
Optimierte Produktionsplanung für einen Stabwerkdemonstrator

Optimized Production Planning for a Load Carrying Demonstrator

Bachelor-Thesis von Thomas Reiser

September 2011



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Optimierung

Optimierte Produktionsplanung für einen Stabwerkdemonstrator
Optimized Production Planning for a Load Carrying Demonstrator

Vorgelegte Bachelor-Thesis von Thomas Reiser

1. Gutachten: PD. Dr. Ulf Lorenz
2. Gutachten: M.Sc. Thorsten Ederer

Tag der Einreichung:

Erklärung zur Bachelor-Thesis

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 28. September 2011

(Thomas Reiser)



Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	3
2.1	Mathematische Grundlagen	3
2.1.1	Lineare Optimierung	3
2.1.2	Diskrete Optimierung	4
2.1.3	Modellierungstechniken	5
2.1.4	Graphentheorie	6
2.2	Betriebswirtschaftliche Grundlagen	8
3	Mathematische Modellierung	11
3.1	Das Grundmodell	11
3.2	Modellerweiterungen	15
3.3	Modellübersicht	18
4	Praktische Umsetzung des Modells anhand eines Beispiels	21
4.1	Die Beispielinstantz	21
4.2	Mathematische Software	25
4.3	Eine Beispielrechnung	26
4.4	Verhaltensanalyse	29
5	Fazit	33
	Literaturverzeichnis	35
	Anhang: CD und Quellcode	37



1 Einleitung

Ziel dieser Bachelor-Thesis ist es, anhand eines Beispiels die Probleme der innerbetrieblichen Produktions- und Losgrößenplanung aufzuzeigen und mithilfe von mathematischen Optimierungsmethoden einen Lösungsansatz zu diskutieren. Sie entstand am Fachgebiet Optimierung an der TU Darmstadt unter PD. Dr. Ulf Lorenz, der in seiner Forschungstätigkeit insbesondere *Optimierung unter Unsicherheit* untersucht. Da die gänzliche Darstellung der optimierten Produktionsplanung unter Unsicherheit den Rahmen dieser Arbeit sprengen würde, werden an der ein oder anderen Stelle lediglich erste Schritte und Ausblicke in diese Richtung gegeben, die eine weitere Diskussion anregen sollen.

Da die Arbeitsgruppe Diskrete Optimierung rund um PD. Dr. Ulf Lorenz im Sonderforschungsbereich 805 eng mit den Maschinenbauern der TU Darmstadt kooperiert, wurde zur praktischen Veranschaulichung der Problematik ein fiktiver Betrieb gewählt, der Rikschas in verschiedenen Ausfertigungen produziert. Eine Rikscha ist ein aus Stäben zusammengebautes Transportmittel, das vor allem in Asien Verwendung findet. Traditionell werden Rikschas von Personen gezogen. Allerdings gibt es mittlerweile in Europa auch Rikschas, die von einem Fahrrad gezogen und hauptsächlich zu Tourismuszwecken eingesetzt werden. Ein Foto zur „klassischen“ Rikscha ist in Abbildung 1.1 zu sehen. Um den Maschinenbauern im Sonderforschungsbereich 805 Anknüpfungspunkte an diese Arbeit zu geben, wurden Rikschas als Stabwerkdemonstrator gewählt.



Abbildung 1.1: Ein Mann zieht eine Rikscha durch eine schmale Straße in Kolkata in der Nähe des Bahnhofs Sealdah. Dieses Foto wurde freundlicherweise von Arne Hückelheim zur Verfügung gestellt. [Hüc11]

Im Folgenden wird ein Mehrprodukt-Betrieb betrachtet, der aus den vorhandenen Rohstoffen unterschiedliche Produkte fertigen kann. Für die Rikschaproduktion werden Stäbe eingekauft und weiterverarbeitet, bevor die Fertigung der Rikschas beginnen kann. Üblicherweise werden in Unternehmen mehrere Produktvarianten auf der gleichen Maschine hergestellt und Produkte aus unterschiedlichen Fertigungsstufen gemeinsam weiterverarbeitet. Während dieses Wertschöpfungsprozesses treten verschiedene Fragestellungen auf, die das Unternehmen beantworten muss:

Was ist die optimale Losgröße? Damit ist gemeint, wie viele Einheiten eines Produktes unmittelbar nacheinander auf einer Maschine produziert werden sollten, bevor die Maschine umgerüstet wird um das nächste Produkt zu fertigen. Je größer die Lose sind, desto höher fallen die Lagerkosten für Zwischenlager aus. Je kleiner die Lose sind, desto öfter kommt es bei der Maschine zu Stillständen und es fallen höhere Rüstkosten und -zeiten an. Diese Arbeit stellt ein Modell vor, mit dem die optimale Lösung dieses Zielkonfliktes – niedrige Produktionskosten einerseits und niedrige Rüstkosten andererseits – berechnet werden kann.

