aus der Bedarfsprognose für den Werksverbund, den Bestellungen für MTO-Artikel und den Bestellungen externer Kunden. Manche Bestellungen von externen Kunden sind so groß,

dass die Produktion der benötigten Artikel auf mehrere Produktionszyklen aufgeteilt wird. Durch diese Aufteilung soll eine Glättung des Bedarfsverlaufs erreicht werden, daher ist es wichtig, die aufgeteilten Bedarfsmengen nicht an einem bestimmten Tag fällig zu stellen, sondern eine gewisse Toleranz zu gewähren. Dies wird erreicht, indem die aufgeteilten Mengen als Bestandsaufbau angesehen werden, also zusätzlich auf Lager liegen sollen. Wie die Verfahren zur Auftragserzeugung und Auftragsfreigabe mit Bestandsaufbau umgehen, wird bei der Beschreibung des jeweiligen Verfahrens erläutert.

Basierend auf der im Prognosemodul erstellten Bedarfsvorhersage müssen die entsprechenden Fertigungsaufträge erzeugt und freigegeben werden. Weiter muss jeder Auftrag einer passenden Maschine zugeordnet werden. Dieses Modul stellt den Kern des Softwarepakets dar. Es werden zwei neu entwickelte Verfahren beschrieben, mit denen das Problem der Auftragserzeugung, Auftragsfreigabe und Arbeitsverteilung bei reihenfolgeabhängigen Rüstzeiten gelöst werden kann. Die Entwicklung neuer Verfahren ist notwendig, da die aus der Literatur bekannten Methoden nicht anwendbar sind. Die einfachen scheitern an der Variantenvielfalt [18], komplexere an der Berücksichtigung von reihenfolgeabhängigen Rüstzeiten oder der Arbeitsverteilung. Das erste neu entwickelte Verfahren ist ein multikriterieller genetischer Algorithmus (siehe Abschnitt 3.3), dessen Individuen die Produktion jeder Maschine für den gesamten Betrachtungszeitraum festlegen. Diese Lösungen werden hinsichtlich der verursachten Fehlmengen, Rüstzeiten, Produktions- und Bestandskosten bewertet. Das zweite Verfahren ist eine einfache Heuristik, die aus einem Pool möglicher Fertigungsaufträge so viele auswählt, dass jede Maschine zumindest für eine vom Anwender festgelegte Periode ausgelastet wird. Die Auswahl erfolgt nach den Kriterien Dringlichkeit und Rüstzeitreduktion (siehe Abschnitt 3.4).

Beide Verfahren legen direkt fest, welche Aufträge auf welcher Maschine produziert werden. Zur Bildung von Auftragsreihenfolgen auf den einzelnen Maschinen kommen Methoden zum Einsatz, die von den Verfahren zur Auftragserzeugung, Auftragsfreigabe und Arbeitsverteilung deutlich abgrenzbar und unabhängig sind. Welche Methoden das sind und wie unter mehreren Lösungen des multikriteriellen Reihenfolgeproblems eine ausgewählt werden kann, wird in Kapitel 3.5 beschrieben. Die Reihenfolgebildung wird aufgrund der Unabhängigkeit als eigenständiges Modul dargestellt. Allerdings können Lösungen der Verfahren zur Auftragserzeugung, Auftragsfreigabe und Arbeitsverteilung

erst nach der Feinterminierung durch Reihenfolgebildung hinsichtlich Fehlmengen und Rüstzeiten bewertet werden. Daher stellt dieses Modul einen integralen Bestandteil der neu entwickelten Verfahren dar.

3.2 Bedarfsprognose durch Künstliche Neuronale Netze

Das Konzept der KNN stellt eine Abstraktion der Funktionsweise des menschlichen Gehirns dar. Ein KNN besteht aus vielen künstlichen Neuronen. Diese Neuronen besitzen mehrere, gewichtete Eingangssignale. Aus den Eingangssignalen wird im Neuron mittels einer einfachen mathematischen Funktion ein Ausgangssignal berechnet, das wiederum als Eingangssignal für nachgelagerte Neuronen dient (siehe Abb. 2). Durch einen Lern-

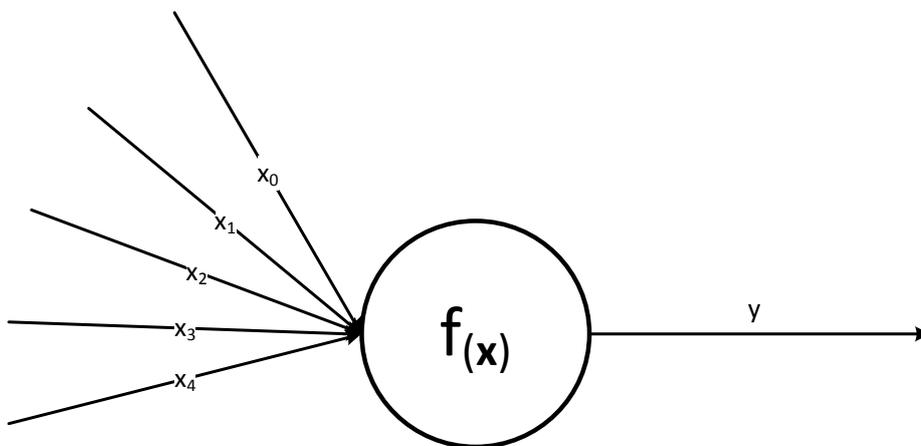


Abbildung 2: Schematische Darstellung eines künstlichen Neurons

prozess kann ein KNN für bestimmte Aufgaben trainiert werden, indem die Gewichtungen der Eingangssignale verändert werden. Eine häufig eingesetzte Art von KNN ist das MLP. Dieses ist gut mit dem Backpropagation-Algorithmus, einem sehr weit verbreiteten Lernverfahren für KNN, kombinierbar. Der Literatur zufolge, ist diese Kombination das am häufigsten angewendete Modell, wenn ein KNN anhand bekannter Daten trainiert werden soll [22], daher wird es auch in diesem Paket für die Bedarfsprognose verwendet. Jedes MLP besteht aus mehreren Arten von Schichten (siehe Abb. 3):

Input-Schicht

Über die Neuronen dieser Schicht werden die Input-Daten in das MLP eingegeben. Sie empfangen daher keine Eingangssignale von anderen Neuronen.

Output-Schicht

Die Neuronen der Output-Schicht enthalten die Ergebnisse des MLP und senden daher kein weiterführendes Ausgangssignal.

Verborgene Schichten

Beliebig viele dieser Schichten liegen zwischen der Input- und der Output-Schicht. Die Neuronen dieser Schichten haben mehrere Eingangssignale und ein Ausgangssignal.

Die Verbindungen (Kanten) zwischen den Neuronen (Knoten) sind beim MLP vorgegeben. Jedes Neuron ist mit jedem Neuron aus der nachfolgenden Schicht verbunden. Zwischen den Neuronen einer Schicht gibt es keine Kanten. Es gibt auch keine Kanten die zurück in eine vorhergehende Schicht führen oder solche, die eine Schicht überspringen und direkt in eine spätere Schicht führen. Die Anzahl der Neuronen in Input- und Output-Schicht wird in der Regel durch die Problemstellung vorgegeben. Wie viele verborgene Schichten genutzt werden und aus wie vielen Neuronen diese jeweils bestehen, muss für jeden Anwendungsfall entschieden werden und stellt eine der Herausforderungen in der Anwendung von KNN dar. Weiter muss entschieden werden, welche Aktivierungsfunktionen für die Berechnung der Ausgangssignale verwendet werden. In diesem Abschnitt wird die Umsetzung der Bedarfsprognose durch KNN für den konkreten Anwendungsfall beschrieben. Es wird dargelegt, welche Topologie, Verfahren und Verfahrensparameter sich als geeignet erweisen. Für die ausführliche Behandlung von theoretischen Grundlagen und weiterführende Informationen zu KNN sei auf die einschlägige Literatur, z.B. Haykin [13], Reed [22] oder Rojas [24] verwiesen.

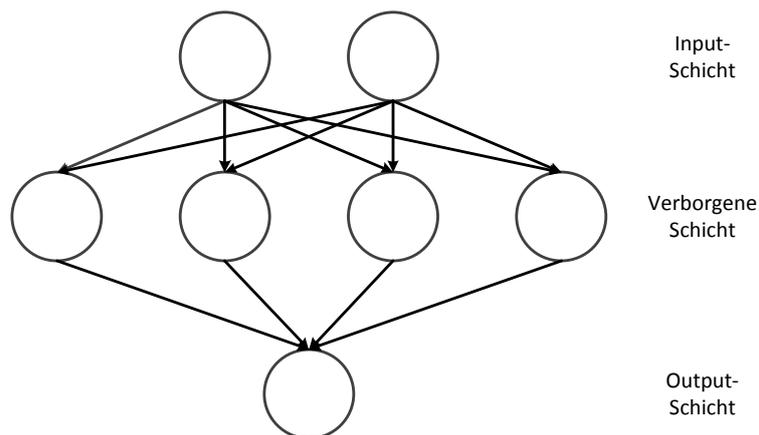


Abbildung 3: MLP mit einer verborgenen Schicht

Das Ziel der Bedarfsprognose ist, für einen gewissen Zeithorizont, die Bedarfe auf Articlebene vorherzusagen. Die Tagesbedarfe für beliebige Paare von Tagen weisen eine sehr geringe Korrelation auf. Daher könnte durch Aggregation, beispielsweise auf Wochenebene, der Prognosefehler reduziert werden [17]. Allerdings würden dadurch Informationen über das exakte Bedarfsdatum verloren gehen. Da die Prognose benötigt wird, um ein feinterminiertes Produktionsprogramm zu erstellen und dessen Qualität zu bewerten, müsste für die Reduktion des Prognosefehlers eine erhöhte Unsicherheit bei der Bewertung der Termintreue in Kauf genommen werden. Um dies zu vermeiden, werden die Tagesbedarfe auf Articlebene prognostiziert. Zwischen den Tagesbedarfen verschiedener Artikel besteht ebenfalls kein nennenswerter Zusammenhang. Daher kann der Bedarf für jeden Artikel getrennt prognostiziert werden. Das geschieht durch die Erstellung eines KNN pro Artikel.

Die 28 Tage des Betrachtungszeitraums können auf 20 Tage reduziert werden, da an Wochenenden kein Bedarf auftritt. Aufgrund der geringen Korrelation zwischen den Tagesbedarfen auf Articlebene sind Vergangenheitsdaten als Input in diesem Anwendungsfall nicht geeignet. Den stärksten Einfluss auf die zukünftige Bedarfsentwicklung haben bestätigte Bestellungen im Produktionswerk und Materialreservierungen für Fertigungsaufträge sowie die Lagerstände in den Montagewerken. Daher werden diese Daten als Input für die Bedarfsprognose herangezogen. Noch nicht freigegebene Bestellanforderungen werden nicht berücksichtigt, da bei diesen häufig noch Bedarfsdatum oder Bedarfsmenge geändert werden. Bestellungen werden nur für Tage bestätigt, an denen auch der Versand möglich ist und bei den Materialreservierungen für Fertigungsaufträge beträgt der Anteil, der an einem Wochenende benötigt wird, weniger als ein Prozent. Diese Mengen werden dem Freitag vor dem tatsächlichen Bedarfstermin zugeschlagen. Dadurch kann auch bei den Input-Daten die Anzahl der betrachteten Tage auf 20 reduziert werden. Tests zeigten, dass die Prognosequalität durch die Entfernung der Wochenenden, d.h. regelmäßig auftretender Tage ohne Bedarf, deutlich ansteigt.

Die Anzahl der Input- und Output-Knoten ist von der Größe der entsprechenden Datensätze abhängig. In einem MLP ist die Rechenzeit für den Lernvorgang direkt proportional zur Anzahl der Knoten. Daher wird der Input-Datensatz durch Aggregation der Daten für die Montagewerke verkleinert. Input- und Output-Schicht des MLP sehen daher aus wie folgt:

Input-Schicht

- 20 Knoten für bestätigte Bestellungen. Jeder Knoten entspricht der bestätigten Bestellmenge eines Wochentages im Betrachtungszeitraum.
- 20 Knoten für Materialreservierungen für Fertigungsaufträge in den Montagewerken. Jeder Knoten entspricht der Summe der reservierten Mengen in allen Montagewerken für einen Wochentag im Betrachtungszeitraum.
- 1 Knoten für die Summe der Lagerstände in den Montagewerken.

Output-Schicht

- 20 Knoten für prognostizierte Bestellmengen. Jeder Knoten entspricht der prognostizierten Bestellmenge für einen Wochentag im Betrachtungszeitraum.

Aus Vergangenheitsdaten werden entsprechende Datensätze erzeugt, die auch Soll-Output-Daten, d.h. die tatsächlich eingetretenen Bedarfe, enthalten. Diese Datensätze werden in drei Gruppen eingeteilt. In diesem Projekt wurden etwa 480 Datensätze als Trainingsdaten, 50 als Validierungsdaten und 10 als Testdaten genutzt. Die Validierungs- und Testdaten werden zufällig aus der Gesamtheit der zur Verfügung stehenden Datensätze ausgewählt. Anhand der Trainingsdaten werden die Kantengewichtungen durch ein Lernverfahren so verändert, dass das Ergebnis des KNN möglichst genau dem Soll-Output entspricht. Die Bestimmung der optimalen Kantengewichtung durch Minimierung des Prognosefehlers gegenüber den Trainingsdaten, birgt jedoch das Risiko, dass das Netz zwar sehr gut auf die Trainingsdaten abgestimmt ist, für andere Inputs aber keine guten Ergebnisse liefert. Daher werden die Validierungsdaten genutzt, um nach jeder Trainings-Iteration den Prognosefehler des Netzes gegenüber diesen Validierungsdaten zu berechnen. Gewählt werden nach Beendigung der Trainingsphase jene Kantengewichtungen, die den Prognosefehler gegenüber den Validierungsdaten minimieren. Die Testdaten werden verwendet, um trainierte Netze, die sich in Topologie oder Verfahrensparametern unterscheiden, zu vergleichen.

Um geeignete Werte für Anzahl und Größen der verborgenen Schichten des MLP sowie für die Verfahrensparameter zu finden, wurden einige Tests durchgeführt. Jeder Test wurde für je einen X-, Y- und Z-Teil (vgl. XYZ-Analyse [32]) durchgeführt, um sicherzustellen, dass das Modell nicht nur für Artikel mit einem bestimmten Verbrauchsverlauf optimiert wird. Es stellte sich jedoch heraus, dass die Eignung von Topologien und Parametern von der XYZ-Klassifizierung unabhängig ist. Nach Reed [22] genügen zwei verborgene Schichten, um bei ausreichend großer Anzahl von Knoten beliebige Funktionen abzubilden. Zhang [31] tendiert für ähnliche Probleme ebenfalls zu zwei verborgenen Schichten. Die Tests bestätigen, dass MLP mit zwei verborgenen Schichten gegenüber solchen mit einer verborgenen Schicht, bei zunehmender Anzahl von Trainingsiterationen, tendenziell bessere Ergebnisse liefern. Bezüglich der Knotenanzahl in den verborgenen Schichten ist festzuhalten, dass eindeutig bessere Ergebnisse erreicht werden, wenn die zweite verborgene Schicht größer ist als die erste. Das beste Ergebnis wird mit einer ersten verborgenen Schicht aus 40 Knoten und einer zweiten verborgenen Schicht aus 130 Knoten erreicht. Für jeden Knoten muss eine Aktivierungsfunktion festgelegt werden. Diese definiert, wie die gewichteten Eingangssignale in ein Ausgangssignal umgewandelt werden. Um das Backpropagation-Verfahren anwenden zu können, dürfen nur differenzierbare Funktionen gewählt werden. In dem für die KNN verwendeten Software-Paket, kann die Aktivierungsfunktion für die Knoten der Input- und der Output-Schicht sowie für die aller verborgenen Schichten, getrennt festgelegt werden. Durch die Wahl unterschiedlicher Aktivierungsfunktionen wird die Prognosequalität beeinflusst. Für die Testdaten werden die besten Resultate mit einer linearen Aktivierungsfunktion für die Input-Knoten, einer Tangens Hyperbolicus-Funktion für die Knoten der verborgenen Schichten und einer sigmoiden Aktivierungsfunktion für die Output-Knoten erzielt. Das Standard-Lernverfahren Backpropagation liefert in den Tests bessere Ergebnisse, wenn es durch einen Trägheitsterm erweitert wird. Dadurch wird bei der Änderung einer Kantengewichtung auch die vorangegangene Änderung der Gewichtung dieser Kante berücksichtigt. Auf diese Weise können Schwingungen der Kantengewichtung reduziert werden und in bestimmten Bereichen der Lernfortschritt beschleunigt werden [22]. Für die Lernrate ergibt sich aus den Testergebnissen, dass sie allenfalls zu Beginn etwas höher sein soll, um das Netz grob zu kalibrieren. Danach ist eine hohe Lernrate kontraproduktiv, da die Kantengewichtungen bei jedem Lernvorgang zu stark an den aktuellen Trainingsdatensatz angepasst werden und keine Verbesserung hinsichtlich der Validierungsdaten erreicht wird. Die Grobabstimmung ist auch mit niedriger Lernrate nicht viel langsamer.

Daher wird für alle Iterationen eine Lernrate von 0,01 verwendet. Für den Trägheitsterm erweisen sich Koeffizienten in einem Bereich zwischen 0,3 und 0,9 als geeignet. Das bedeutet, dass vergangene Änderungen relativ starken Einfluss auf die aktuelle Änderung haben. Möglicherweise führen die relativ großen Unterschiede zwischen den einzelnen Trainingsdatensätzen bei kleinem Trägheitsterm zu starken Schwingungen der Kantengewichtungen. Bei einem Trägheitskoeffizienten von 0,3 wird der minimale Prognosefehler gegenüber den Validierungsdaten erst später erreicht als mit den höheren Werten. Allerdings ist auch das Ergebnis etwas besser, daher wird der Trägheitskoeffizient auf 0,3 festgelegt. Es muss erwähnt werden, dass, bei Veränderung der Knotenanzahl in den verborgenen Schichten, andere Werte für Lernrate und Trägheitskoeffizient das beste Ergebnis liefern. Tendenziell sind jedoch eine niedrige Lernrate und ein eher hoher Trägheitskoeffizient eine gute Wahl. Für MLP mit nur einer verborgenen Schicht ergeben die Tests bessere Ergebnisse, wenn für die Neuronen in den verborgenen Schichten ebenfalls eine sigmoide Aktivierungsfunktion gewählt wird. Dadurch ergeben sich allerdings wieder Änderungen bei der idealen Lernrate. Das zeigt, dass das Finden einer guten Topologie und guter Werte für die Verfahrensparameter ein komplexes Problem ist, das meist durch systematische Tests gelöst wird [31]. Im Folgenden werden die Resultate dieser Tests für das konkrete Problem in kompakter Form präsentiert.