In welcher Reihenfolge sollen die Produkte gefertigt werden? Diese Fragestellung ist eng mit der Bestimmung der optimalen Losgröße verwandt, da die Produktionsmenge der einzelnen Produkte die der Folgeprodukte implizit vorgibt. Um ein Endprodukt zu fertigen, müssen alle Vorprodukte bereits ausreichend hergestellt worden sein. Die Reihenfolge der Produktion spielt also eine entscheidende Rolle bei der Produktionsplanung.

Wann soll mit der Produktion begonnen werden? Je später die Produktion beginnt, desto geringer sind die Durchlaufzeiten des Wertschöpfungsprozesses und damit die Lagerkosten. Wird allerdings zu knapp mit der Produktion begonnen, so kann es aufgrund der Produktions- und Rüstzeiten passieren, dass die Produktion zu lange dauert und das Unternehmen in Lieferverzug gerät.

Wie groß darf die Nachfrage höchstens sein? Werden zu viele Aufträge angenommen, so ist es nicht mehr gewährleistet, dass der Betrieb pünktlich liefern kann. Dann treten Lieferrückstände auf und es muss entschieden werden, welche Aufträge noch bedient werden können und welche Aufträge storniert werden müssen.

Diese und weitere Fragen werden im Laufe der Arbeit diskutiert und beantwortet. Im zweiten Kapitel werden notwendige mathematische Grundlagen und Kenntnisse der Betriebswirtschaftslehre erläutert. Im Anschluss daran versucht Kapitel drei die obigen Fragestellungen mithilfe eines mathematischen Modells zu beantworten. Die Funktionsweise und das Verhalten dieses Modells wird in Kapitel vier praktisch anhand einer fiktiven Rikschafirma aufgezeigt. Schließlich wird es im fünften Kapitel bewertet und ein Ausblick auf die Verwendbarkeit des Modells gegeben.

2 Grundlagen

Dieses Kapitel führt die für das Verständnis dieser Arbeit notwendigen mathematischen und betriebswirtschaftlichen Grundkenntnisse ein. Zunächst werden die zentralen Begrifflichkeiten der linearen und diskreten Optimierung erläutert und im zweiten Teil die betriebswirtschaftlichen Grundlagen der Produktionsplanung, der Material-Logistik und der Losgrößenplanung vorgestellt.

2.1 Mathematische Grundlagen

Ziel der Optimierung ist es, aus einer Menge von Alternativen die beste auszuwählen. Mathematisch ausgedrückt bedeutet dies, Maxima oder Minima einer Funktion über einer Menge zu finden. Dieser Abschnitt beschäftigt sich genauer mit diesem Problem und stellt verschiedene Lösungsansätze vor. Zunächst werden lineare Programme eingeführt, die unter Zuhilfenahme mathematischer Software noch relativ leicht zu lösen sind. Im Anschluss daran wird genauer auf gemischt-ganzzahlige lineare Programme eingegangen. Eine Darstellung verschiedener Modellierungstechniken und ein Abschnitt über die Graphentheorie vervollständigen diesen Teil der Arbeit. Die meisten Definitionen sind aus den Vorlesungsskripten „Einführung in die Optimierung“ [BDMU11] und „Diskrete Optimierung“ [Mar11] der TU Darmstadt entnommen.

2.1.1 Lineare Optimierung

Es gibt verschiedene Möglichkeiten ein Optimierungsproblem mathematisch zu beschreiben. Das *lineare Optimierungsproblem* (englisch: Linear Program, LP) in Standardform lautet

$$\min c^T x \quad \text{s. t.} \quad Ax = b, x \geq 0 \quad (2.1)$$

mit $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m,n}$, $x \in \mathbb{R}^n$. Dabei heißen $x = (x_1, \dots, x_n)^T$ die nichtnegativen, reellen *Variablen* des Problems. Eine konkrete *Instanz* eines solchen Problems ist gegeben durch die *Parameter* A , b und c . Jede Variablenbelegung $x \in \mathbb{R}^n$, welche die linearen *Nebenbedingungen* $Ax = b, x \geq 0$ erfüllt, heißt *zulässige Lösung*. Dasjenige x^* , welches bezüglich der linearen *Zielfunktion* $\min c^T x$ das Minimum aller zulässiger Lösungen bildet, heißt *Optimallösung* von (2.1). Die Optimallösung muss dabei keinesfalls eindeutig sein, u.U. bilden unendlich viele Lösungen das Minimum über alle zulässigen Lösungen.