Art des KNN

- Mehrschichtiges Perzeptron (MLP)

Topologie

- 1 Input-Schicht (41 Knoten)
- 2 verborgene Schichten (40/130 Knoten)
- 1 Output-Schicht (20 Knoten)

Lernverfahren

- Backpropagation mit Trägheitsterm (Momentum)
- Lernrate: 0,01; Trägheitskoeffizient: 0,3;

Aktivierungsfunktion

- Input-Schicht: lineare Funktion
- verborgene Schichten: Tangens Hyperbolicus
- Output-Schicht: sigmoide Funktion

3.3 Auftragserzeugung und Auftragsfreigabe durch Genetischen Algorithmus

Genetische Algorithmen sind Metaheuristiken, die das Funktionsprinzip der Evolution nachahmen [28]. Als Metaheuristiken sind sie nicht auf ein spezifisches Optimierungsproblem beschränkt, sondern definieren eine allgemeine Folge von Schritten, die zur Erreichung eines Ziels führen sollen. Um eine Metaheuristik auf ein spezielles Problem anzuwenden, müssen diese Schritte problemspezifisch implementiert werden [9]. Genetische Algorithmen bestehen aus den Teilen Selektion, Kombination und Mutation [19]. Außerdem muss eine geeignete Codierung der Lösungen gefunden werden. Bei der Lösung des Problems der Auftragserzeugung, Auftragsfreigabe und Arbeitsverteilung unter Berücksichtigung reihenfolgeabhängiger Rüstzeiten müssen verschiedene, zueinander in Konkurrenz stehende Ziele berücksichtigt werden (vgl. Kapitel 2.4). Daher wird der GA zum multikriteriellen Genetischen Algorithmus erweitert. Dabei treten insbesondere für die Lösungsbewertung zusätzliche Herausforderungen auf [7]. Der Zusatz *multikriteriell* wird im weiteren Verlauf weggelassen. Sofern nicht explizit anders angegeben, ist aber immer der multikriterielle Genetische Algorithmus gemeint. In diesem Abschnitt wird eine Implementierung des GA beschrieben, mit der das in Kapitel 2 beschriebene Problem behandelt werden kann. Es wird angenommen, dass die zugrunde liegende Theorie weitestgehend bekannt ist, daher wird auf diese nur am Rande eingegangen. Für die Grundlagen Genetischer Algorithmen sei auf Michalewicz [19] verwiesen, ausführliche Informationen zum multikriteriellen GA werden bei Deb [7] präsentiert.

Der gewählte Ansatz für die Anwendung eines GA zur Lösung des Problems der Auftragserzeugung, Auftragsfreigabe und Arbeitsverteilung unter Berücksichtigung reihenfolgeabhängiger Rüstzeiten ist, auf Basis der prognostizierten Bedarfe ein feinterminiertes Produktionsprogramm für den gesamten Betrachtungszeitraum zu erstellen. Für jeden im Betrachtungszeitraum bearbeiteten Auftrag sind daher die Maschine, auf der er bearbeitet wird, der geplante Produktionsstart- und der geplante Fertigstellungszeitpunkt festzulegen. Unter der Annahme, dass die Bedarfsprognose korrekt ist, können für ein solches Produktionsprogramm Fehlmengen, Rüstzeiten, Produktionskosten, Bestände und Kapazitätsauslastung exakt berechnet werden. Die Qualität einer Lösung hängt von der Zielerreichung der Teilziele ab. Wie diese in die Lösungsbewertung einfließen, wird in Abschnitt 3.3.3 beschrieben.

3.3.1 Struktur der Individuen

Durch die Codierung wird festgelegt wie das einer Lösung zugeordnete Individuum aussieht. Wichtig ist, dass jedem Individuum genau eine Lösung zuzuordnen sein muss. Die Codierung der Individuen ist jener Teil des GA, der am stärksten problemspezifisch anzupassen ist. Für viele Anwendungsfälle ist die binäre Codierung [19] geeignet und ausreichend. Bei komplexen Problemen, deren Lösungen viele Informationen beinhalten, führt diese Art der Codierung jedoch zu sehr großen und unübersichtlichen Individuen. Beim gewählten Ansatz für die Implementierung des GA enthält jedes Individuum Informationen über mehrere hundert Fertigungsaufträge. Für jeden Auftrag muss durch die Codierung eindeutig bestimmt sein, auf welcher Maschine er wann produziert wird. Außerdem müssen die Informationen derart codiert sein, dass die genetischen Operatoren vernünftig eingesetzt werden können.

Der Grundgedanke hinter der gewählten Codierung ist, die benötigten Informationen in aggregierter Form darzustellen, sodass sie ohne umfangreiche Verarbeitung verständlich sind. Dadurch müssen die genetischen Operatoren komplexer gestaltet werden als bei Binärcodierung. Andererseits ist die Problemstellung besser fassbar und die Operatoren können leichter so gestaltet werden, dass sie sinnvolle Ergebnisse liefern. In Hinblick auf die qualitative Zusammensetzung des Produktionsprogramms, wird der Lösungsraum durch die erlaubten Maschine-Artikel-Kombinationen beschränkt. Lösungen, die andere Kombinationen enthalten, d.h. die Produktion eines Artikels auf einer Maschine vorschlagen, auf der dieser nicht gefertigt werden kann, sind ungültig. Ungültige Lösungen können entweder verworfen oder repariert werden, besser ist jedoch ihrer Entstehung vorzubeugen. In der Implementierung wird daher jede erlaubte Maschine-Artikel-Kombination durch ein *Maschine-Artikel-Gen* repräsentiert. Ungültige Kombinationen können dadurch, mangels Darstellungsmöglichkeit, nicht auftreten. Da viele Artikel auf mehreren Maschinen des gleichen Typs gefertigt werden können und einige Artikel auch auf unterschiedlichen Maschinentypen, ergeben sich etwa 500 Gene pro Individuum. Die Gene sind nach Maschine und Material sortiert, daher steht ein Gen, das eine bestimmte Maschinen-Artikel-Kombination darstellt in jedem Individuum an der gleichen Position. Dies ist insbesondere für die Kombinationsoperatoren von Bedeutung.

Jedes Maschine-Artikel-Gen (Maschine: m ; Artikel: a) enthält eine Liste aller Fertigungsaufträge für die Produktion von a auf m , die in der dargestellten Lösung vorkommen. In Analogie zur Binärcodierung von Individuen stellt diese Auftragsliste den Wert des Gens dar. Die Aufträge enthalten Informationen zu Auftragsmenge und Produktionszeitpunkt. Die Codierung des Produktionszeitpunktes kann aber nicht direkt durch die Angabe des Zeitpunktes erfolgen, an dem die Fertigung startet. Dies würde bei Manipulation der Individuen durch Kombinations- und Mutations-Operatoren unvermeidlich zu zahlreichen ungültigen Lösungen führen, etwa wenn zwischen zwei Individuen Fertigungsaufträge getauscht werden, deren Fertigung nicht exakt zur gleichen Zeit startet und endet. Daher wird eine indirekte Codierung gewählt. Dahinter steht die Idee, dass der GA die Fertigungsaufträge erzeugt, die Produktionsmenge bestimmt und sie einer Maschine zuordnet. Die Reihenfolgeerstellung auf den Maschinen und damit die Feinterminierung übernimmt das Modul zur Reihenfolgeoptimierung. Da dieses Modul exakte und heuristische Methoden ohne Zufallseinfluss einsetzt, ist das Resultat der Feinterminierung bei gleichem Input immer dasselbe. Dadurch ist die Bedingung erfüllt, dass jedem Individuum genau eine Lösung zugeordnet sein muss. Allerdings hat der GA bei diesem Ansatz keinen Einfluss auf den Zeitpunkt der Produktion. Die Untersuchung des Lösungsraums in dieser Dimension gänzlich auf die Verfahren der Reihenfolgeoptimierung zu übertragen, ist problematisch, da dieser Teil des Lösungsraums sehr groß ist und daher in akzeptabler Laufzeit nur zu einem geringen Teil durchsucht werden kann. Die Methoden der Reihenfolgeoptimierung haben gegenüber dem GA den Nachteil, dass bereits gefundene Informationen über gute und schlechte Lösungen nicht berücksichtigt werden. Daher wird ein Mittelweg gewählt und der Betrachtungszeitraum in mehrere Produktionsintervalle geteilt. Der GA weist den Aufträgen das Produktionsintervall zu, in dem sie produziert werden sollen. Die Feinterminierung innerhalb des Intervalls wird im Zuge der Reihenfolgeoptimierung durchgeführt. Daraus entsteht auch der Vorteil, dass bei der Reihenfolgeoptimierung der Lösungsraum deutlich verkleinert und daher gründlicher durchsucht werden kann. Außerdem erweist sich die Einteilung in Produktionsintervalle auch bei der Lösungsbewertung als nützlich. In der Implementierung wurden für die Intervalle folgende Längen in Tagen festgelegt: $\{3, 4, 7, 7, 7\}$. Da das Ergebnis des Verfahrens ein Vorschlag für die Auftragsfreigabe der nächsten 3 - 10 Schichten ist, sind die ersten Tage des Betrachtungszeitraums von höherer Bedeutung. Durch die kürzeren Intervalle in diesem Bereich erhält der GA mehr Kontrolle über die Terminierung und für die Reihenfolgeoptimierung können eher exakte Verfahren zum Einsatz kommen.

3.3.2 Lösungsauswertung

Der GA erzeugt laufend neue Individuen. Unabhängig davon, ob die neuen Individuen bei der Initialisierung des Verfahrens oder durch genetische Operatoren erzeugt werden, bestehen sie zu Beginn nur aus den in Abschnitt 3.3.1 beschriebenen Genen, die eine Liste von nicht terminierten Fertigungsaufträgen enthalten. In diesem Zustand könnten manche der Kombinations- und Mutationsoperatoren eingesetzt werden. Aber für die Bewertung des Individuums und daher auch für den Einsatz der Selektionsverfahren ist eine Bearbeitung notwendig. Aus dem Individuum muss die zugehörige Lösung, d.h. das feinterminierte Produktionsprogramm entwickelt werden. Einen großen Teil dieser Aufgabe übernimmt das Modul zur Reihenfolgebildung. Dieses berechnet aus den jeweils einer Maschine und einem Produktionsintervall zugeordneten Fertigungsaufträgen die Auftragsreihenfolge je Maschine (siehe Abschnitt 3.5). Im Zuge der Reihenfolgebildung wird für jeden Auftrag der exakte Start- und Endzeitpunkt der Produktion berechnet. Es wird auch berechnet und berücksichtigt, wie lange eine Maschine zwischen Aufträgen für unterschiedliche Artikel umgerüstet werden muss. Diese exakte Berechnung wird als *Feinterminierung* bezeichnet. Die Feinterminierung wird vom Programm auf die Millisekunde genau durchgeführt. In der Realität ist es natürlich nicht möglich diesen Plan exakt einzuhalten, das ist aber auch nicht notwendig. Wichtig ist, dass die angenommenen Rüst- und Produktionszeiten in etwa dem Mittelwert der tatsächlichen Zeiten entsprechen. Dann kann die Feinterminierung genutzt werden, um sicherzustellen, dass die benötigten Ressourcen für die eingeplanten Aufträge auch tatsächlich vorhanden sind [18]. Außerdem nutzt die Software das Ergebnis der Feinplanung für die Bewertung der Lösung hinsichtlich der Einzelziele. In gängigen Tools zur Feinplanung erfolgt die Feinterminierung meist minutengenau. Dies würde auch hier für praktischen Belange völlig ausreichen. Allerdings basieren die Zeitpunkt-Objekte, die in der Software für sämtliche Terminierungen benutzt werden, auf der Basiseinheit Millisekunden. Um Fehlerquellen zu vermeiden werden alle zeitbezogenen Größen in Millisekunden berechnet.

Für diese Bewertung wird aus dem Anfangslagerbestand, den prognostizierten Bedarfsdaten und den Fertigstellungszeitpunkten der Fertigungsaufträge der *virtuelle Bestandsverlauf* für jeden Artikel berechnet. Bedarfsmengen werden zum Bedarfstermin vom virtuellen Bestand abgezogen. Das bedeutet, dass dieser negativ wird, wenn Bedarfe nicht gedeckt werden können. Außer den Bedarfen werden gegebenenfalls die Mengen

für einen geplanten Lagerbestandsaufbau (siehe Abschnitt 3.1) abgezogen. In der Implementierung wird jedem Bestandsaufbau ein Parameter zugewiesen, der angibt, welcher Anteil der Zielmenge mindestens erreicht werden muss. Über diesen Anteil hinausgehende Mengen werden nur abgezogen, bis der Bestand auf 0 sinkt, und führen daher nicht zu Fehlmengen. Bei der Aufteilung eines großen Kundenauftrags auf vier Wochen wird die Zielmenge des Bestandsaufbaus in jeder Woche um 25% des Auftragsvolumens erhöht. Der Anteil der Zielmenge, der mindestens erreicht werden muss, steigt von Woche zu Woche. Die Zahlen sind zusammengefasst in Tabelle 1 dargestellt. Dem zugrunde liegt die

Tabelle 1: Beispiel Bestandsaufbau

	Woche 1	Woche 2	Woche 3	Woche 4
Zielmenge als Anteil der Auftragsmenge	25%	50%	75%	100%
Mindestmenge als Anteil der Zielmenge	0%	50%	80%	100%

Idee, dass am Ende des Bestandsaufbaus für die Bedienung des Kundenauftrags die volle Menge fertig produziert sein muss, die Fertigstellung der ersten Wochenmengen jedoch nicht unbedingt fristgerecht erfolgen muss. Wenn doch von Beginn an die vollen Mengen produziert werden, werden diese vom bewerteten Bestand abgezogen, sodass nicht ein hoher Wert des Lagerbestandes den Bestandsaufbau verhindert. Durch dieses Vorgehen soll der GA die Möglichkeit erhalten, große Kundenbestellungen zur Bedarfsglättung einzusetzen.

Analog dem virtuellen Bestandsverlauf wird ein zweiter Verlauf berechnet, der dem virtuellen Bestandsverlauf abzüglich des Sicherheitsbestandes entspricht. Dieser Verlauf zeigt, um wie viel der Bestand über oder unter dem Sicherheitsbestand liegt, und wird daher als *virtueller Sicherheitsbestandsverlauf* bezeichnet. Nach der Berechnung dieser virtuellen Bestandsverläufe sind alle Daten für die Bewertung der Lösung vorhanden. Wie eine Lösung in Hinblick auf die Einzelziele bewertet wird, die sich aus den in Abschnitt 2.4 festgehaltenen Anforderungen ergeben, wird nachfolgend beschrieben. Welcher Wert für jedes Ziel berechnet wird und welche Einheit dieser hat, wird in Tabelle 2 in kompakter Form gezeigt. Wie die Zielerreichung der Einzelziele in die Bewertung der Lösung einfließt, wird in Abschnitt 3.3.3 beschrieben. Vier der fünf Einzelziele müssen minimiert

Tabelle 2: Bewertung der Einzelziele

Ziel	bewertete Größe	Einheit
Rüstzeit	Summe Rüstzeiten aller Maschinen	Millisekunden
Fehlmengen	gewichtete Fehlteiltage aller Artikel	Fehlteiltage
Bestand	gewichtete Bestandswerte im Betrachtungszeitraum	Euro
Kapazitätsauslastung	gewichtete freie Maschinenkapazität	Stück
Produktionskosten	Summe Opportunitätskosten für Fertigung mit anderer als der günstigsten Fertigungstechnologie	Euro

werden, daher wird das Problem als Minimierungsproblem beschrieben. Die Lösungsbeurteilung hinsichtlich der Rüstzeit ist einfach. Im Zuge der Reihenfolgebildung wird für die Auftragsreihenfolge jeder Maschine die Gesamtrüstzeit in Millisekunden berechnet. Die Summe dieser Rüstzeiten wird direkt zur Bewertung des Individuums herangezogen. Bei Fehlmengen ist sowohl die fehlende Menge, als auch die Dauer, bis der Bedarf gedeckt werden kann, von Bedeutung. Daher werden die Fehlmengen nach *Fehlteiltagen* bewertet, wobei ein Fehlteiltag einem Bedarf von einem Stück, der für einen Tag nicht gedeckt werden kann, entspricht. Die Fehlteiltage je Artikel können durch Aufsummieren über die negativen Werte der virtuellen Bestandsverläufe ermittelt werden. Da für die Bewertung der Lösung die ersten Produktionsintervalle von höherer Bedeutung sind, werden die Fehlteiltage je Produktionsintervall ermittelt, gewichtet und aufsummiert. Die Unterschreitung eines Sicherheitsbestandes ist ein negatives Merkmal einer Lösung, aber nicht so problematisch, wie wenn Bedarfe nicht gedeckt werden können. Daher werden die gewichteten Fehlteiltage des virtuellen Bestandsverlaufs stärker gewichtet als die gewichteten Fehlteiltage des virtuellen Sicherheitsbestandsverlaufs. Die Summe dieser beiden gewichteten Werte ist das zu minimierende Merkmal hinsichtlich der Fehlmengen. Der Bestand wird einmal pro Woche berücksichtigt. Zu diesem Zeitpunkt wird der Gesamtwert aller auf Lager liegenden Artikel berechnet. Für die Lösungsbeurteilung werden die Ergebnisse der terminlich näher liegenden Bestandsbewertung stärker gewichtet und die gewichteten Ergebnisse aufsummiert. Die Kapazitätsauslastung ist das einzige Einzelziel, das maximiert werden muss. Die Umwandlung in ein zu minimierendes Ziel kann aber einfach durch die Betrachtung der nicht belegten Maschinenkapazität erfolgen. Es wird allerdings nicht der gesamte Betrachtungszeitraum bewertet, sondern nur der vom Benutzer festgelegte Zeitraum, für den die Software Aufträge zur Freigabe vorschlagen soll.

Freie Maschinenkapazität wird in Stück angegeben und nach der Maschine gewichtet, für die sie anfällt, da die Auslastung der Maschinen mit günstiger Fertigungstechnologie wichtiger ist. Die Produktionskosten dürfen nicht direkt als zu minimierendes Ziel verwendet werden, da sonst die in Hinblick auf die Produktionskosten optimale Lösung eine Stilllegung der Fertigung wäre. Stattdessen werden Opportunitätskosten berechnet, wenn ein Auftrag nicht mit der günstigsten Fertigungstechnologie hergestellt wird, die für diesen Artikel zur Verfügung steht. Die Summe dieser Opportunitätskosten wird über alle Aufträge des Betrachtungszeitraums gebildet und stellt das Maß der Zielerreichung hinsichtlich der Produktionskosten dar. Hier muss darauf hingewiesen werden, dass im Rahmen dieser Arbeit der Begriff Produktionskosten nur die Kosten für die tatsächliche Produktion auf der Maschine sowie die Kosten für die Herstellung oder Beschaffung des benötigten Vormaterials umfasst. Insbesondere die Rüstkosten, die typischerweise zu den Produktionskosten gehören, sind ausgenommen und werden gesondert behandelt.

3.3.3 Zielfunktion und Selektionsmethoden

Die Berücksichtigung mehrerer unabhängiger und teilweise konkurrierender Ziele in einem Optimierungsproblem bringt zusätzliche Herausforderungen mit sich. Diese treten insbesondere im Bereich der Lösungsbewertung auf. Bei Optimierungsproblemen mit einem einzigen Ziel ist der Vergleich zweier Lösungen einfach. Entweder beide Lösungen erreichen das Ziel gleich gut oder eine der beiden ist besser. Sobald mehr als ein Ziel berücksichtigt wird, ist die Situation nicht mehr so eindeutig, wie nachfolgend am Beispiel zweier Lösungen mit je zwei zu maximierenden Zielwerten gezeigt wird. Die Werte sind in Tabelle 3 enthalten. Fall 1 ist insofern trivial, als beide Lösungen bei beiden Zielwerten das gleiche Ergebnis erzielen und damit gleich gut sind. In Fall 2 erzielt Lösung A bei beiden Zielwerten ein besseres Ergebnis. Damit ist klar, dass Lösung A besser ist als Lösung B. Man sagt A *dominiert* B. Bei Fall 3 kann ohne Zusatzinformationen keine Entscheidung über die optimale Lösung getroffen werden, da beide Lösungen bei einem Zielwert das bessere Ergebnis erreichen. Der schwächere der beiden Zielwerte ist bei Lösung A höher, aber wenn der zweite Wert wichtiger ist als der erste, kann dennoch Lösung B besser sein. Die gleichen Fälle treten auch auf, wenn mehr Lösungen oder Zielwerte betrachtet werden. Bei einer größeren Menge von Lösungen ist die Unterscheidung in *dominierte* und *nicht-dominierte* Lösungen von Interesse. Wie in Fall 2 des Beispiels, gilt eine Lösung als dominiert, wenn es zumindest eine weitere Lösung gibt, die bei keinem Zielwert schlechter ist und zumindest bei einem Zielwert besser ist. Für nicht-dominierte Lösungen gibt es keine andere Lösung, für die das gilt. In Fall 2 ist

Tabelle 3: Beispiel 2 Zielwerte

	Lösung A	Lösung B
Fall 1:	{1;2}	{1;2}
Fall 2:	{3;3}	{1;2}
Fall 3:	{3;2}	{1;3}

Lösung A eine nicht-dominierte Lösung. In Fall 3 sind beide Lösungen nicht-dominiert. Es ist offensichtlich, dass bei einer Menge von Lösungen die optimale Lösung nur aus der Teilmenge der nicht-dominierten Lösungen stammen kann. Dieses Konzept kann auf die Menge aller möglichen Lösungen eines multikriteriellen Optimierungsproblems erweitert werden. Die Menge der nicht-dominierten Lösungen des gesamten Lösungsraums wird als *Pareto-optimale Front* bezeichnet, die darin enthaltenen Lösungen als *Pareto-optimale Lösungen*. Die optimale Lösung eines multikriteriellen Optimierungsproblems muss also eine Pareto-optimale Lösung sein. Daher spielt die Suche nach diesen Lösungen eine wichtige Rolle in der multikriteriellen Optimierung [7]. Es sei darauf hingewiesen, dass dies nicht bedeutet, dass jede nicht-dominierte Lösung besser ist, als jede dominierte Lösung. Gegeben seien die Lösungen $A=\{5;5\}$, $B=\{4;4\}$ und $C=\{1;6\}$. A und C sind nicht-dominierte Lösungen, aber unter der Annahme, dass der erste Wert wichtiger ist als der zweite, ist Lösung B wahrscheinlich besser als Lösung C.

Wie bereits in Fall 3 des ersten Beispiels gezeigt, kann zwischen nicht-dominierten Lösungen ohne zusätzliche Information nicht darüber entschieden werden, welche Lösung besser ist. Eine Möglichkeit ist, nicht-dominierte Lösungen zu suchen und nach einem zu definierenden Verfahren aus dieser Menge einige Lösungen auszuwählen. Denkbar wäre beispielsweise, die Lösungen so zu bestimmen, dass die Hamming-Distanz [23] zwischen den Lösungen maximiert wird. So wird ein möglichst großer Teil des Lösungsraums abgedeckt und es werden verschiedenartige Lösungen ausgewählt. Diese Lösungen können einem Experten vorgelegt werden, der anhand seines Wissens die beste Lösung identifiziert. Problematisch an diesem Vorgehen ist, dass durch die Auswahl nicht garantiert werden kann, dass die beste gefundene Lösung dem Experten vorgelegt wird. Wenn alle nicht-dominierten Lösungen vorgelegt werden, ist deren Anzahl in praktischen Anwendungen oft zu groß um eine sinnvolle Unterstützung darzustellen. Außerdem wird von computergestützten Verfahren oft erwartet, dass sie dem Experten die Entscheidung

abnehmen. Bei GA müssen in jeder Generation Lösungen bewertet werden, um die Individuen zu selektieren, die ihre Eigenschaften an die nächste Generation weitergeben dürfen. Diese Aufgabe kann kein Experte übernehmen, sodass andere Möglichkeiten gefunden werden müssen. Es gibt verschiedene Methoden um ein multikriterielles Problem als einkriterielles Problem zu formulieren [7]. Geläufig ist die Methode der gewichteten Summe, deren Implementierung für den GA zur kombinierten Auftragszeugung, Auftragsfreigabe und Arbeitsverteilung bei reihenfolgeabhängigen Rüstzeiten nachfolgend beschrieben wird. Alternativ kann für den Selektions-Operator des GA auch auf die klassische Lösungsbewertung verzichtet werden und eine Selektion nach Dominanzebenen durchgeführt werden. Die Beschreibung dieses Verfahrens bildet den Abschluss dieses Abschnitts.

Methode der gewichteten Summe

Bei diesem Verfahren werden die Einzelziele gewichtet und die erreichten Werte aufsummiert. Je nach Art des Optimierungsproblems wird diese Summe minimiert oder maximiert. Für das vorliegende Problem erscheint die Methode wegen der bereits integrierten Möglichkeit der Gewichtung von Einzelzielen geeignet. Wie in Kapitel 2.4 beschrieben, muss beeinflussbar sein, auf welchen Zielen der Fokus der Optimierung liegt. Durch die Gewichtung des Einflusses der Einzelziele auf die Zielfunktion ist dies direkt möglich. Allerdings stellt sich das Problem, dass die Einzelziele zuvor normiert werden müssen. Ohne Normierung wird die Gewichtung durch die Einheit beeinflusst, in der die Einzelziele bewertet werden. Beispielsweise wird die Rüstzeit in Millisekunden gemessen und die Opportunitätskosten für die Produktion mit einer anderen als der günstigsten Fertigungstechnologie in Euro. Ohne Normierung wären zehn Minuten Rüstzeitreduktion 600000 Euro wert und die Reduktion von Produktionskosten würde praktisch keine Rolle spielen. Die klassische Min/Max Normierung [14] ist in diesem Fall jedoch als Normierungsverfahren ungeeignet, weil die Einzelziele unterschiedlich stark streuen und der Wertebereich um ein Vielfaches größer ist, als der Bereich, der für gute Lösungen in Frage kommt. Es kann beispielsweise ein Produktionsprogramm erstellt werden, demzufolge kein einziger Auftrag produziert wird. Abhängig von Anfangslagerbestand und Bedarfssituation weist diese Lösung wahrscheinlich sehr hohe Fehlmengen auf. Aufgrund solcher Ausreißer muss der Zielwert Fehlmengen für einen sehr großen Wertebereich normiert werden. Für die Opportunitätskosten der Produktion ist der Wertebereich deutlich kleiner. Da ein hoher Servicegrad angestrebt wird, ist eine Lösung mit geringeren Fehlmengen meist zu bevorzugen. Gegeben seien etwa zwei Lösungen A und

B, wobei A deutlich geringere Fehlmengen, aber etwas höhere Opportunitätskosten aufweist. Durch die unterschiedlichen Wertebereiche wird ohne Gewichtung möglicherweise B besser bewertet, obwohl aufgrund der Fehlmengen sicher A zu bevorzugen wäre. Das gleiche Problem tritt auf, wenn die oberen und unteren Schranken des Wertebereichs für die Min/Max-Normierung nicht im Vorhinein global festgelegt werden, sondern für jeden Vergleich einer Menge von Lösungen der höchste und der niedrigste in der Menge vorkommende Wert des jeweiligen Einzelziels ermittelt werden. Jeder Ausreißer in der zu vergleichenden Menge verzerrt die Gewichtung. Im Unterschied zur globalen Festlegung der Grenzen ist in diesem Fall die Verzerrung nicht im Vorhinein bekannt. Bei globaler Festlegung der Grenzen kann die Verzerrung durch geschickte Wahl der Gewichtung ausgeglichen werden. Dafür ist aber keine vorhergehende Min/Max-Normierung notwendig. Dieser Ansatz wird in der Implementierung gewählt. Da die Vermeidung von Fehlmengen sehr wichtig ist, werden die anderen Einzelziele so gewichtet, dass eine Einheit jedes Ziels in etwa einer Einheit der Fehlmengen entspricht (Einheit der Zielwerte: siehe Abschnitt 3.3.2). Die gewählten Gewichtungsfaktoren und die Gründe für die Festlegung in der jeweiligen Höhe werden in Tabelle 4 gezeigt. Diese Werte werden als Standardgewichtung betrachtet. Der Fokus der Optimierung kann verändert werden, indem die Standardgewichtungen mit entsprechenden Faktoren multipliziert werden. Die gewichtete Summe wird für jede Lösung berechnet. Dadurch wird aus dem multikriteriellen Problem ein einkriterielles. Für den GA bedeutet das, dass die üblichen Selektionsverfahren eingesetzt werden können. Implementiert wurden die Fitnessproportionale Selektion und die Rangbasierte Selektion, wobei keine signifikanten Unterschiede festgestellt wurden.

Tabelle 4: Gewichtungsfaktoren

Zielgröße	Gewichtung	Begründung
Rüstzeit	0,001	1 sec Rüstzeit $\hat{=}$ 1 Fehltehtag
Fehlmengen	1	Bezugsgröße
Bestand	0,1	10 Euro Lagerbestandswert $\hat{=}$ 1 Fehlteitag
Kapazitätsauslastung	0,0333	30 Stk. (\approx 1 min) Auslastung $\hat{=}$ 1 Fehlteitag
Produktionskosten	2	0,5 Euro Produktionskosten $\hat{=}$ 1 Fehlteitag

Selektion nach Dominanzebenen

Die Schwierigkeiten bei der Überführung des multikriteriellen Problems in ein einkriterielles Problem können zumindest in Hinblick auf die Selektion umgangen werden, indem ein Selektionsverfahren eingesetzt wird, das nicht vom direkten Vergleich von Lösungen abhängt. Ein solches Verfahren ist die Selektion nach Dominanzebenen [6]. Bei dieser Methode werden die Lösungen einer Menge M_{Loes} zuerst in Dominanzebenen eingeteilt. In der ersten Dominanzebene sind die nicht-dominierten Lösungen aus M_{Loes} enthalten. Jede weitere Dominanzebene enthält alle Lösungen, die nur von Lösungen aus den höheren Dominanzebenen dominiert werden. Das Verfahren wird fortgesetzt, bis jede Lösung einer Dominanzebene zugeordnet ist. Für die effiziente Berechnung der Dominanzebenen sei auf Deb [7] verwiesen. Beginnend mit der ersten Dominanzebene wählt das Selektionsverfahren alle Lösungen einer Dominanzebene, bis so viele Lösungen wie gewünscht selektiert sind. Werden nicht alle Lösungen einer Dominanzebene selektiert, erfolgt unter diesen eine Auswahl, sodass die Hamming-Distanz zwischen den gewählten Lösungen maximal ist. Dadurch soll eine Diversifikation der Selektion erzielt werden. Das Verfahren gilt als elitär. Allerdings sei wiederum darauf hingewiesen, dass eine Lösung nicht unbedingt besser ist, weil sie zu einer höheren Dominanzebene gehört. Bei diesem Selektionsverfahren ist nicht eindeutig, welche die beste Lösung einer Generation ist. Um zum Abschluss des Verfahrens eine einzige Lösung als Ergebnis zu präsentieren, müssen zusätzliche Informationen berücksichtigt werden. In der Implementierung wird dies durch die Methode der gewichteten Summe erreicht.

3.3.4 Mutationsoperatoren

Die Testergebnisse zeigen, dass die Mutationsoperatoren für die Lösungsqualität von entscheidender Bedeutung sind. Diese Ergebnisse und die Diskussion, warum die Mutation so wichtig ist, werden in Kapitel 4.3 präsentiert. Im Folgenden werden lediglich die Mutationsoperatoren gezeigt, die für das Verfahren implementiert wurden. Im Vergleich zu den Mutationsoperatoren für Probleme mit Binärcodierung sind die hier gezeigten Operatoren deutlich komplexer, da verschiedene Arten von Information vorhanden sind, die von der Mutation betroffen sein können. Neben den klassischen Mutationsoperatoren, die zufällige Änderungen an den Individuen hervorrufen, werden auch Mutationsoperatoren gezeigt, die gezielt Veränderungen vornehmen. Auch bei diesen ist ein Zufallsfaktor enthalten, aber ihr Einsatz zielt auf bestimmte Verbesserungen, wie z.B. Reduktion der Fehlmengen, ab. Dies wird nachfolgend als *gezielte Mutation* bezeichnet.

Mit diesen Operatoren wird versucht, die Techniken, die der Disponent anwendet, um eine gute Produktionsplanung zu erstellen, auf den GA zu übertragen. Dabei besteht eine gewisse Ähnlichkeit zu *Scatter Search*, einer Heuristik aus der Klasse der evolutionären Algorithmen, die ebenfalls versucht, zusätzliche Informationen über den Lösungsraum zu integrieren, um in jeder Generation gezielt bessere Lösungen zu finden [16]. Es folgen zuerst die klassischen Mutations-Operatoren (Algorithmen 1, 2 und 3) danach die für die gezielte Mutation (Algorithmen 4, 5, 6 und 7).

Algorithmus 1 : Mutation Maschinenzuordnung

Schritt 1:

Bestimme anhand der Mutationswahrscheinlichkeit für Individuen, welche Individuen mutieren.

Schritt 2:

Für jedes in Schritt 1 ausgewählte Individuum: Bestimme anhand der Mutationswahrscheinlichkeit für Gene welche Gene mutieren.

Schritt 3:

Für jedes in Schritt 2 ausgewählte Gen: Bestimme nach dem Zufallsprinzip, auf welche Maschine Aufträge verschoben werden sollen. In Frage kommen alle Maschinen auf denen der Artikel des Gens gefertigt werden kann. Mit einer definierten Wahrscheinlichkeit P_{SwapAll} werden alle Aufträge des Gens auf diese Maschine verschoben, sonst werden alle Aufträge eines zufällig ausgewählten Produktionsintervalls verschoben.

Algorithmus 2 : Mutation Produktionsmenge

Schritt 1:

Wie Schritt 1 von Algorithmus 1

Schritt 2:

Wie Schritt 2 von Algorithmus 1

Schritt 3:

Für jedes in Schritt 2 ausgewählte Gen: Bestimme anhand der Mutationswahrscheinlichkeit für Fertigungsaufträge, welche Fertigungsaufträge mutieren. Bestimme nach dem Zufallsprinzip die Höhe einer Mengenänderung im Intervall $[-\text{Mindestlosgröße}; \text{Mindestlosgröße}]$. Bei allen ausgewählten Aufträgen des Gens wird die Produktionsmenge entsprechend der Mengenänderung angepasst.

Schritt 4:

Nach der Mengenänderung müssen gegebenenfalls Fertigungsaufträge zusammengefasst werden, um die Mindestlosgröße nicht zu unterschreiten. Evt. können Fertigungsaufträge auch geteilt werden, um Aufträge mit sehr großen Produktionsmengen zu verhindern.

Algorithmus 3 : Mutation Produktionsintervall

Schritt 1:

Wie Schritt 1 von Algorithmus 1

Schritt 2:

Wie Schritt 2 von Algorithmus 1

Schritt 3:

Für jedes in Schritt 2 ausgewählte Gen: Bestimme anhand der Mutationswahrscheinlichkeit für Fertigungsaufträge, welche Fertigungsaufträge mutieren. Alle ausgewählten Aufträge des Gens werden in dasselbe, zufällig bestimmte Produktionsintervall verschoben.

Algorithmus 4 : Gezielte Mutation - Fehlmengenreduktion durch zusätzliche Fertigungsaufträge

Versucht durch Hinzufügen von Aufträgen für Artikel, die hohe Fehlmengen verursachen, eine Lösungsverbesserung zu erzielen.

Schritt 1:

Wähle zu mutierende Individuen aus. Es können z.B. gute Lösungen gewählt werden, die noch weiter verbessert werden sollen oder solche mit grundsätzlich guten Eigenschaften, aber zu hohen Fehlmengen.

Schritt 2:

Für jedes in Schritt 1 ausgewählte Individuum: Bestimme die Artikel, die die höchsten Fehlmengen verursachen. Die Anzahl der Artikel wird durch eine festzulegende Konstante $NUMB_{(Art-MissPartReduct)}$ bestimmt.

Schritt 3:

*Für jeden in Schritt 2 bestimmten Artikel: Wähle das Produktionsintervall aus, dem Aufträge hinzugefügt werden. Es kommen alle Produktionsintervalle in Frage, in denen Fehlmengen auftreten. Unter diesen wird zufällig das Intervall bestimmt, dem Aufträge hinzugefügt werden. Die ersten Produktionsintervalle werden mit höherer Wahrscheinlichkeit gewählt. Bestimme zufällig die Maschine, auf der die zusätzlichen Aufträge produziert werden sollen. Erstelle einen neuen Fertigungsauftrag mit zufälliger Produktionsmenge aus dem Intervall $[Mindestlosgröße; 2 * Mindestlosgröße]$ und füge ihn der Auftragsliste des Gens hinzu, das durch den Artikel und die zufällig bestimmte Maschine definiert ist.*

Algorithmus 5 : Gezielte Mutation - Hinzufügen von Aufträgen für Schnelldreher

Versucht durch Hinzufügen von Aufträgen für Schnelldreher in den ersten beiden Produktionsintervallen eine Verbesserung der Lösung herbeizuführen. Dies kann zu Rüstzeitreduktion und Verbesserung der Kapazitätsauslastung führen. Auch eine Bestandsreduktion ist möglich, wenn durch das Vorziehen von Schnelldrehern die Produktion von Langsamdrehern näher an das Bedarfsdatum rückt.

Schritt 1:

Wähle zu mutierende Individuen aus. Es können z.B. gute Lösungen gewählt werden, die noch weiter verbessert werden sollen oder solche mit grundsätzlich guten Eigenschaften, aber zu hoher Rüstzeit oder zu niedriger Kapazitätsauslastung.

Schritt 2:

Für jedes in Schritt 1 ausgewählte Individuum: Wähle die Gene aus, die mutieren. Es kommen nur Gene in Frage die Schnelldreher repräsentieren und Fertigungsaufträge für eines der ersten beiden Produktionsintervalle beinhalten. Aus den in Frage kommenden Genen werden anhand der Mutationswahrscheinlichkeit $P_{(TargetMut-AddFastmover)}$ die Gene bestimmt, die mutieren.

Schritt 3:

*Für jedes in Schritt 2 ausgewählte Gen erstelle einen neuen Fertigungsauftrag mit zufälliger Produktionsmenge aus dem Intervall $[Mindestlosgröße; 2 * Mindestlosgröße]$. Füge den neuen Auftrag einem der beiden ersten Produktionsintervalle hinzu. Der Auftrag darf nur einem Produktionsintervall hinzugefügt werden in dem der entsprechende Artikel bereits produziert wird.*

Algorithmus 6 : Gezielte Mutation - Fehlmengenreduktion durch Vorziehen von Fertigungsaufträgen

Versucht durch das Vorziehen von Aufträgen für Artikel, die hohe Fehlmengen verursachen, eine Lösungsverbesserung zu erzielen.