Typischerweise werden Probleme, die man betrachten möchte, nicht immer in der Standardform (2.1) angegeben sein. Es gibt aber verschiedene Modellierungstechniken, mit deren Hilfe sich jedes Problem in die Standardform überführen lässt. Einige dieser Techniken werden in Abschnitt 2.1.3 vorgestellt. Ein Vorteil der Standardform ist, dass die bekannteste Lösungsmethode für lineare Programme, der *Simplexalgorithmus*, als Eingabe ein Problem in Standardform erwartet.

Die Menge aller zulässigen Lösungen bildet ein *Polyeder* $P = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$. Jedes Polyeder ist eine (abgeschlossene) konvexe Menge, was bedeutet, dass mit je zwei Lösungen p_1 und p_2 , die in P liegen, auch die Verbindungsstrecke $\lambda p_1 + (1 - \lambda)p_2$ mit $\lambda \in [0, 1]$ wieder in P liegt. Ist ein Polyeder

zusätzlich noch beschränkt, also eine kompakte Menge, so heißt es auch *Polytop*. Ein Punkt $p \in P$ heißt *Ecke*, wenn er nicht als echte Konvexkombination von verschiedenen Punkten aus P dargestellt werden kann, d.h. wenn aus $p = \lambda p_1 + (1 - \lambda)p_2$ mit $p_1, p_2 \in P$ und $\lambda \in (0, 1)$ folgt $p = p_1 = p_2$.

Der Vorteil der linearen Optimierung besteht darin, dass eine lineare Zielfunktion über einem nichtleeren Polytop ihr (nicht unbedingt eindeutiges) Minimum und Maximum in (mindestens) einer Ecke annimmt. Man muss also lediglich alle Ecken des Polytops absuchen, um die Optimallösung zu finden. Genau dies ist der Grundgedanke des Simplexalgorithmus, dessen genaue Funktionsweise in zahlreicher Literatur nachgelesen werden kann [BDMU11]. Hat man allerdings ein unbeschränktes Polyeder vorliegen, so kann es passieren, dass es keine Optimallösung gibt. Dies ist genau dann der Fall, wenn die Zielfunktion in eine unbeschränkte Richtung verläuft, d.h. zu jeder zulässigen Lösung x lässt sich eine bessere Lösung $\tilde{x} = x - \delta c$ mit $\delta > 0$ konstruieren. Ein weiterer Sonderfall liegt dann vor, wenn in der Nebenbedingungsmatrix widersprüchliche Nebenbedingungen vorkommen. Dann ist die Menge der zulässigen Lösungen nur die leere Menge und das Problem ist unlösbar.

Der Nachteil der Simplexmethode liegt in der Laufzeit: Der Simplex-Algorithmus benötigt im „worst-case“ exponentielle Laufzeit. Der Nachweis, dass sich das Problem (2.1) dennoch in polynomialer Zeit lösen lässt, liefert die *Ellipsoidmethode* [BDMU11]. Allerdings herrscht zwischen Theorie und Praxis eine Diskrepanz, denn in der Praxis liefert der Simplexalgorithmus eine wesentlich bessere Performance als die Ellipsoidmethode, weshalb er auch überwiegend in Lösungssoftware (Solvem) verwendet wird. Eine weitere, sowohl in der Theorie als auch in der Praxis erfolgreiche Methode ist das sogenannte *Innere-Punkte-Verfahren*, welches ausgehend von einem inneren Punkt einen zentralen Pfad im Inneren des Polyeders in Richtung des Optimums sucht.

In vielen Anwendungen treten weitere Beschränkungen der Variablen auf, z.B. werden üblicherweise in der Produktionsplanung nur ganzzahlige Produktionsmengen erlaubt, oder eine Maschine hat einen der binären Zustände „an“ oder „aus“. Hat man Probleme dieser Art vorliegen, so spricht man von gemischt-ganzzahligen linearen Programmen, welche im folgenden Abschnitt diskutiert werden.