Schritt 1:

Wie Schritt 1 von Algorithmus 4

Schritt 2:

Für jedes in Schritt 1 ausgewählte Individuum: Bestimme die Artikel, die die höchsten Fehlmengen verursachen. Die Anzahl der Artikel wird durch eine festzulegende Konstante $NUMB_{(Art-MissPartReduct)}$ bestimmt. Die so gewählten Artikel werden nach den Fehlmengen sortiert und es werden entsprechend der Reihenfolge Ränge zugeordnet. Durch Rangbasierte Selektion wird ein Artikel a ausgewählt.

Schritt 3:

Für den in Schritt 2 bestimmten Artikel: Wähle das Produktionsintervall $prodInt$ aus, in das Aufträge vorgezogen werden. Es kommen alle Produktionsintervalle in Frage, in denen Fehlmengen auftreten. Unter diesen wird zufällig das Intervall bestimmt, in das Aufträge vorgezogen werden. Die ersten Produktionsintervalle werden mit höherer Wahrscheinlichkeit gewählt.

Schritt 4:

Bestimme die Maschine m , auf der die vorgezogenen Aufträge produziert werden. Aus allen Maschinen auf denen der Artikel gefertigt werden kann, wird eine Maschine durch Rangbasierte Selektion ausgewählt. Die Ränge ergeben sich aus der Summe der Verspätungen von Aufträgen auf der Maschine im betroffenen Produktionsintervall. Je höher die Verspätung, umso höher der Rang

Schritt 5:

Für alle Aufträge für a , die nach $prodInt$ auf einer beliebigen Maschine produziert werden: Bestimme anhand der Mutationswahrscheinlichkeit $P_{(TargetMut-MissPartReduct-Prep)}$ welche Aufträge vorgezogen werden. Anmerkung zum konkreten Anwendungsfall: Laut ERP-System bereits freigegebene Aufträge dürfen nicht auf eine andere Maschine verschoben werden und kommen daher nur in Frage, wenn sie auf m produziert werden.

Schritt 6:

Für jeden in Schritt 5 bestimmten Auftrag: Verschiebe den Auftrag in das Produktionsintervall $prodInt$.

Algorithmus 7 : Gezielte Mutation - Rüstzeitreduktion

Versucht durch Zusammenfassung von Aufträgen aufeinanderfolgender Produktionsintervalle zu einem Produktionsblock, die Rüstzeit zu reduzieren.

Schritt 1:

Wähle zu mutierende Individuen aus. Es können z.B. gute Lösungen gewählt werden, die noch weiter verbessert werden sollen oder solche mit grundsätzlich guten Eigenschaften, aber zu hoher Rüstzeit.

Schritt 2:

Für jedes in Schritt 1 ausgewählte Individuum: Bestimme für jedes Gen ob es mutiert. Dies ist abhängig von der Mutationswahrscheinlichkeit

$P_{(TargetMut-SetupReduct-Gen)}$ und davon, ob eine Rüstzeitreduktion durch Zusammenfassung von Fertigungsaufträgen in einem Produktionsintervall möglich ist. Eine Zusammenführung gilt als möglich, wenn der gleiche Artikel auf der Maschine des Gens in zwei aufeinanderfolgenden Produktionsintervallen produziert wird.

Schritt 3:

Für jedes in Schritt 2 ausgewählte Gen: Fasse die Aufträge zweier aufeinanderfolgender Produktionsintervalle zusammen. In welchem der beiden Intervalle die Aufträge zusammengefasst werden, kann davon abhängen, ob alle Aufträge in ein anderes Produktionsintervall verschoben werden dürfen. Auch die Auslastung der Produktionsintervalle kann berücksichtigt werden.

3.3.5 Initialisierung, Kombinationen und Optimierung je Maschine

Die Initialisierung des Verfahrens mit Ausgangslösungen und die Kombination ausgewählter Lösungen mit dem Ziel, diese ihre guten Eigenschaften an die nächste Lösungsgeneration weitergeben zu lassen, sind integrale Bestandteile jedes GA. Der Vollständigkeit halber wird ihre Umsetzung geschildert, der mit der Materie vertraute Leser wird jedoch nur wenig Neues oder Unerwartetes entdecken. Daher wird die Behandlung dieser Punkte kurz gehalten. Den Abschluss dieses Kapitels über den GA bildet eine Beschreibung, wie das Verfahren angepasst werden kann, um statt des Produktionsprogramms für die gesamte Fertigung, die Belegung der einzelnen Maschinen zu optimieren und wie die Ergebnisse dieser lokalen Optimierung in die Lösung des Gesamtproblems integriert werden können.

Die Initialisierungsverfahren sollen Lösungen brauchbarer Qualität liefern, die aber verschiedene Eigenschaften aufweisen, um mit einem diversifizierten Gen-Pool zu starten. Bei dem vorliegenden Problem ist bekannt, dass sich die im Betrachtungszeitraum produzierte Menge je Artikel bei guten Lösungen am auftretenden Bedarf orientieren muss. Aus diesem Grund erstellen die implementierten Initialisierungsverfahren für jeden Artikel Aufträge im entsprechenden Umfang. Eine Diversifizierung in der Dimension der Produktionsmenge wird erreicht, indem bei jeder Lösung, für jeden Artikel per Zufallsprinzip eine Abweichung der Produktionsmenge von der Bedarfsmenge bestimmt wird. In welchem Bereich diese Abweichung liegen darf, ist im Vorhinein festzulegen. Die weitere Diversifizierung wird durch die zeitliche Verteilung der Aufträge erreicht. Wie in Abschnitt 3.3.1 beschrieben, muss jeder Auftrag einem Produktionsintervall zugewiesen werden. Die Initialisierung erfolgt ohne Berücksichtigung des Bedarfsverlaufs, daher werden die Fertigungsaufträge den Produktionsintervallen per Zufallsprinzip zugewiesen. Der wesentliche Unterschied zwischen den Initialisierungsverfahren ist, ob die Zuweisung der Aufträge zu einem Produktionsintervall unabhängig voneinander geschieht oder ob Aufträge für den gleichen Artikel in Gruppen demselben Produktionsintervall zugewiesen werden, um Auftragsblöcke zu bilden. Durch die Kombination beider Verfahren sind sowohl die Eigenschaft Blockbildung, als auch die Eigenschaft einen Artikel zu verschiedenen Zeitpunkten zu produzieren, in der Ausgangspopulation enthalten. Die Maschinenzuweisung erfolgt je Ausgangslösung, für alle Aufträge eines Artikels, auf die gleiche Maschine. Dabei wird eine der Maschinen gewählt, deren Fertigungsverfahren

für den Artikel die niedrigste Versionsnummer aufweisen. Diese Bevorzugung niedriger Versionsnummern wird in Abschnitt 3.4.2 begründet. Wenn nach diesem Kriterium mehrere Maschinen in Frage kommen, wird zufällig eine ausgewählt. Da diese Auswahl für jede initialisierte Lösung neu getroffen wird, besteht auch eine Diversifikation der Maschinenzuordnung.

Die Kombinationsoperatoren sind einfache Abwandlungen des Two-Point-Crossover (TPX)-Verfahrens [28]. Wie in Abschnitt 3.3.1 beschrieben, steht das Gen, welches eine bestimmte Maschinen-Artikel-Kombination repräsentiert, in jedem Individuum an der gleichen Stelle. Aufgrund der gewählten Codierung können Aufträge zwischen Genen mit gleicher Maschinen-Artikel-Kombination beliebig verschoben werden, ohne ungültige Lösungen zu erzeugen. Für die Anwendung des TPX-Verfahrens müssen daher nur Kreuzungspunkte bestimmt werden, dann kann der Inhalt der Gene, die zwischen diesen Kreuzungspunkten liegen, ausgetauscht werden. Neben der zufälligen Bestimmung von Kreuzungspunkten können diese auch so gewählt werden, dass die vollständige Planung einer Maschine ausgetauscht wird. Das ist möglich, weil die Gene so sortiert sind, dass alle Gene, die eine Maschine betreffen, in einem Block stehen.

Aus dem gleichen Grund ist auch die Anpassung des GA für die Optimierung einer einzelnen Maschine ohne großen Aufwand möglich. Die genetischen Operatoren behandeln in diesem Fall nur den Gen-Block, der die jeweilige Maschine betrifft. Es müssen einige kleine Anpassungen durchgeführt werden, damit die Methoden nur den relevanten Bereich bearbeiten und die Kreuzungslänge bei der Kombination nach dem TPX-Verfahren muss reduziert werden, aber sonst können die Methoden des GA für die gesamte Fertigung, übernommen werden. Auch die Lösungsbewertung erfolgt weiterhin für die gesamte Fertigung. Da ausschließlich an den Genen einer Maschine Änderungen vorgenommen werden, wird auf diese Art mit den Methoden des GA die Planung einer Maschine unter der Annahme optimiert, dass die Planung der weiteren Maschinen in der Fertigung nicht verändert wird. Es sei darauf hingewiesen, dass durch diese Art der Lösungsbewertung viele unnötige Berechnungen durchgeführt werden müssen. Durch geschicktere Implementierung kann der Aufwand für die Lösungsbewertung reduziert werden. Jede Optimierung einer Maschinenbelegung stellt dabei eine lokale Suche dar. Da der Lösungsraum für das

lokale Problem deutlich kleiner ist, können mit dieser Methode deutliche Verbesserungen erzielt werden (siehe Kapitel 4). Die Schwierigkeit besteht darin, die Ergebnisse der lokalen Suchen so zu integrieren, dass eine Verbesserung für die Planung der gesamten Fertigung erzielt wird. In der Implementierung wird dieses Problem gelöst, indem die Ergebnisse der lokalen Suchen so kombiniert werden, dass in den Kombinationslösungen für jede Maschine Gene verwendet werden, die aus einem für diese Maschine optimierten Ergebnis der lokalen Suche stammen. Das dafür verwendete Verfahren wird nachfolgend als Algorithmus 8 beschrieben. Die Anzahl möglicher Kombinationslösungen entspricht dem Produkt der Anzahl von lokalen Lösungen je Maschine, die für die Kombination zugelassen werden. Um den Rechenaufwand für die Lösungsbewertung in Grenzen zu halten, wird die Anzahl der erlaubten Kombinationslösungen begrenzt.

Algorithmus 8 : Kombination von Ergebnissen der lokalen Suche

Schritt 1:

Wähle das beste Ergebnis der lokalen Suche für jede Maschine, entferne es aus der Liste der Ergebnisse der lokalen Suche und füge es zur Menge $L_{Basis-Kombination}$ hinzu.

Schritt 2:

Entferne die Lösung mit dem besten Zielfunktionswert aus den Ergebnissen der lokalen Suche und prüfe, ob die Lösung zu $L_{Basis-Kombination}$ hinzugefügt werden kann ohne die erlaubte Anzahl von Kombinationslösungen zu überschreiten. Wenn möglich füge die Lösung hinzu. Wiederhole Schritt 2, bis die Menge der Ergebnisse der lokalen Suche leer ist, oder für keine Maschine mehr ein Ergebnis in $L_{Basis-Kombination}$ eingefügt werden kann, ohne die Anzahl erlaubter Kombinationslösungen zu überschreiten.

Schritt 3:

Erzeuge nach dem oben beschriebenen Verfahren alle Kombinationslösungen aus den Lösungen in $L_{Basis-Kombination}$ und bewerte diese. Eine definierte Anzahl der besten Kombinationslösungen wird an den GA für die Optimierung der Gesamtfertigung zurückgegeben.

3.4 Auftragserzeugung und Auftragsfreigabe durch KRAF

Die Kapazitäts- und Rüstorientierte Auftragsfreigabe (KRAF) ist ein heuristisches Verfahren, das auf Korma [25] aufbaut und diese Methode um die Funktionen Arbeitsverteilung und Rüstoptimierung erweitert. Das bei der Entwicklung dieser Heuristik verfolgte Ziel ist eine Reduktion der Komplexität gegenüber dem GA. Dieses Ziel wird in erster Linie durch vier Faktoren erreicht.

1. Die Prozesse Auftragserzeugung und Auftragsfreigabe werden konzeptuell getrennt.
2. Das Produktionsprogramm wird nur für Schichten berechnet, für deren Kapazitäten Fertigungsaufträge freigegeben werden sollen. Beim GA wird ein Produktionsprogramm für den gesamten Betrachtungszeitraum definiert.
3. Für einzelne Fertigungsaufträge oder für Gruppen von Fertigungsaufträgen wird die Freigabeentscheidung getroffen. Ist diese für einen Auftrag positiv, wird sie nicht mehr zurückgenommen.
4. Zufallseinflüsse werden vermieden.

Die folgenden Abschnitte beschreiben die einzelnen Teile des Verfahrens. Für Verfahrensparameter wird jeweils bei der ersten Erwähnung des Parameters der für die Implementierung gewählte Wert in Klammern angeführt.

3.4.1 Auftragserzeugung, Terminierung

Der erste Schritt ist die Definition sämtlicher Fertigungsaufträge, die im Betrachtungszeitraum theoretisch eingeplant werden können. Um die Datenkonsistenz zu gewährleisten, sind dies zuerst alle, laut ERP-System bereits freigegebenen, Fertigungsaufträge. Weiter werden für jeden Artikel so viele Fertigungsaufträge mit Mindestlosgröße erzeugt, dass am Ende des Betrachtungszeitraums der Sicherheitsbestand nicht unterschritten wird. Dieses Verfahren wird nachfolgend als Algorithmus 9 beschrieben. Die neu erzeugten Aufträge sind noch keiner Maschine zugeordnet. Es wird jedoch vereinfachend angenommen, dass die Fertigungsaufträge eines Artikels in der Reihenfolge produziert werden, in welcher sie in der zugehörigen Liste des Ergebnisses von Algorithmus 9 stehen. Unter Berücksichtigung des Anfangslagerbestands und des Bedarfsverlaufs kann so für jeden Fertigungsauftrag das Datum bestimmt werden, zu dem dieser fertiggestellt sein muss, damit der Lagerstand des entsprechenden Artikels den Sicherheitsbestand nicht unterschreitet. Gleiches gilt für das Datum zu dem der Lagerstand unter 0 fallen

würde, also ein Bedarf nicht bedient werden könnte. Diese Daten werden als SB-Datum und NB-Datum für jeden Fertigungsauftrag gespeichert. Da eine Unterschreitung des Sicherheitsbestandes vermieden werden soll, muss jeder Fertigungsauftrag FE zu einem Zeitpunkt freigegeben werden, der um $Produktionszeit_{FE} + Uebergangszeit$ vor $SB - Datum_{FE}$ liegt. Die $Uebergangszeit$ ist die Zeitspanne zwischen Auftragsfreigabe und Produktionsstart, und a priori nicht bekannt. Daher muss ein für die jeweilige Fertigung typischer Wert gewählt werden (Implementierung: 3 Tage). Das berechnete Datum wird für jeden Fertigungsauftrag als $Plan - Freigabedatum$ gespeichert.

Algorithmus 9 : Auftragserzeugung für KRAF

Result : Eine Liste P , die für jeden Artikel mat eine Liste P_{mat} mit allen im Betrachtungszeitraum möglichen Fertigungsaufträgen für mat enthält.

Schritt 1:

Erzeuge eine leere Liste P und führe die Schritte 2 bis 5 für jeden zu planenden Artikel mat durch.

Schritt 2:

Erzeuge eine leere Liste P_{mat} und füge alle Fertigungsaufträge für mat hinzu, die laut ERP-System aktuell freigegeben sind.

Schritt 3:

Beschaffe folgende Informationen, jeweils bezüglich des Artikels mat :

- *Summe der Auftragsmengen aller laut ERP-System freigegebenen Fertigungsaufträge ($mengeFrei_{mat}$);*
- *Bedarf im Betrachtungszeitraum ($bedarf_{mat}$);*
- *Sicherheitsbestand (sb_{mat});*
- *Mindestlosgröße ($minLosgroe_{mat}$);*

Berechne die Anzahl neu zu erzeugender Aufträge für mat ($anzAuf_{mat}$) nach der Formel $\lceil \frac{bedarf_{mat} - mengeFrei_{mat} + sb_{mat}}{minLosgroe_{mat}} \rceil$

Schritt 4:

Erzeuge $anzAuf_{mat}$ neue Fertigungsaufträge mit Mindestlosgröße und füge diese zu P_{mat} hinzu.

Schritt 5:

Füge P_{mat} zu P hinzu.

Wie in Abschnitt 3.1 erläutert, muss das Verfahren den Aufbau von Lagerbestand ermöglichen, um die Aufteilung der Produktionsmenge bei sehr großen Bestellungen abzubilden. Dies geschieht über eine entsprechende Umterminierung von Fertigungsaufträgen. Nach der zuvor beschriebenen Terminierung ist den Aufträgen, die für die Bedienung einer übergroßen Bestellung vorgesehen sind, als SB-Datum das Bedarfsdatum der Bestellung zugewiesen. Das Verfahren zur Umterminierung von Fertigungsaufträgen wird in Algorithmus 10 beschrieben. Für das Verfahren werden folgende Parameter benötigt:

maxTagFrüher; maxTageSpäter

Die maximale Anzahl von Tagen, die ein Fertigungsauftrag vor/nach dem Bedarfstermin einer großen Bestellung liegen darf, damit er dieser Bestellung zugeordnet werden kann (Implementierung: *maxTagFrüher* : 3; *maxTageSpäter* : 1). Die Festlegung dieser Grenzen ist notwendig, da sonst bei teilweise bereits erfolgtem Bestandsaufbau Fertigungsaufträge zu früh freigegeben werden.

prodZyklusLaenge

Die Länge der Produktionszyklen, auf die die Fertigungsaufträge verteilt werden. Implementierung: Eine Woche, da die Mengen in der Praxis wochenweise aufgeteilt werden.

tPuffer

Zeitpuffer, um den die Produktion vor dem Bedarfstermin abgeschlossen sein soll. Kann sinnvoll sein, wenn durch eine Verspätung hohe Kosten entstehen, z.B. weil für die übergroße Bestellung ein Sondertransport notwendig ist, der nur zum Bedarfsdatum zur Verfügung steht.

Die durch das Verfahren neu berechneten Werte für SB-Datum und NB-Datum zeigen nicht die tatsächlichen Zeitpunkte der Unterschreitung des jeweiligen Bestandes an, sondern stellen ein Maß für die Dringlichkeit des Auftrags dar. Das größere Zeitfenster zwischen Plan-Freigabedatum und SB-Datum bei weiter nach vor terminierten Fertigungsaufträgen führt dazu, dass diese über einen längeren Zeitraum freigegeben, aber noch nicht besonders dringlich sind. Sie können daher eher rüsto optimal eingeplant werden oder der Glättung des Bedarfsverlaufs dienen. Der Grund dafür ist die Art der Auftragsauswahl für die Auftragsfreigabe, die in den nächsten Abschnitten behandelt wird.