2.1.2 Diskrete Optimierung

Einige Optimierungsprobleme, wie z.B. das Rucksackproblem¹, bestehen im Gegensatz zum linearen Programm nur aus ganzzahligen Variablen. Ein solches *ganzzahliges Programm* (englisch: Integer Program, IP) kann wie folgt beschrieben werden:

$$\min c^T x \quad \text{s.t.} \quad Ax = b, x \geq 0, x \in \mathbb{Z}^n \quad (2.2)$$

Weitaus häufiger treten jedoch Probleme auf, bei denen ein Teil der Variablen kontinuierlich, der andere ganzzahlig ist. Dann spricht man von *gemischt-ganzzahligen linearen Programmen* (englisch: Mixed Integer Program, MIP), die wie folgt definiert sind:

$$\min c^T x \quad \text{s.t.} \quad Ax = b, x \geq 0, x \in \mathbb{Z}^{n-p} \times \mathbb{R}^p \quad (2.3)$$

mit $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m,n}$ und $p = \{0, \dots, n\}$. In dieser Formulierung sind für $p = 0$ und $p = n$ implizit die Definitionen des IP und LP enthalten.

¹ Ein Wanderer möchte auf eine Reise verschiedene Gegenstände mitnehmen. Er kann aber nur eine gewisse Last in seinem Rucksack tragen. Daher möchte er diejenigen Gegenstände auswählen, die ihm den Gesamtnutzen maximieren.

Bei MIPs ist der Simplexalgorithmus i.A. nicht direkt anwendbar, es sei denn, das Problem hat eine bestimmte Form². In der diskreten Optimierung gibt es daher andere Lösungsverfahren, die aber in Teilproblemen auf den Simplexalgorithmus zurückgreifen können. Die bekanntesten und effektivsten Lösungsverfahren sind der Branch&Bound-Algorithmus sowie der Branch&Cut-Algorithmus, welche nun genauer erläutert werden.

Vernachlässigt man die Ganzzahligkeitsbedingungen in (2.3), so erhält man die sogenannte *LP-Relaxierung*, die wiederum mit dem Simplexalgorithmus gelöst werden kann. Allerdings liefert diese nur eine untere Schranke für (2.3), da die LP-Lösung i.A. nicht ganzzahlig ist. Neben der LP-Relaxierung existieren weitere Relaxierungen, wie z.B. die Lagrange-Relaxierung. Auf der anderen Seite gibt es heuristische Verfahren, mit denen sich je nach Anwendungsfall schnell zulässige Lösungen finden lassen, die aber u.U. beliebig schlecht sein können. Beispielsweise seien an dieser Stelle der *Greedy-Algorithmus* zum Finden einer zulässigen Lösung erwähnt und die *Lokale Suche*, die ausgehend von einer zulässigen Lösung in einer Nachbarschaft bessere Lösungen sucht.

Der *Branch&Bound-Algorithmus* verbindet diese beiden Ansätze. Der gesamte Lösungsraum wird dabei iterativ mit Hilfe einer Baumstruktur durchsucht. Ausgehend von einer Lösung der LP-Relaxierung als untere Schranke (Bound) wird der Lösungsraum in zwei Teilprobleme aufgeteilt, die dann jeweils separat untersucht werden können (Branch). Je tiefer man im Baum kommt, desto „einfacher“ werden die Teilprobleme. Die Methode macht sich dabei zu Nutze, dass die Optimallösung keinesfalls schlechter als die bisher beste gefundene zulässige Lösung sein kann. Somit können gewisse Teilprobleme von vorneherein ausgelotet werden. Im „worst-case“ ist der Branch&Bound-Algorithmus ein vollständiges Enumerationsverfahren. In der Praxis funktioniert er allerdings gut und wird von vielen Solvern verwendet.

Verwendet man zusätzlich zur LP-Relaxierung *Schnittebenenverfahren*, wie z.B. Gomory-Schnitte [Mar11], so spricht man von *Branch&Cut-Verfahren*. Der CPLEX-Solver [CPL11], auf den am Ende der Arbeit zum Lösen des Problems der Produktionsplanung zurückgegriffen wird, verwendet u.a. einen solchen Branch&Cut-Algorithmus.