Algorithmus 10 : Vortermiierung der Aufträge für Bestandsaufbau

foreach Bestellung eines Artikels (*mat*) b_{mat} , bei der die Produktion der Bestellmenge $menge_{b_{mat}}$ aufgrund der Größe der Menge auf mehrere Produktionszyklen aufgeteilt wird, **do**

Schritt 1:

Erstelle eine leere Liste U_{mat} . Diese Liste wird in weiterer Folge mit den Fertigungsaufträgen befüllt, die für den Bestandsaufbau umterminiert werden. Es werden so viele Aufträge zur Umterminierung ausgewählt, dass Summe der Auftragsmengen $\geq menge_{b_{mat}}$. Für die Auswahl in Frage kommen zuerst alle Fertigungsaufträge, deren SB-Datum dem Bedarfsdatum von b_{mat} ($Bedarfsdatum_{b_{mat}}$) entspricht. Wenn die Auftragsmenge dieser Fertigungsaufträge nicht ausreicht, können auch Aufträge gewählt werden, deren SB-Datum bis zu $maxTageFrüher$ Tage vor oder bis zu $maxTageSpäter$ Tage nach $Bedarfsdatum_{b_{mat}}$ liegt.

Schritt 2:

$tBisBedarf$ ist die Zeitspanne bis $Bedarfsdatum_{b_{mat}}$. Berechne die Anzahl der Produktionszyklen $anzProdZyklen$, auf die die Fertigungsaufträge aus U_{mat} aufgeteilt werden. $anzProdZyklen = \lfloor \frac{zeitBisBedarfsDatum - tPuffer}{prodZyklusLaenge} \rfloor$

Schritt 3:

Teile die Fertigungsaufträge aus U_{mat} in $anzProdZyklen$ Gruppen von Aufträgen, wobei die Gruppen möglichst gleich groß sein sollen. Wähle die Gruppe, die als letzte produziert werden soll. Sind die Gruppen nicht gleich groß, so wähle eine der kleineren. Setze für alle Fertigungsaufträge aus dieser Gruppe NB-Datum und SB-Datum auf $Bedarfsdatum_{b_{mat}} - tPuffer$. Berechne die Gesamtproduktionszeit T_{prod} aller Aufträge in der gewählten Gruppe. Setze das Plan-Freigabedatum für jeden Auftrag der Gruppe auf SB-Datum - T_{prod} - Uebergangszeit. $anzUmterminiert$ sei die Anzahl der bereits umterminierten Gruppen von Aufträgen. Setze $anzUmterminiert$ auf 1.

Schritt 4:

Wähle eine der gebildeten Gruppen von Fertigungsaufträgen, die noch nicht umterminiert wurde, und setze NB-Datum und SB-Datum aller Aufträge dieser Gruppe auf $Bedarfsdatum_{b_{mat}} - (prodZyklusLaenge \times anzUmterminiert)$. Berechne die Gesamtproduktionszeit T_{prod} aller Aufträge in der gewählten Gruppe. Setze das Plan-Freigabedatum für jeden Auftrag der Gruppe auf SB-Datum - T_{prod} - Uebergangszeit - $tPuffer$. Erhöhe $anzUmterminiert$ um 1. Wiederhole Schritt 4 bis $anzUmterminiert \geq anzProdZyklen$.

3.4.2 Freigabe fälliger Aufträge und Verbesserung der Arbeitsverteilung

Es wird eine Auswahl der erzeugten und terminierten Fertigungsaufträge gesucht, durch deren Freigabe jede Maschine für eine, vom Benutzer festgelegte Zeitspanne (*tAuslast*, Implementierung: 3 - 10 Schichten) ausgelastet wird. Die Kriterien für diese Auswahl werden in diesem und dem folgenden Abschnitt behandelt. Das Ziel ist für jede Maschine eine Liste von Aufträgen, die in der durch die Liste vorgegebenen Reihenfolge bearbeitet werden. Die nach Algorithmus 9 erzeugte Liste von Listen aller möglichen Fertigungsaufträge je Artikel wird im weiteren Verlauf als *Liste möglicher Fertigungsaufträge* bezeichnet, die Listen von tatsächlich freizugebenden Fertigungsaufträgen je Maschine als *Freigabelisten je Maschine*. Um die Datenkonsistenz zwischen ERP-System und dem Ergebnis des Verfahrens zu gewährleisten, müssen alle Aufträge, die laut ERP-System bereits freigegeben, aber noch nicht rückgemeldet sind, durch KRAF als auf der gleichen Maschine freizugeben, gekennzeichnet werden. Bereits freigegebene Fertigungsaufträge dürfen nur in Ausnahmefällen wieder gelöscht oder auf eine andere Maschine verschoben werden. Diese Aufgabe bleibt dem Disponenten überlassen. Daher werden zunächst alle Aufträge, die laut ERP-System aktuell freigegeben sind, der Freigabeliste jener Maschine zugewiesen, für die sie bereits freigegeben wurden. Aus der Liste möglicher Fertigungsaufträge werden diese Aufträge entfernt. Die Auftragsreihenfolgen auf den Maschinen werden an das Reihenfolgeoptimierungsmodul übergeben und verbessert (siehe Abschnitt 3.5). Sofern nicht gesondert erwähnt, sind mit *freigegebenen Fertigungsaufträgen* ab hier immer die Fertigungsaufträge in den Freigabelisten je Maschine und nicht die laut ERP-System bereits freigegebenen Aufträge gemeint.

Freigabe fälliger Aufträge

Im nächsten Schritt werden all jene Aufträge den Freigabelisten hinzugefügt, deren Plan-Freigabedatum zum Zeitpunkt des Planungslaufs in der Vergangenheit liegt. Diese Aufträge werden als *fällig* bezeichnet, in einer eigenen Liste gespeichert und aus der Liste möglicher Fertigungsaufträge entfernt. Für die Auftragsfreigabe muss bestimmt werden, auf welcher Maschine ein Auftrag produziert wird. In diesem Prozessschritt werden alle fälligen Aufträge für einen Artikel der gleichen Maschine zugeordnet. Eine mögliche Aufteilung erfolgt erst im nächsten Schritt. Vor der Beschreibung des Prozesses zur Maschinenauswahl (siehe Algorithmus 11) sind einige Vorbemerkungen notwendig.

Maximale Maschinenzahl je Artikel

Ein Artikel A darf in $t_{Auslast}$ nicht auf beliebig vielen Maschinen freigegeben werden. Wenn Fertigungsaufträge für eine Maschine freigegeben werden sollen, auf welcher A noch nicht produziert wird, so muss die Belastung der Fertigung durch A ermittelt werden. Die freizugebende Produktionsmenge wird aus den freigegebenen Fertigungsaufträgen für A zuzüglich der neu hinzukommenden Aufträge ermittelt. Als Belastung der Fertigung durch A ist die Zeit definiert, die benötigt wird, um die freizugebende Produktionsmenge auf der Standardmaschine von A zu produzieren. Die durchschnittliche Maschinenbelastung durch A ist die Belastung der Fertigung durch die Anzahl Maschinen, auf denen A bereits freigegeben ist. Wenn diese durchschnittliche Belastung einen definierten Grenzwert (*BelastGrenz_ZusatzMach*, Implementierung: 2 Tage) überschreitet, dürfen die Aufträge auf der zusätzlichen Maschine freigegeben werden. Sonst müssen sie auf einer Maschine produziert werden, auf der bereits Aufträge für A freigegeben sind. Da selbst bei extrem hohem Bedarf ein Artikel nicht die gesamte Fertigung blockieren soll, wird eine höchste erlaubte Anzahl Maschinen je Artikel definiert (Implementierung: 4). Eine Ausnahme, die die Regeln bezüglich der maximalen Maschinenzahl je Artikel außer Kraft setzt, liegt vor, wenn die laut ERP-System freigegebenen Aufträge auf mehr Maschinen verteilt sind, als nach den Regeln gestattet. Solche Aufträge dürfen nicht auf eine andere Maschine verschoben werden. Daher ist in diesem Fall die Freigabe auf allen Maschinen erlaubt, auf denen laut ERP-System bereits Aufträge freigegeben sind.

Bewertung der Rüstzeit

Für die Beurteilung von Rüstzeiten, werden zunächst die Rüstzeiten sämtlicher auf einer Maschine möglichen Rüstfälle berechnet. Die verschiedenen Grenzwerte entsprechen der Rüstzeit jenes Rüstfalls mit der kleinsten Rüstzeit, für den gilt, dass die Rüstzeit länger ist, als die eines definierten Anteils der möglichen Rüstfälle, wobei im Einzelnen folgende Grenzwerte für jede Maschine berechnet werden:

- *Rüstgrenzwert_kurz_fällige* (Implementierung: höchstens 20% der Rüstfälle dürfen kürzere Rüstzeiten aufweisen)
- *Rüstgrenzwert_kurz_vorzuziehende* (Implementierung: höchstens 20% der Rüstfälle dürfen kürzere Rüstzeiten aufweisen)
- *Rüstgrenzwert_mittel* (Implementierung: höchstens 60% der Rüstfälle dürfen kürzere Rüstzeiten aufweisen)

Um zu bewerten, ob ein Fertigungsauftrag gut in die aktuelle Freigabeliste einer Maschine passt, wird der Auftrag an jede erlaubte Position der Reihenfolge gesetzt und die Rüstzeit für das Umrüsten auf diesen Artikel berechnet. Liegt die kürzeste der so ermittelten Rüstzeiten unter dem entsprechenden Grenzwert, wird der Auftrag akzeptiert, sonst abgelehnt.

Bevorzugte Fertigungsversion

Den Fertigungsversionen von Artikeln, die auf verschiedenen Maschinentypen hergestellt werden können, sind Versionsnummern zugeordnet. Näherungsweise kann davon ausgegangen werden, dass bei Artikeln mit mehreren Fertigungsversionen die Fertigungsversion mit der niedrigsten Versionsnummer die bevorzugte und günstigste ist.

Verbesserung der Arbeitsverteilung

Bei der Arbeitsverteilung der fälligen Fertigungsaufträge wird die Belastung der einzelnen Maschinen nur am Rande berücksichtigt und alle Aufträge, die denselben Artikel betreffen, werden derselben Maschine zugeordnet. Dadurch können hohe Belastungen einzelner Maschinen und damit einhergehend, unnötige Verspätungen auftreten. Deshalb wird versucht, die freigegebenen Fertigungsaufträge besser auf die Maschinen zu verteilen. Dabei muss beachtet werden, dass Fertigungsaufträge, die auch im ERP-System schon freigegeben sind, nicht mehr auf eine andere Maschine verschoben werden dürfen. Die Verspätung jedes Auftrags wird im Zuge der Reihenfolgebildung berechnet (siehe Abschnitt 3.5). Algorithmus 12 zeigt das Verfahren, mit dem versucht wird, die Aufträge gleichmäßiger auf die Maschinen zu verteilen, um die Verspätungen zu reduzieren. Da bei jeder Verschiebung eines Fertigungsauftrags auf eine andere Maschine die Auftragsreihenfolgen optimiert werden, sind nach Abschluss des Verfahrens alle Reihenfolgen optimal.

Algorithmus 11 : Maschinenzuordnung fälliger Fertigungsaufträge

Schritt 1:

Weise alle Aufträge für Schnelldreher der Standardmaschine des jeweiligen Artikels zu, außer wenn dadurch das Kriterium der maximalen Maschinenanzahl je Artikel verletzt würde. In diesem Fall finde unter allen Maschinen, auf denen der Artikel bereits produziert wird, diejenige mit der geringsten Auslastung und weise alle Aufträge dieser Maschine zu. Lösche zugeordnete Aufträge aus der Liste der fälligen Aufträge.

Schritt 2:

Versuche für jeden in der Liste fälliger Fertigungsaufträge verbleibenden Auftrag, diesen einer Maschine zuzuordnen, auf der er hinsichtlich der Rüstzeit gut zu den Aufträgen passt, die bereits dieser Maschine zugeordnet sind. Eine Zuordnung ist als passend anzunehmen, wenn für die Umrüstung von mindestens einem Auftrag aus der Freigabeliste der Maschine auf den zuzuordnenden Auftrag gilt: $Rüstzeit < Rüstgrenzwert_{kurz_fällige}$. Das Kriterium der maximalen Maschinenanzahl darf nicht verletzt werden. Wenn mehrere Maschinen zur Auswahl stehen, wird diejenige gewählt, deren Fertigungsversion die niedrigste Versionsnummer hat. Stehen nach diesem Kriterium noch immer mehrere Maschinen zur Auswahl, wird jene mit der geringsten Auslastung gewählt. Lösche zugeordnete Aufträge aus der Liste der fälligen Aufträge und trage sie in die entsprechende Freigabeliste ein.

Schritt 3:

Wie Schritt 2, aber mit $Rüstgrenzwert_{mittel}$ statt $Rüstgrenzwert_{kurz_fällige}$.

Schritt 4:

Wie Schritt 2, aber ohne Berücksichtigung der Rüstzeit

Schritt 5:

Wenn alle fälligen Fertigungsaufträge einer Maschine zugeordnet sind, werden die Auftragsreihenfolgen jeder einzelnen Maschine durch das Reihenfolgeoptimierungsmodul verbessert.

Algorithmus 12 : Verbesserung der Arbeitsverteilung

input : Freigabeliste für jede Maschine

$summeVerspaet \leftarrow highValue$

$neueSummeVerspaet \leftarrow berechneSummeVerspaet(Freigabelisten)$

while $neueSummeAllerVerspaet < summeAllerVerspaet$ **do**

$summeVerspaet \leftarrow neueSummeVerspaet$

Schritt 1:

Finde Maschine M , so dass die Summe der Verspätungen in der Freigabeliste von M größer ist als bei jeder anderen Maschine.

Schritt 2:

Finde den Fertigungsauftrag Fe aus der Freigabeliste von M , der die höchste Verspätung aufweist und in der Freigabeliste nicht an erster Stelle steht.

Schritt 3:

Finde alle Aufträge, durch deren Verschiebung auf eine andere Maschine als M , die Verspätung von Fe reduziert werden kann. Folgende Richtlinien müssen dabei befolgt werden:

- *Nur Aufträge, deren Artikel auf mehr als einer Maschine produziert werden kann.*
- *Nur Aufträge, die laut ERP-System noch nicht freigegeben sind.*
- *Wenn ein Artikel auf mehr als einer Maschine freigegeben werden darf, können einzelne Aufträge verschoben werden. Sonst müssen alle Aufträge zu diesem Artikel gemeinsam verschoben werden. Das ist nur möglich, wenn alle diese Aufträge verschoben werden dürfen. Erstellt wird daher eine Liste L , die weitere Listen enthält. Aufträge, die einzeln verschoben werden können, werden einzeln in Listen der Länge eins gespeichert. Aufträge, die gemeinsam verschoben werden müssen, werden gemeinsam in eine Liste gespeichert.*
- *Grundsätzlich werden nur Aufträge betrachtet, die in der Reihenfolge vor Fe kommen. Wenn alle Aufträge zu einem Artikel gemeinsam verschoben werden müssen, müssen auch die später gereihten Aufträge für den Artikel verschoben werden.*

Schritt 4:

Versuche Verschiebung auf andere Maschine(L, M)

→ siehe Algorithmus 13

$neueSummeVerspaet \leftarrow berechneSummeVerspaet(Freigabelisten)$

Algorithmus 13 : Versuche Verschiebung auf andere Maschine

input : Liste L von Listen von Aufträgen die verschoben werden können,
Maschine M von der die Aufträge aus L wegverschoben werden sollen

foreach *Liste* $l \in L$ **do**

Schritt 1:

Finde alle Maschinen M_{alt} , auf welche die Fertigungsaufträge aus l verschoben werden können, unter Berücksichtigung der maximalen Maschinenanzahl je Artikel.

Schritt 2:

Prüfe für jede Maschine $m_{alt} \in M_{alt}$, ob durch die Verschiebung der Fertigungsaufträge aus l eine Reduktion der Gesamtverspätung der Aufträge aus den Freigabelisten von m_{alt} und M erreicht werden kann.

Schritt 3:

Wenn eine Verbesserung möglich ist, verschiebe die entsprechenden Aufträge und optimiere die Auftragsreihenfolgen.

3.4.3 Kapazitätsauffüllung unter Berücksichtigung reihenfolgeabhängiger Rüstzeiten

Bei Korma werden freie Maschinenkapazitäten aufgefüllt, indem erzeugte Fertigungsaufträge, deren Plan-Freigabedatum in einem gewissen Vorgriffshorizont liegt, vorgezogen werden [25]. Die Freigabe rein nach zeitlicher Nähe des Plan-Freigabedatums berücksichtigt nicht die Arbeitsverteilung und auch nicht die Rüstzeiten. Das Verfahren kann aber relativ einfach in dieser Hinsicht erweitert werden. Die Idee ist, ausgehend von der aktuellen Freigabeliste einer Maschine zur Kapazitätsauffüllung nach Aufträgen im Vorgriffshorizont zu suchen, die in Hinsicht auf die Rüstzeit gut zu den Aufträgen der Freigabeliste passen. Als Nebenbedingung ist dabei die maximale Anzahl von Maschinen je Artikel zu beachten. Weiter sollen Maschinen mit günstiger Fertigungstechnologie, insbesondere in Hinblick auf die Rüstzeitreduktion, bevorzugt behandelt werden und für die Kapazitätsauffüllung eher Schnelldreher eingeplant werden. Die Bevorzugung günstiger Fertigungstechnologien wird durch eine Maschinen-Prioritätsliste (*MaschPrioList*) erreicht, in der Maschinen aufsteigend nach den durchschnittlichen Produktionskosten sortiert sind. Indem zuerst die Kapazität von Maschinen mit günstiger Fertigungstechnologie aufgefüllt wird, steigt die Wahrscheinlichkeit, dass für diese Maschinen im Vorgriffshorizont Fertigungsaufträge vorhanden sind, die bezüglich der Rüstzeit gut in die Freigabeliste passen. Für Schnelldreher wird der Vorgriffshorizont um einen zu definierenden

Faktor (*SchnelldreherKoeffizient*; Implementierung: 1,5) erweitert. Ein Maximalwert für den Vorgriffshorizont wird ebenfalls definiert (*maxVorgriff*; Implementierung: 14 Tage). Dadurch kann die Vorproduktion und damit der Lagerstand eingeschränkt werden. Das gesamte Verfahren ist in Algorithmus 14 festgehalten.

Algorithmus 14 : Kapazitätsauffüllung unter Berücksichtigung reihenfolgeabhängiger Rüstzeiten

input : Freigabeliste für jede Maschine, Liste möglicher Fertigungsaufträge

foreach Maschine $m \in \text{MaschPrioList}$ **do**

$\text{kapa}_m \leftarrow \text{berechneMaschinenkapazitaet}(m)$

$\text{ProdMenge}_m \leftarrow \text{berechneProdMenge}(\text{Freigabeliste}_m)$

$\text{vorgrieffsHorizont} \leftarrow 1$

while $\text{ProdMenge}_m < \text{kapa}_m$ und $\text{vorgrieffsHorizont} \leq \text{maxVorgriff}$ **do**

$\text{planZeit} \leftarrow$ Zeitpunkt zu dem die Planung durchgeführt wird.

$\text{start}_{\text{normal}} \leftarrow \text{planZeit} + \text{vorgrieffsHorizont} - 1$

$\text{ende}_{\text{normal}} \leftarrow \text{planZeit} + \text{vorgrieffsHorizont}$

$\text{start}_{\text{Schnelldreher}} \leftarrow$

$\text{planZeit} + (\text{vorgrieffsHorizont} * \text{Schnelldreherkoeffizient}) - 1$

$\text{ende}_{\text{Schnelldreher}} \leftarrow$

$\text{planZeit} + (\text{vorgrieffsHorizont} * \text{Schnelldreherkoeffizient})$

$\text{suchraum}_{\text{normal}} \leftarrow$ von $\text{start}_{\text{normal}}$ bis $\text{ende}_{\text{normal}}$

$\text{suchraum}_{\text{Schnelldreher}} \leftarrow$ von $\text{start}_{\text{Schnelldreher}}$ bis $\text{ende}_{\text{Schnelldreher}}$

Schritt 1:

Finde freizugebende Aufträge $A_{\text{frei}} \subset$ mögliche Fertigungsaufträge, die folgende Bedingungen erfüllen:

- Artikel kann auf m produziert werden.
- $\min(\text{Rüstzeit von Auftrag aus Freigabeliste}_m) < \text{Rüstgrenzwert}_{\text{kurz_vorzuziehende}}$
- Falls der Artikel ein Schnelldreher ist:
Plan-Freigabedatum in $\text{suchraum}_{\text{Schnelldreher}}$;
Sonst: Plan-Freigabedatum in $\text{suchraum}_{\text{normal}}$
- Können auf m freigegeben werden, ohne die maximale Maschinenanzahl je Artikel zu überschreiten.
- Können auf m freigegeben werden, ohne die Summe der Verspätungen der Aufträge aus Freigabeliste_m zu erhöhen.

Schritt 2:

Füge A_{frei} zu Freigabeliste_m hinzu und entferne A_{frei} aus der Liste möglicher Fertigungsaufträge

Schritt 3:

Optimiere die Auftragsreihenfolge auf m .

Schritt 4:

$\text{ProdMenge}_m \leftarrow \text{ProdMenge}_m + \text{berechneProdMenge}(A_{\text{frei}})$

$\text{vorgrieffsHorizont} \leftarrow \text{vorgrieffsHorizont} + 1$

3.5 Reihenfolgeoptimierung

Sowohl der GA als auch die KRAF-Heuristik benötigen das Modul zur Reihenfolgebildung, um Lösungen für das Problem der kombinierten Auftragserzeugung, Auftragsfreigabe und Arbeitsverteilung unter Berücksichtigung von reihenfolgeabhängigen Rüstzeiten zu erstellen. Damit das Modul die richtigen Lösungen liefert, muss die Zielstellung für die Reihenfolgeoptimierung mit der der übergeordneten Verfahren in Einklang gebracht werden. Von den in Kapitel 2.4 definierten Optimierungszielen können die Produktionskosten und die Kapazitätsauslastung vernachlässigt werden, da die Auftragsreihenfolge auf diese Ziele keinen Einfluss hat, wenn davon abgesehen wird, die Auslastung durch eine ungünstige Rüstreihenfolge zu erhöhen. Die Bestände können unter bestimmten Voraussetzungen von der Auftragsreihenfolge abhängen. Im vorliegenden Fall, in dem die vorhandene Kapazität ausgenutzt werden soll, der Ausstoß der Maschinen vom produzierten Artikel unabhängig ist und die Werte der einzelne Artikel keine großen Unterschiede aufweisen, ist diese Abhängigkeit aber ebenfalls nicht relevant. Im Grunde kann bei einer vorgegebenen Menge von zu reihenden Aufträgen der Bestand ebenfalls nur durch eine ungünstige Rüstreihenfolge gesenkt werden. Aufgrund dieser Überlegungen ergibt sich, dass für die Reihenfolgeoptimierung nur die Rüstzeit und die Fehlmengen beachtet werden müssen. Im Kontext der Reihenfolgeoptimierung werden Fehlmengen üblicherweise durch die Verspätung von Aufträgen abgebildet. Da die Ermittlung der Fehlmengen bzw. Fehlteiltage eine zusätzliche Berechnung erfordert, wird dies auch im vorliegenden Fall so gehandhabt. Um die Verspätung eines Auftrags ermitteln zu können, muss das Bedarfsdatum bekannt sein. In Kapitel 3.4.1 wurde beschrieben, wie das SB-Datum und das NB-Datum auf Basis einer vermuteten Reihenfolge für die Aufträge des gleichen Artikels berechnet werden. Für den GA wird diese Berechnung vom Reihenfolgeoptimierungsmodul durchgeführt, wobei bei den Lösungen des GA die Zuordnung der Aufträge zu Produktionsintervallen beachtet werden muss, wenn die vermutete Reihenfolge der Aufträge eines Artikels erstellt wird. Die Verspätung eines Auftrags ist 0, wenn die Bearbeitung des Auftrags vor seinem NB-Datum abgeschlossen ist, sonst die Zeit zwischen NB-Datum und Fertigstellungszeitpunkt. Für die Berechnung der Rüstzeit einer Reihenfolgelösung müssen die Rüstzeiten für jeden möglichen Rüstfall auf jeder Maschine bekannt sein.

Da mit Rüstzeit und Verspätung zwei Ziele optimiert werden sollen, ist auch die Reihenfolgeoptimierung ein multikriterielles Optimierungsproblem, bei dem die gleichen Schwierigkeiten auftreten wie in Abschnitt 3.3.3 für den multikriteriellen GA beschrieben. Die Methode der gewichteten Summe ist für dieses Problem nicht anwendbar, weil a priori keine sinnvollen Annahmen bezüglich der Wertebereiche für Rüstzeit und Verspätung getroffen werden können. Für das vorliegende Problem wird diese Schwierigkeit umgangen, indem die Ziele priorisiert werden. Die Vermeidung von Lieferengpässen ist in dieser Anwendung deutlich wichtiger, als die Reduktion von Rüstzeiten. Daher wird für die Bestimmung der optimalen Lösung aus einer Gruppe von Reihenfolgelösungen zuerst nur auf die Verspätung geachtet und die Rüstzeit lediglich als Entscheidungskriterium bei gleicher oder ähnlicher Verspätung verwendet. Die Verspätung einer Reihenfolgelösung ist die Summe der Verspätungen aller Aufträge in der Reihenfolge. Die Rüstzeit ist die Summe der Rüstzeiten, die bei der Abarbeitung der Reihenfolge anfallen. Das Verfahren zur Bestimmung der optimalen Lösung durch Priorisierung des Verspätungsziels ist nachfolgend in Algorithmus 15 dargestellt.

Algorithmus 15 : Lösungsauswahl KO-Kriterium Verspätung

Schritt 1:

Bestimme die Lösungen, die aufgrund ihrer Verspätung für die Auswahl in Frage kommen. Wenn es zumindest eine Lösung ohne Verspätung gibt, sind nur Lösungen ohne Verspätung zulässig. Sonst ist die Lösung $L_{gerVersp}$ mit der geringsten Verspätung zulässig sowie alle Lösungen, deren Verspätung nur z.B. 10% höher ist als die Verspätung von $L_{gerVersp}$

Schritt 2:

Wähle aus allen zulässigen Lösungen die mit der geringsten Rüstzeit.

Mit dieser Methode zur Lösungsauswahl können verschiedene Verfahren zur Erzeugung von guten Reihenfolgen eingesetzt werden. Wenn nur wenige Aufträge gereiht werden müssen, z.B. in den kurzen ersten Produktionsintervallen beim GA oder in der Anfangsphase der KRAF-Heuristik kann das Problem durch *Exhaustive Search* gelöst werden. Da beim GA sehr viele Reihenfolgeprobleme gelöst werden, muss dort die maximale Anzahl von Aufträgen, bis zu der diese Methode angewendet wird, deutlich niedriger

angenommen werden als bei der KRAF-Heuristik. In der Implementierung wurde die Exhaustive Search beim GA bis 7 Aufträge angewendet, bei der KRAF-Heuristik bis zu 11. Höhere Werte sind aufgrund des Laufzeitverhaltens nicht geeignet. Durch eine kleine Modifikation kann die Anwendbarkeit für den GA deutlich gesteigert werden. Unter der Annahme, dass auf einer Maschine innerhalb eines Produktionsintervalls alle Aufträge für den gleichen Artikel direkt hintereinander produziert werden, können alle diese Artikel in ein Container-Objekt zusammengefasst werden. Da häufig mehrere Aufträge des gleichen Artikels produziert werden, ist die Anzahl dieser Container-Objekte innerhalb eines Produktionsintervalls oft geringer als die Anzahl der Aufträge und die Container-Objekte können durch Exhaustive-Search gereiht werden. Allerdings geht dadurch die Flexibilität verloren, z.B. zu Beginn und am Ende eines Produktionsintervalls den gleichen Artikel zu produzieren. Wenn die Exhaustive Search nicht angewendet werden kann, wird auf ein zweistufiges heuristisches Verfahren zurückgegriffen. Zuerst wird eine Ausgangslösung erstellt. Da die Vermeidung von Verspätung das priorisierte Ziel ist, wird dafür die Early Due Date (EDD)-Heuristik eingesetzt. Diese liefert das optimale Ergebnis für die Minimierung der maximalen Verspätung, erzeugt aber auch in Hinblick auf die Summe aller Verspätungen brauchbare Lösungen [21]. Zur Verbesserung der Ausgangslösung wird eine Variante der Insertion Search verwendet, bei der jeweils eine Teilsequenz aus der Reihenfolge entfernt wird, um sie an verschiedenen Stellen wieder in die Restreihenfolge einzufügen und die neu entstehende Reihenfolge zu bewerten. Wird eine bessere Lösung gefunden, beginnt das Verfahren von vorne. Wie gründlich der Lösungsraum durchsucht wird, hängt davon ab, wie viele verschiedene Teilsequenzen aus der Ausgangssequenz entnommen werden und an wie vielen jede dieser Teilsequenzen wieder eingefügt wird.

4 Präsentation und Diskussion der Ergebnisse

In dieser Arbeit werden zwei Verfahren gezeigt, die entwickelt wurden, um das Problem der kombinierten Auftragsproduktion, Auftragsfreigabe und Arbeitsverteilung unter Berücksichtigung von reihenfolgeabhängigen Rüstzeiten zu lösen. Das erste ist eine Implementierung des GA, das zweite ein als KRAF-Heuristik bezeichnetes Verfahren, das ausgehend vom Auftragsfreigabeverfahren Korma entwickelt wurde. Für beide Verfahren soll geprüft werden, ob sie geeignet sind, qualitativ hochwertige Lösungen für das Problem zu finden. In Kapitel 2.4 wurden die Ziele definiert, hinsichtlich derer die Lösungen optimiert werden sollten. Außerdem wurden einige zusätzliche Anforderungen definiert, welche die praktische Anwendbarkeit der Verfahren gewährleisten sollten. In diesem Kapitel wird für beide Verfahren gezeigt, wie gut sie die verschiedenen Ziele erreichen.

Bezüglich der Ergebnisqualität ist die Führung eines stichhaltigen Nachweises schwierig. Im Gegensatz zu vielen Standardproblemen des Operations Research, gibt es keine Bibliotheken mit Beispielinstanzen, an denen die Verfahren getestet werden können, um die Ergebnisse mit denen von anderen Verfahren zu vergleichen. Der bestmögliche Nachweis einer hohen Planungsqualität wäre, über einen längeren Zeitraum hinweg, das computergestützte Verfahren gegen den Disponenten antreten zu lassen. Ausgehend vom Zustand der Fertigung zu einem Startzeitpunkt müsste der Disponent die reale und die Software eine simulierte Fertigung steuern. Als Input müssten der Simulation dabei stets aktuelle Informationen über die Bedarfsentwicklung oder das Kapazitätsangebot übergeben werden. Immer wenn der Disponent für die reale Fertigung Aufträge freigibt, müsste auch die Software einen Planungslauf starten und die entsprechenden Aufträge in der simulierten Fertigung freigeben. Auf diese Weise könnte man über einen längeren Zeitraum beobachten, wie sich die Software in Hinblick auf die definierten Ziele gegenüber dem Disponenten schlägt. Allerdings würde dieses Vorgehen bewerten, wie gut das Programm den Produktionsplaner ersetzen kann, während die eigentliche Aufgabenstellung vorsieht, den Disponenten in seiner Arbeit zu unterstützen. Auch ohne diesen Test durchgeführt zu haben, ist es aufgrund der Komplexität der Problemstellung als eher unwahrscheinlich einzuschätzen, dass eines der Verfahren den Disponenten dauerhaft ersetzen könnte. Außerdem wäre dieses Vorgehen extrem aufwendig und würde den Rahmen dieser Arbeit sprengen. Aufgrund dieser Tatsachen wurde für die qualita-

tive Ergebnisbewertung das Prozedere fortgeführt, anhand dessen die Verfahren nach Abschluss der grundlegenden Entwicklungsarbeit verbessert wurden: Die Software liefert eine Lösung für die aktuelle Planungssituation. Der Disponent vergleicht diese mit seinen eigenen Überlegungen und kommentiert die Lösung der Software. Aus den Kommentaren können gewisse Rückschlüsse auf die Lösungsqualität gezogen werden. Neben dem drastisch reduzierten Aufwand bietet dieses Vorgehen den weiteren Vorteil, das Risiko zu reduzieren, dass das Programm als Konkurrent statt als Unterstützung gesehen wird. Die Ergebnisse dieser Bewertungen werden in Abschnitt 4.1 präsentiert. Danach folgt ein Vergleich der beiden Konzepte, wobei die Anwendbarkeit im Fokus steht. Den Abschluss dieses Kapitels bilden die Testergebnisse für einige Operatoren und Parametereinstellungen des GA. Diese wurden eigens für das vorliegende Problem entwickelt. Die Testergebnisse sollen zeigen, ob die Ansätze funktionieren und möglicherweise Ausgangspunkt für weitere Forschungsarbeiten auf diesem Gebiet sein.

4.1 Qualitative Lösungsbewertung durch den Produktionsplaner

Ab dem Zeitpunkt, zu dem der Implementierungsfortschritt der Verfahren die Erzeugung von sinnvollen Ergebnissen zuließ, wurden mehrmals die Vorschläge der Software mit dem Disponenten besprochen. Dafür wurden aktuelle Daten aus dem Betrieb, wie Bestellungen, freigegebene Fertigungsaufträge oder der aktuelle Schichtplan zur Verfügung gestellt und durch die in Kapitel 3 beschriebenen Module verarbeitet. Die erzeugten Vorschläge für die Auftragsfreigabe wurden mit dem Disponenten besprochen und basierend auf seinen Anmerkungen wurden Verbesserungen an den Verfahren vorgenommen. Wie zu Beginn dieses Kapitels beschrieben, ist es schwierig, eine objektive Bewertung der Lösungsqualität vorzunehmen. Der beste gefundene Anhaltspunkt sind die Anmerkungen des Disponenten, der mit seiner Erfahrung und dem Wissen um die Zusammenhänge in der Planung für diese Fertigung, beurteilen kann, ob bestimmte Auftragsvorschläge sinnvoll sind oder nicht. Es muss aber auch auf ein Problem hingewiesen werden, das sich aus dieser Form der Lösungsbewertung ergibt. Dem Disponent wird gleichsam zugestanden, unfehlbar zu sein. Wenn er einen guten Vorschlag nicht als solchen erkennt, besteht nur eine geringe Chance, dass der Irrtum aufgedeckt wird, da für jeden anderen die Bewertung eines Vorschlags schwieriger und mit deutlich mehr Aufwand verbunden ist. Grundsätzlich kann allerdings davon ausgegangen werden, dass die Lösungsbewertung durch den Disponenten korrekt ist. Daher werden nachfolgend einige Ergebnisse präsentiert, die aus der Analyse der Anmerkungen des Disponenten gewonnen wurden.

Die Bemerkungen des Disponenten betreffen drei getrennte Ergebnisse, die verteilt über einen Zeitraum von ca. 2 Wochen entstanden. Es wurden also drei voneinander unabhängige Bedarfssituationen und Zustände der Fertigung betrachtet. Jedes Mal wurde ein Ergebnis des GA und eines der KRAF-Heuristik durch den Disponenten bewertet. Bei der Analyse, der die folgenden Ergebnisse entstammen, wurden jeweils die ersten drei Tage nach dem Planungszeitpunkt berücksichtigt. Das entspricht in etwa dem maximalen Zeitraum, für den die Verfahren in der Praxis Fertigungsaufträge freigeben sollen. Die von der Software zur Freigabe vorgeschlagenen Aufträge werden in drei Gruppen aufgeteilt. Erstens Fertigungsaufträge, die bereits zu einem früheren Zeitpunkt freigegeben wurden und im ERP-System als freigegeben, aber noch nicht rückgemeldet, geführt werden. Zweitens Freigabevorschläge, die im Planungslauf neu erstellt werden und die der Disponent für gut befindet, und schließlich Vorschläge, die aus verschiedenen Gründen vom Disponenten abgelehnt werden. Da Aufträge, die im ERP-System als freigegeben aufscheinen, nicht mehr gelöscht werden dürfen, verliert die Software an Gestaltungsraum, je mehr Aufträge bereits freigegeben sind. Das gilt besonders für die KRAF-Heuristik, da diese für einen kürzeren Zeithorizont plant und deshalb bereits freigegebene Aufträge nicht so weit nach hinten verschieben kann wie der GA. Die Ergebnisse der Analyse werden in Tabelle 5 präsentiert. Die unterschiedliche Gesamtanzahl von Aufträgen ergibt sich aus den unterschiedlichen Produktionsmengen der Aufträge. Die Anzahl der abgelehnten Auftragsvorschläge ist eher hoch, wird jedoch etwas rela-

Tabelle 5: Ergebnisse Planungsbewertung

	GA	KRAF
bereits freigegeben	89	116
angenommene Vorschläge	38	42
abgelehnte Vorschläge	35	18

tiviert, wenn die Gründe für das Entstehen dieser Vorschläge betrachtet werden. Zwei abgelehnte Vorschläge des GA sind auf fehlerhafte Stammdaten zurückzuführen, genauer gesagt auf falsche Angaben darüber, auf welchen Maschinen ein Artikel gefertigt werden kann. Vier Ablehnungen von Vorschlägen der KRAF-Heuristik kommen daher, dass ein Artikel aus technischen Gründen nicht produziert werden kann. Dieses Problem kann durch eine entsprechende Eingabemöglichkeit leicht behoben werden. Der

Großteil der abgelehnten Vorschläge (GA: 11; KRAF: 7) liegt jedoch darin begründet, dass in einer Engpasssituation in der Realität die strategische Entscheidung getroffen wurde, eine große Kundenbestellung nicht vollständig zu produzieren und stattdessen die Bedarfe des Werksverbunds abzudecken. Die Verfahren halten jedoch aufgrund des näherliegenden Bedarfstermins an der Produktion für die Kundenbestellung fest und versuchen, die Bedarfe des Werksverbundes mittels alternativer Fertigungstechnologien abzudecken. Laut Produktionsplaner ist dies aber keine gute Lösung. Dieses Problem kann möglicherweise ebenfalls durch eine Eingriffsmöglichkeit für den Anwender behoben werden. Grundsätzlich ist aber festzuhalten, dass die Verfahren als Unterstützung im operativen Tagesgeschäft konzipiert sind und nicht für strategische Entscheidungen. Die oben genannten Fälle lassen keinen Schluss darauf zu, dass die Lösungsqualität der Verfahren mangelhaft ist, da sie entweder auf mangelhaften Daten oder auf dem Einsatz unter ungeeigneten Umständen beruhen. Daraus ergibt sich das in Tabelle 6 gezeigte, bereinigte Ergebnis. Dieses lässt den Schluss zu, dass sich die KRAF-Heuristik etwas