2.1.3 Modellierungstechniken

Im späteren Modellierungsteil der Arbeit werden einige besondere Techniken verwendet. Zum besseren Verständnis wird an dieser Stelle auf ein paar grundlegende Techniken eingegangen. Zunächst werden einige Umformungen vorgestellt. Im Anschluss wird erläutert, wie man logische „wenn-dann“ Beziehungen realisieren kann.

In Abschnitt 2.1.1 wurde schon darauf hingewiesen, dass ein LP häufig nicht in Standardform vorliegt. Man kann aber jedes Problem in die Form von (2.1) überführen. So kann ein Maximierungsproblem in ein Minimierungsproblem umgeschrieben werden, indem man das Inverse der Zielfunktion bildet:

$$\max c^T x = - \min (-c^T x).$$

Ungleichungen können zu Gleichungen gemacht werden, indem man sogenannte *Schlupfvariablen* $y = (y_1, \dots, y_m)^T$ einführt (I bezeichnet die $m \times m$ Einheitsmatrix):

$$Ax \leq b \Rightarrow Ax + Iy = b, y \geq 0$$

² z.B. Totale Unimodularität, Totale Duale Integralität

Auch die Nichtnegativitätsbedingungen an den Variablenvektor x stellen keine Einschränkung dar, da man x in seinen Positiv- und Negativanteil zerlegen kann, die jeweils für sich genommen positiv sind:

$$x = x^+ - x^- \text{ mit } x^+, x^- \geq 0$$

Möchte man anstelle von Gleichungen lieber Ungleichungen haben, so lässt sich dies ebenfalls realisieren:

$$Ax = b \Rightarrow Ax \leq b \wedge Ax \geq b$$

Im späteren Teil der Arbeit treten vorwiegend ganzzahlige Variablen auf. Eine besondere Variante davon sind *Binärvariablen*, die nur zwei Werte (null oder eins) annehmen können. Binärvariablen treten in der Praxis immer dann auf, wenn „Ja-Nein-Entscheidungen“ getroffen werden müssen. Möchte man zwei Binärvariablen x und y aneinander koppeln, z.B. durch die logische Aussage „wenn x , dann auch y “, so lässt sich dies wie folgt realisieren:

$$x \leq y, \quad x, y \in \{0, 1\}$$

Wenn x auf null gesetzt wird, so kann y null oder eins gewählt werden, falls x aber auf eins gesetzt wird, dann muss y auch auf eins gesetzt werden.

Mit dieser Technik sind auch komplexere Zusammenhänge darstellbar. Möchte man z.B. erreichen, dass die Produktionsmenge q_p eines Produktes p von der Bereitschaft von Maschine x und der maximalen Produktionsmenge C abhängt, so spiegelt dies der folgende Zusammenhang wieder:

$$q_p \leq xC, \quad x \in \{0, 1\}$$

Ist die Maschine x nicht bereit Produkt p zu produzieren ($x = 0$), so wird die Produktionsmenge auf null gedrückt. Andernfalls können bis zur maximalen Produktionsmenge viele Einheiten produziert werden. Dieser und weitere Produktionszusammenhänge werden im nächsten Kapitel beschrieben.

2.1.4 Graphentheorie

Da im Laufe der späteren Modellierung Graphen verwendet werden, um die Beziehung zwischen den einzelnen Produkten zu verdeutlichen, werden in diesem Abschnitt die notwendigen graphentheoretischen Grundlagen erläutert. Die Definitionen dazu kann man unter [Aig06] nachlesen.

Ein *Graph* ist ein geordnetes Paar $G = (V, E)$ bestehend aus einer nichtleeren, endlichen *Knotenmenge* V und einer *Kantenmenge* E . Man unterscheidet gerichtete und ungerichtete Graphen. Bei ungerichteten Graphen haben die Kante keine Orientierung, d.h. $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$. Bei gerichteten Graphen sind die Elemente aus E geordnete Paare von Elementen aus V , d.h. $E \subseteq \{(u, v) \mid u, v \in V, u \neq v\}$. Zur optischen Veranschaulichung der Definition siehe Abbildung 2.1.

Eine besondere Familie von ungerichteten Graphen sind *Bäume*, die nach Definition zusammenhängend und kreisfrei sind sowie einen ausgewiesenen Knoten w besitzen, den man Wurzel nennt. Sie finden häufig in hierarchischen Strukturen Verwendung, wie z.B. in Stammbäumen. Die Baumstruktur machen sich viele Algorithmen zunutze, wie z.B. der Branch&Bound-Algorithmus, dessen Funktionsweise schon erläutert wurde.