Tabelle 6: Ergebnisse Planungsbewertung (bereinigt)

	GA	KRAF
bereits freigegeben	89	116
angenommene Vorschläge	38	42
abgelehnte Vorschläge	22	7

mehr an den bereits freigegebenen Fertigungsaufträgen orientiert. Das entspricht auch dem erwarteten Ergebnis, da sie alle bereits freigegebenen Aufträge in einem kürzeren Planungshorizont unterbringen muss. Bei den neu zur Freigabe vorgeschlagenen Aufträgen scheint KRAF etwas bessere Ergebnisse zu liefern. Nachfolgend werden einige mögliche Erklärungen für dieses Verhalten präsentiert. Der erste Erklärungsansatz bezieht sich auf die Art der Lösungserstellung. Ein Vergleich der Arten der Lösungserstellung wird in Abschnitt 4.2 präsentiert. Dort wird gezeigt, dass die KRAF-Heuristik die Optimierungsziele direkter umsetzt. Möglicherweise werden dadurch qualitativ bessere Lösungen erzielt. Andererseits ist die Lösungserstellung der KRAF-Heuristik deutlich stärker an der Vorgehensweise des Disponenten orientiert. Es ist daher nicht auszuschließen, dass die Lösungen besser bewertet werden, weil sie näher am erwarteten Ergebnis liegen. Eine weitere mögliche Erklärung für die höhere Lösungsqualität der KRAF-Heuristik hängt mit der Bedarfsprognose zusammen. Wie in Kapitel 3.1 beschrieben, nimmt die Prognosequalität ab, je weiter in die Zukunft geblickt wird. Die Implementierung des

GA berücksichtigt die Bedarfsprognose für den gesamten Betrachtungszeitraum, während die KRAF-Heuristik besonders die Bedarfe in naher Zukunft beachtet und Bedarfe, die weiter als zwei Wochen (Schnelldreher: 3 Wochen) in der Zukunft liegen vollständig ignoriert. Der insgesamt größere Prognosefehler in der Lösung des GA könnte sich auch auf die zeitnahen Freigabevorschläge auswirken. Schlussendlich kann die niedrigere Lösungsqualität des GA auch darin begründet liegen, dass das Verfahren zu früh abgebrochen wird. Wie in Abschnitt 4.2 beschrieben wird, ist der GA zu langsam. Um den GA nicht auf Basis von Ergebnissen zu bewerten, die wegen zu langer Laufzeit in der Praxis nicht erzielt werden können, wird die Anzahl der Generationen auf 50 beschränkt. Tests zeigen, dass bessere Ergebnisse erzielt werden können, wenn mehr Generationen berechnet werden.

4.2 Konzeptueller Vergleich zwischen GA und KRAF hinsichtlich der Anwendbarkeit

Neben den Anforderungen an die Lösungsqualität muss ein Verfahren auch gewisse Nebenbedingungen erfüllen, um die Anwendbarkeit zu gewährleisten. Beispielsweise kann die Lösung für ein Optimierungsproblem noch so gut sein, wenn sie nicht zu dem Zeitpunkt verfügbar ist, zu dem sie benötigt wird, ist jeglicher Aufwand für ihre Erstellung verloren. Einige dieser Faktoren wurden zu Beginn des Projekts definiert und in Kapitel 2.4 festgehalten. Andere Unterschiede zwischen den beiden entwickelten Verfahren zeigten sich erst im Zuge der Implementierung, bedingen aber ebenfalls Unterschiede in der praktischen Anwendbarkeit. Die verschiedenen Aspekte werden nachfolgend angeführt. Am Ende dieses Abschnitts erfolgt eine Diskussion, was diese Faktoren für die Anwendbarkeit der Verfahren in verschiedenen Bereichen bedeuten.

Laufzeit

In der Projektdefinition wurde festgehalten, dass die Laufzeit des Verfahrens ohne die Datengewinnung aus dem ERP-System nicht mehr als 10 Minuten betragen soll. Der Grund dafür ist, dass die Produktionsplanung 2 bis 3 Mal pro Tag aktualisiert wird, und Software, die so häufig gebraucht wird, nicht gut angenommen wird, wenn lange auf das Ergebnis gewartet werden muss. Der GA kann diese Bedingung bei weitem nicht einhalten. Die aufwendige Lösungsbewertung, insbesondere die für jede Lösung, für jedes Produktionsintervall auf jeder Maschine durchgeführte Reihenfolgebildung, führt zu deutlich längeren Laufzeiten. Die in Abschnitt 4.1 bewerteten Ergebnisse des GA wurden in 50 Generation mit je 20 Individuen er-

reicht, aber schon die Laufzeit für diese Einstellungen liegt über einer Stunde. Die KRAF-Heuristik hat dieses Problem nicht. Bei sinnvollem Einsatz der Exhaustive Search in der Reihenfolgeoptimierung wird normalerweise eine Laufzeit von unter 3 Minuten erzielt. Diese Laufzeitangaben enthalten für beide Verfahren nicht die benötigte Zeit für das Einlesen der Daten und die Bedarfsprognose. Wenn die Stammdaten schon eingelesen sind, benötigen diese beiden Schritte in etwa eine Minute.

Reproduzierbarkeit des Ergebnisses

Der Zufallsfaktor ist ein integraler Bestandteil von GA. Wenn die Parameter richtig gewählt sind, liefern diese Verfahren dennoch zuverlässig konstant gute Lösungen. Abgesehen davon, dass die richtige Wahl der Parameter alles andere als ein triviales Problem ist, bedeutet das, dass konstant gute Werte der Zielfunktion erreicht werden. Bei multikriteriellen Problemen können jedoch Lösungen, die große Unterschiede aufweisen ähnliche Zielfunktionswerte erreichen. Dies erklärt, warum der GA für dieselben Eingangsdaten, bei mehreren Durchläufen zu deutlich unterschiedlichen Lösungen kommt. Unter der Annahme, dass die Zielfunktion richtig gestaltet ist, stellt das zwar objektiv kein großes Problem dar, weil die Lösungen gleich gut sind. In der praktischen Anwendung ist ein solches Verhalten aber eher nicht erwünscht. Besonders wenn die Software auch von Personen eingesetzt werden soll, die im Umgang damit nicht viel Erfahrung haben, können unterschiedliche Ergebnisse zu Verwirrung und Verunsicherung führen. Die KRAF-Heuristik ist nicht von Zufallseinflüssen abhängig, sondern trifft nach fixen Regeln für dasselbe Problem immer dieselbe Entscheidung. Daher führt dieses Verfahren für dieselben Eingangsdaten immer zum selben Ergebnis.

Komplexität des Verfahrens

Die KRAF-Heuristik ist deutlich einfacher. Teile des Verfahrens orientieren sich am Vorgehen des Disponenten. Dadurch ist, im Vergleich zum GA, leichter nachzuvollziehen, wie und warum das Verfahren funktioniert. Für das vollständige Verständnis des GA müsste beispielsweise die Schematheorie erklärt werden [28]. Das Verständnis des Verfahrens ist jedoch wichtig, da der Disponent manche Verfahrensparameter anpassen muss, wenn Änderungen bei den Zielen oder den Gegebenheiten in der Fertigung auftreten. Die Anzahl der benötigten Verfahrensparameter ist ein weiteres Unterscheidungsmerkmal der beiden Verfahren. Während die KRAF-

Heuristik 15 Parameter benötigt, sind es beim GA mehr als 50. Den Überblick über diese vielfältigen Möglichkeiten zu behalten, ist eine große Herausforderung und es ist zu vermuten, dass diese Aufgabe wegen des großen damit verbundenen Aufwands eher vernachlässigt wird. Neben den Schwierigkeiten für den Disponenten bringt die hohe Komplexität des GA auch Nachteile bei der Wartbarkeit der Software mit sich. Während die KRAF-Heuristik mit etwas 3000 Zeilen Code implementiert wird, besteht die Implementierung des GA aus mehr als 10000 Zeilen. Allerdings wird dieser Unterschied dadurch relativiert, dass die Implementierung der anderen Module (vgl. Kapitel 3.1) und aller benötigten Geschäftsobjekte wie Maschinen, Artikel oder Aufträge mehr als 40000 Zeilen Code erfordert.

Art der Lösungserstellung

Der GA erstellt unter Einfluss von Zufallsfaktoren verschiedene Lösungen und bewertet diese. Gute Lösungen werden eher akzeptiert und die guten Lösungseigenschaften bleiben der Population erhalten. Abgesehen von der in Abschnitt 3.3.4 beschriebenen gezielten Mutation werden die Optimierungsziele nur indirekt, durch die Ansammlung von Eigenschaften verfolgt, die eine gute Erreichung der Zielfunktion bedeuten. Der Grund für dieses Ansammeln von guten Eigenschaften wird durch die Schematheorie behandelt. Die Regeln der KRAF-Heuristik andererseits zielen direkt auf verschiedene Optimierungsziele ab, ohne den Umweg über eine einzige Zielfunktion nehmen zu müssen. Der Grad der Rüstoptimierung kann beispielsweise durch den Schwellenwert für die Akzeptanz eines Rüstfalls und die Größe des erlaubten Vorgriffshorizontes beeinflusst werden. Ein sehr niedriger Grenzwert in Verbindung mit großem Vorgriffshorizont führt dazu, dass Aufträge die eine sehr niedrige Rüstzeit verursachen, weit vorgezogen werden können. Wird der Grenzwert jedoch erhöht, so wird mehr gerüstet und die Fertigung wird stärker am Bedarfsverlauf orientiert. Daraus ergibt sich, dass der Fokus der Optimierung beim GA über die Gewichtung der Zielfunktion sehr leicht eingestellt werden kann, aber die KRAF-Heuristik den eingestellten Fokus direkter verfolgt.

Anpassbarkeit für andere Fertigungsumgebung

Beide Verfahren sind in ihrer derzeitigen Auslegung für eine einstufige Lagerfertigung vorgesehen. Die Adaptierung für andere Fertigungen der gleichen Art scheint möglich, muss jedoch im Einzelfall geprüft werden. Wenn beispielsweise vor der Auftragsfreigabe eine Materialverfügbarkeitsprüfung durchgeführt werden muss, sind größere Anpassungen nötig. Gleiches gilt noch verstärkt für mehrstufige Fertigungen mit komplexem Materialfluss. Es ist anzunehmen, dass der GA leichter angepasst werden kann, da das Verfahren grundsätzlich flexibler ist und auch in der vorliegenden Implementierung die Eigenschaften der konkreten Fertigung weniger stark miteinbezieht.

Zusammenfassend kann festgehalten werden, dass die KRAF-Heuristik für den in Kapitel 2 beschriebenen Einsatzzweck besser geeignet ist. Die kürzere Laufzeit, höhere Lösungsqualität und bessere Verständlichkeit ergeben für den täglichen Einsatz ein eindeutiges Urteil. Es kann jedoch durchaus auch für den GA sinnvolle Einsatzbereiche geben. Insbesondere die Planung für die Produktion am Wochenende ist hier zu nennen. Da Aufträge für einen größeren Zeitraum freigegeben werden, ist möglicherweise die Bewertung der Lösung für den gesamten Betrachtungszeitraum besser geeignet als die Erstellung nach fixen Regeln. Da am Wochenende keine Änderung der Bedarfssituation eintritt, spielt auch die Variabilität des Ergebnisses eine geringere Rolle. Der größte Faktor ist allerdings, dass die Laufzeit für den Einsatz einmal pro Woche eine geringere Rolle spielt.

4.3 Veränderungen im Laufzeitverhalten des GA durch gezielte Mutation und lokale Suche

Das Konvergenzverhalten von GA wird durch die Auswahl der genetischen Operatoren sowie die Festlegung der Parameterwerte bestimmt. In diesem Abschnitt soll gezeigt werden, wie es sich durch den Einsatz der in Kapitel 3.3.4 beschriebenen Operatoren für gezielte Mutation verändert. Außerdem wird der Einfluss der lokalen Suche durch Optimierung von einzelnen Maschinen präsentiert. Es wird dazu der erreichte Wert der summierten gewichteten Zielfunktion der Laufzeit gegenübergestellt. Häufig wird in der Präsentation der Ergebnisse von GA statt der verstrichenen Laufzeit die Anzahl berechneter Generationen als Bezugsmaß verwendet. Im vorliegenden Fall ist jedoch die Laufzeit ein kritischer Faktor. Da das Ziel der Tests die Verbesserung des Ergebnisses in kurzer Zeit war, wird diese Darstellung gewählt. Die Ergebnisse basieren auf einer

Testumgebung mit 20 Artikeln und 4 Maschinen. Diese Reduktion ist notwendig, um mehr Generationen berechnen zu können. Jeder Test wurde 10 mal durchgeführt und die Ergebnisse gemittelt, um den Einfluss des Zufallsfaktors zu reduzieren. Die Abbildungen 4, 5 und 6 zeigen Diagramme mit der Entwicklung der Zielfunktion bei verschiedenen Verfahren der gezielten Mutation. Als Referenz ist jeweils ein Verlauf ohne gezielte Mutation enthalten. Dieser Referenzverlauf ist in jedem Diagramm gleich. Die beiden weiteren Verläufe entstehen durch den Einsatz gezielter Mutation. Sie unterscheiden sich in der Frequenz des Einsatzes des jeweiligen Mutations-Operators. Bei jedem Einsatz eines Mutations-Operators kommen die 20 besten bisher gefundenen Lösungen für die Mutation infrage. Die neu entstehenden Lösungen werden der aktuellen Population hinzugefügt. Die unterschiedlichen Endpunkte der Verläufe ergeben sich daraus, dass der GA jeweils nach 520 Generationen abgebrochen wird, auf der Abszisse aber die Laufzeit aufgetragen ist. Es ist eindeutig erkennbar, dass mit gezielter Mutation bessere Lösungen gefunden werden als ohne. Eine mögliche Erklärung ist, dass schon kleine Änderungen, wie das Vorziehen von einigen wenigen Aufträgen eine starke Verbesserung der Lösungsqualität bewirken. Da der Lösungsraum aber unendlich groß ist, kann es sehr leicht sein, dass diese guten Eigenschaften in der Ausgangspopulation nicht vorkommen. In diesem Fall ist Mutation die einzige Möglichkeit diese Eigenschaften in die Population einzubringen. Tatsächlich ist auch ungezielte Mutation ein sehr wichtiger Faktor und es werden gute Ergebnisse erzielt, wenn nach Abflachen der Zielfunktion die Mutationswahrscheinlichkeit für ungezielte Mutation auf sehr hohe Werte, beispielsweise 70%, gesetzt wird. Vor allem in Hinblick auf die schnelle Konvergenz zu Beginn ist die gezielte Mutation der ungezielten aber klar überlegen. Abbildung 7 zeigt die Entwicklung der Zielfunktion beim Einsatz des GA als lokales Suchverfahren für die Optimierung einzelner Maschinen. Wiederum ist der gleiche Datensatz wie zuvor als Referenzverlauf im Diagramm enthalten. Bei diesem wird keine lokale Suche durchgeführt. Die beiden anderen Verläufe unterscheiden sich in der Häufigkeit der lokalen Suche und der Anzahl Generationen des GA, die für jede Maschine berechnet werden. Durch den Einsatz der lokalen Suche werden deutlich schneller gute Lösungen und auch absolut bessere Lösungen gefunden. Bei genauer Betrachtung kann man erkennen, wie die Zielfunktion in regelmäßigen Abständen sprunghaft bessere Werte annimmt. Der Grund dafür ist die lokale Suche.

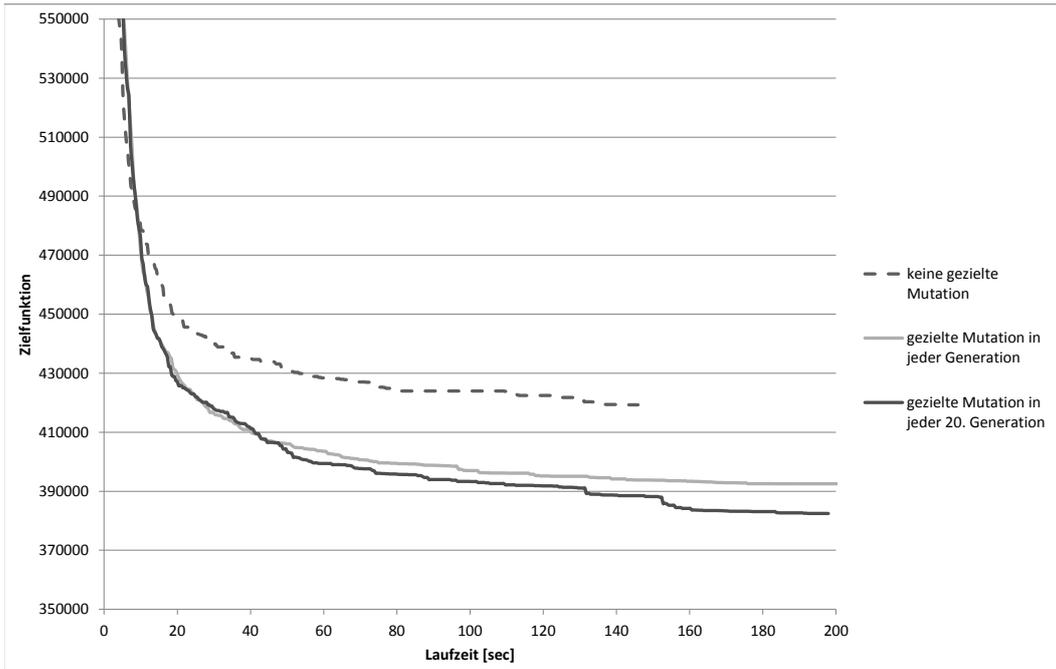


Abbildung 4: Gezielte Mutation: Hinzufügen von Aufträgen für Schnelldreher

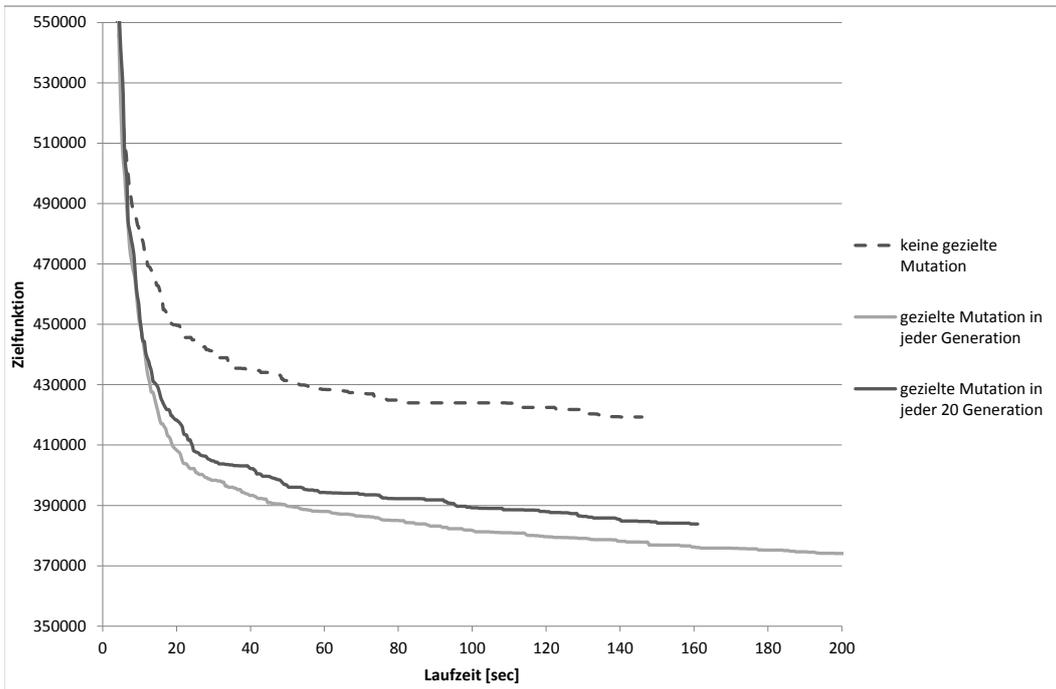


Abbildung 5: Gezielte Mutation: Reduktion der Rüstkosten

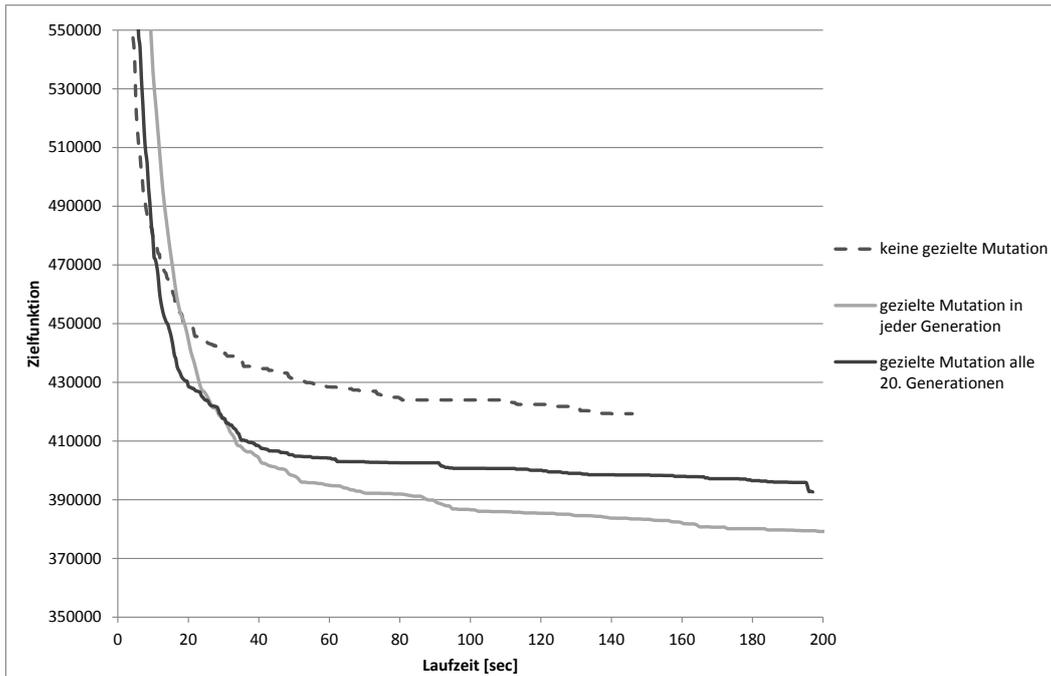


Abbildung 6: Gezielte Mutation: Fehlmengenreduktion durch Vorziehen von Aufträgen

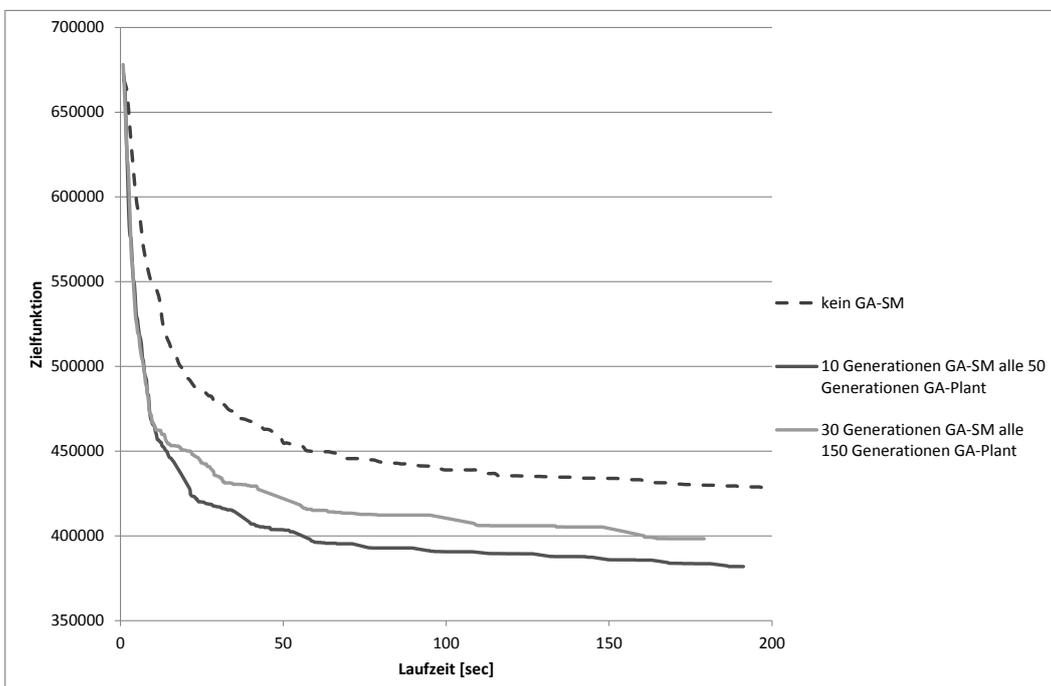


Abbildung 7: Lokale Suche durch Optimierung von Einzelmaschinen

5 Fazit und Ausblick

In der vorliegenden Arbeit wird die Lösung des Problems der kombinierten Auftrags-erzeugung, Auftragsfreigabe und Arbeitsverteilung bei reihenfolgeabhängiger Rüstzeit behandelt. Aufgrund kurzer Lieferzeiten und damit einhergehender, sehr kurzer Pla-nungszyklen fallen diese Funktionen in dem realen Fall, welcher der Arbeit zugrunde liegt, zusammen. Es wird gezeigt, welche Anforderungen ein Verfahren erfüllen muss, um das Problem zu lösen und dass die aus der Literatur bekannten Verfahren diesen Anforderungen nicht genügen. Es werden daher zwei neue Verfahren zur Lösung des Problems entwickelt und in ein Softwarepaket integriert. Dieses Paket umfasst neben den beiden neu entwickelten Verfahren Module für das Einlesen von Daten, die Bedarfs-prognose und die Reihenfolgeplanung, welche kurz beschrieben werden.

Bei den beiden neuen Verfahren handelt es sich um die Implementierung eines Ge-netischen Algorithmus GA und die Kapazitäts- und Rüstorientierte Auftragsfreigabe (KRAF)-Heuristik. Für den GA wird eine Lösungscodierung gezeigt, die eine große Klas-se von ungültigen Lösungen ausschließt, und wie Lösungen nach dem Zielerreichungsgrad von mehreren, einander zum Teil widersprechenden Zielen, bewertet werden können. Au-ßerdem wird die gezielte Mutation vorgestellt, bei der versucht wird, Schwachstellen in Lösungen zu finden und diese auszubessern. Die KRAF-Heuristik nimmt Anleihen beim Korma-Verfahren und dem Vorgehen des Disponenten bei der manuellen Planung. Das Kernelement ist, dass nach der Einplanung der dringenden Aufträge, freie Maschinen-kapazitäten durch das Vorziehen von Aufträgen aufgefüllt werden, wobei nur Aufträge vorgezogen werden, die hinsichtlich der Rüstzeit gut in die aktuelle Auftragsreihenfolge passen.

Im Vergleich der beiden Verfahren zeigt sich, dass die KRAF-Heuristik dem GA in den meisten Fällen überlegen ist. Sämtliche Mängel des GA hängen, zumindest zu ei-nem gewissen Grad, mit der zu geringen Geschwindigkeit dieses Verfahrens zusammen. Hier könnten weitere Verbesserungen vorgenommen werden, beispielsweise durch Par-allelisierung von Berechnungsprozessen. Etwa 95 % der Laufzeit des GA entfallen auf die Bewertung von Reihenfolgelösungen. Häufig liegt eine Menge von alternativen Rei-

henfolgelösungen vor, deren Bewertung parallel statt sequenziell durchgeführt werden könnte. Da die Reihenfolgebildung auf jeder Maschine je Produktionsintervall durchgeführt wird, könnten auch die Reihenfolgeoptimierungen parallelisiert werden. Hardware mit ausreichender Anzahl von Prozessorkernen vorausgesetzt, könnte, grob und ohne Berücksichtigung des Overheads berechnet, alleine diese Maßnahme eine Laufzeitreduktion von 90% bewirken.

Bei der präsentierten Implementierung zeigt sich allerdings, dass die Einfachheit der KRAF-Heuristik ihr großer Vorzug ist. Neben Wartbarkeit und Verständnis sind auch die Fehlerbehebung und das Feintuning in der Entwicklungsphase deutlich einfacher. Werden in der Lösung unerwünschte Eigenschaften festgestellt, kann in die Regeln zur Lösungserstellung eingegriffen werden, um direkt ein besseres Verhalten zu erzielen. Beim GA sind die kausalen Zusammenhänge viel weniger eindeutig und es ist in der Regel mit deutlich mehr Aufwand verbunden, ein gewünschtes Verhalten zu erzielen. Feintuning dieser Art sollte auch gegenüber dem präsentierten Stand der Implementierung durchgeführt werden, um die Fehler weiter zu reduzieren. Dieser Prozess muss in enger Abstimmung mit dem Disponenten stattfinden. Gemeinsam sind auch genaue Regeln zu erarbeiten, nach denen die Verfahrensparameter geändert werden müssen, um ein bestimmtes Verhalten des Verfahrens zu erreichen. Durch diese Arbeit soll das Vertrauen in die Software gestärkt werden. Nur wenn Vertrauen in die Ergebnisse besteht, kann das Ziel erreicht werden, dem Disponenten Arbeit abzunehmen. Weiter ist seine Meinung vermutlich ausschlaggebend für die Verwendung der Software durch seine Kollegen im Vertretungsfall.

Beim Feintuning muss aber immer berücksichtigt werden, dass die Lösungsqualität nicht höher sein kann, als die Qualität der Eingangsdaten. Es gilt hier die bekannte Regel: "*Eine Kette ist so stark wie ihr schwächstes Glied*". Daher sind einerseits die Stammdaten gründlich zu überprüfen. Andererseits scheint aus diesem Grund eine intensivere Beschäftigung mit dem Modul zur Bedarfsprognose großes Verbesserungspotenzial zu bergen. Neben den bereits beschriebenen Potentialen für die Verbesserung der Lösungsqualität des konkreten Problems, erscheinen besonders die Ergebnisse zum Einsatz der gezielten Mutation als vielversprechende Anknüpfungspunkte für zukünftige Forschungstätigkeit.

Es wurde gezeigt, dass durch die Integration von Informationen darüber, wie gute Lösungen in der manuellen Planung zustande kommen, der GA schneller gute Lösungen findet. Es wäre interessant zu überprüfen, ob ähnliche Informationen auch in anderen Anwendungsfällen zur Verfügung stehen und ob deren Einbeziehung einen Vorteil mit sich bringt.

Grundsätzlich kann die Umsetzung der Verfahren für die Lösung des Problems der kombinierten Auftragszeugung, Auftragsfreigabe und Arbeitsverteilung bei reihenfolgeabhängiger Rüstzeit als erfolgreich angesehen werden. Besonders die KRAF-Heuristik erzeugt brauchbare Lösungen und ist auch von den Rahmenbedingungen her in der Praxis einsetzbar. Wie beschrieben gibt es noch verschiedene Verbesserungspotenziale. Die Umsetzung dieser Potenziale ist sicher zielführend. Besonders wenn es gelingt das Verfahren so zu verallgemeinern, dass auch Fertigungen mit komplexerem Materialfluss gesteuert werden können, eröffnen sich zahlreiche Möglichkeiten für die praktische Anwendung.

Literatur

- [1] Dieter Arnold, Kai Furmans, Heinz Isermann, Axel Kuhn, and Horst Tempelmeier. *Handbuch Logistik (VDI-Buch) (German Edition)*. VDI-Buch. Springer, Dordrecht, 3rd ed. edition, 2008.
- [2] Jochen Benz and Markus Höflinger. *Logistikprozesse mit SAP: Eine anwendungsbezogene Einführung ; mit durchgehendem Fallbeispiel ; geeignet für SAP Version 4.6A bis ERP 2005 ; [mit Online-Service]*. Studium. Vieweg + Teubner, Wiesbaden, 2., aktualisierte aufl. edition, 2008.
- [3] Herbert Bernstein. *Elektrotechnik/Elektronik für Maschinenbauer: Grundlagen und Anwendungen*. Vieweg+Teubner Verlag, Wiesbaden, 2., überarb. aufl. 2012 edition, 2012.
- [4] A. M. Bonvik, C. E. Couch, and S. B. Gershwin. A comparison of production-line control mechanisms. *International Journal of Production Research*, 35(3):789–804, 1997.
- [5] J.A. Buzacott and J.G. Shantikumar. A general approach for coordinating production in multiple-cell manufacturing systems. 1(1):34–52, 1992.
- [6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [7] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, Ltd, Chichester, 1st edition, 2004.
- [8] Jürgen Finger. *Erfolgreiche ERP-Projekte*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [9] Verena Gondek. *Hybrid Flow-Shop Scheduling mit verschiedenen Hybrid Flow-Shop Scheduling mit verschiedenen Restriktionen: Heuristische Lösung und LP-basierte untere Schranken*. Dissertation, Universität Duisburg-Essen, 01.01.2011.
- [10] ROBERT J. GRAVES, JOHN M. KONOPKA, and R. JOHN MILNE. Literature review of material flow control mechanisms. *Production Planning & Control*, 6(5):395–403, 1995.

- [11] Willibald A. Günthner and Julia Boppert. *Lean Logistics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [12] Robert W. Hall. *Driving the productivity machine: Production planning and control in Japan : a research report*. American Production and Inventory Control Society, Falls Church, Va., 1981.
- [13] Simon Haykin. *Neural networks: A comprehensive foundation*. Pearson Education, Delhi, 2nd ed. edition, 1999.
- [14] Mohamad Bin Ismail and Usman Dauda. Standardization and its effects on k-means clustering algorithm. *Research Journal of Applied Sciences, Engineering & Technology*, 6(17):3299–3303, 2013.
- [15] Ismail Kasikci. *Elektrotechnik für Architekten, Bauingenieure und Gebäudetechniker: Grundlagen und Anwendung in der Gebäudeplanung*. SpringerLink : Bücher. Springer Vieweg, Wiesbaden, 2013.
- [16] Manuel Laguna and Rafael Martí. *Scatter Search: Methodology and Implementations in C*. Springer US, Boston, MA, 2003.
- [17] Rainer Leisten. *Iterative Aggregation und mehrstufige Entscheidungsmodelle: Einordnung in den planerischen Kontext, Analyse anhand der Modelle der Linearen Programmierung und Darstellung am Anwendungsbeispiel der Hierarchischen Produktionsplanung*. Produktion und Logistik. Physica-Verlag HD, Heidelberg, 1996.
- [18] Hermann Lödding. *Verfahren der Fertigungssteuerung*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
- [19] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures: Evolution Programs*. Springer Berlin Heidelberg, third, revised and extended edition edition, 1996.
- [20] Peter Nyhuis and Hans-Peter Wiendahl. *Logistische Kennlinien*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [21] Michael Pinedo. *Scheduling: Theory, algorithms, and systems*. Springer, 4th ed. edition, 2012.
- [22] Russell D. Reed and Robert J. Marks. *Neural smithing: Supervised learning in feedforward artificial neural networks*. MIT Press, Cambridge, Mass., 1999.

- [23] Derek John Scott Robinson. *An introduction to abstract algebra*. de Gruyter, Berlin [u.a.], 2009.
- [24] Raúl Rojas. *Theorie der neuronalen Netze*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.
- [25] Paul Schönsleben. *Integriertes Logistikmanagement: Operations- und Supply-chain-Management in umfassenden Wertschöpfungsnetzwerken*. Springer, Berlin and Heidelberg and New York, 5., bearb. und erw. Aufl. edition, 2007.
- [26] Günther Schuh and Volker Stich. *Produktionsplanung und -steuerung 1*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [27] Stephan Seeck. *Erfolgsfaktor Logistik: Klassische Fehler erkennen und vermeiden*. SpringerLink : Bücher. Gabler Verlag / Springer Fachmedien Wiesbaden, Wiesbaden, Wiesbaden, 2010.
- [28] S. N. Sivanandam and S. N. Deepa. *Introduction to genetic algorithms*. Springer, 2007.
- [29] Helmut Wannenwetsch. *Integrierte Materialwirtschaft und Logistik*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [30] Hans-Peter Wiendahl. *Fertigungsregelung: Logistische Beherrschung von Fertigungsabläufen auf Basis des Trichtermodells : 42 Tabellen*. Hanser, München [u.a.], 1997.
- [31] Guoqiang Zhang, B. Eddy Patuwo, and Michael Y. Hu. Forecasting with artificial neural networks. *International Journal of Forecasting*, 14(1):35–62, 1998.
- [32] Helmut Zsifkovits. *Logistik*, volume 3673 : Betriebswirtschaftslehre, Technik, Ingenieurwesen of *Reihe Grundwissen der Ökonomik: Betriebswirtschaftslehre*. UTB, Stuttgart, 1., neue Ausg. edition, 2012.
- [33] Taiichi Ōno. *Toyota production system: Beyond large-scale production*. Productivity Press, Cambridge, Mass., 1988.

A Benötigte Daten

Stammdaten

- Artikeldaten
 - Artikelnummer
 - Artikelbezeichnung
 - Sicherheitsbestand
 - Mindestlosgröße
 - Rundungswert
 - durchschnittlicher Bedarf
 - Wert pro Stück
 - MTS/MTO
 - Schnelldreher/Langsamdreher
- Maschinendaten
 - Maschinenkennnummer
 - Maschinenbezeichnung
 - durchschnittliche Produktionsmenge pro Schicht
 - Rüstmatrix (Rüstzeiten für Umrüstung zwischen allen auf der Maschine produzierten Artikeln)
- Fertigungsversionen
 - Auf welchen Maschinen kann Artikel A gefertigt werden
 - Produktionskosten von Artikel A auf Maschine M

Bewegungsdaten

- Bedarfsdaten für den Betrachtungszeitraum
 - je Artikel und Tag
 - Unterscheidung zwischen Bedarf des Werksverbunds und dem externer Kunden
- Freigegebene Fertigungsaufträge
- Kapazitätsangebot je Maschine im Betrachtungszeitraum