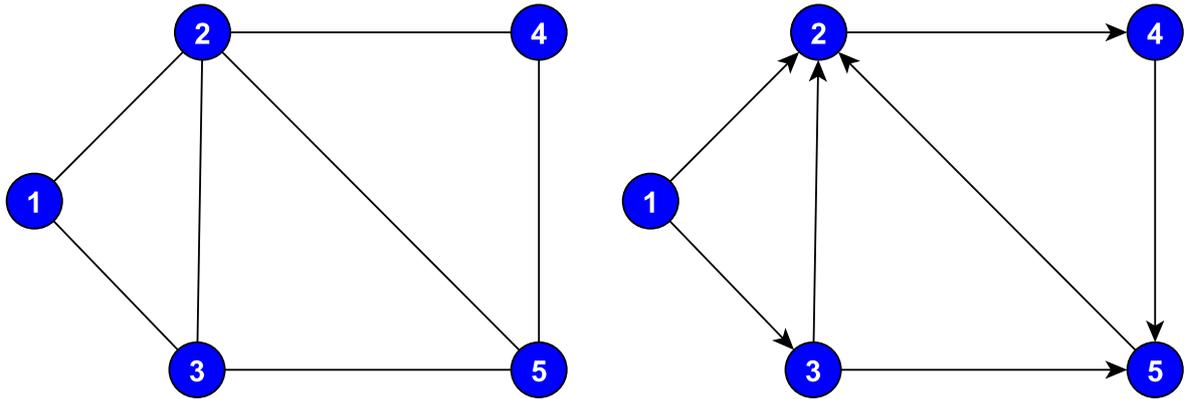


Abbildung 2.1: Beispiele für Graphen. Links ein ungerichteter Graph $G = (V, E)$ mit $V = \{1, 2, 3, 4, 5\}$ und $E = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 5\}, \{4, 5\}\}$. Rechts ein gerichteter Graph $\tilde{G} = (V, \tilde{E})$ mit $\tilde{E} = \{(1, 2), (1, 3), (3, 2), (2, 4), (5, 2), (3, 5), (4, 5)\}$. Die Pfeilrichtungen geben die Orientierung der Kanten wieder.

Häufig werden Flussprobleme in Netzwerken mithilfe von Graphen modelliert. Ein *Netzwerk* besteht aus einem gerichteten Graphen $G = (V, E)$, einem ausgezeichneten Start- und Zielknoten s und t mit $s \neq t$ und einer Gewichtsfunktion $c : E \rightarrow \mathbb{R}_+$, die jeder Kante eine nichtnegative Zahl zuordnet. Man nennt s Quelle, t Senke und c die Kapazität des Netzwerkes. Ein *Fluss* f in einem Netzwerk ist eine Funktion $f : E \rightarrow \mathbb{R}_+$. Der Netto-Fluss in einem Knoten $v \in V$ ist gegeben durch

$$(\delta f)(v) = \sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e)$$

d.h. $(\delta f)(v)$ misst die Differenz des Flusses aller eingehenden Kanten $\delta^+(v)$ und aller ausgehenden Kanten $\delta^-(v)$ in Knoten v . Um einen zulässigen Fluss von s nach t zu erhalten, müssen an f zwei Bedingungen gestellt werden:

$$\begin{aligned} 0 \leq f(e) \leq c(e) \quad \forall e \in E \\ (\delta f)(v) = 0 \quad \forall v \in V \setminus \{s, t\} \end{aligned}$$

Die erste Bedingung fordert, dass der Fluss entlang einer jeden Kante die Nichtnegativitätsbedingung und die Kapazitätsbeschränkung einhält. Die zweite Bedingung, auch *Flusserhaltung* genannt, drückt aus, dass alles, was in einen Knoten hineinfließt auch wieder herausfließen muss, wobei der Start- und Zielknoten ausgenommen sind. Dadurch wird sichergestellt, dass unterwegs kein Fluss verloren geht oder aus dem Nichts entsteht. Der *Wert eines Flusses* ist also der Nettofluss, der an der Senke t ankommt bzw. aus der Quelle s herausfließt: $w(f) = (\delta f)(t) = -(\delta f)(s)$. Die klassische Fragestellung zu Netzwerken, der sich verschiedenste Algorithmen widmen, beschäftigt sich mit dem Finden maximaler Flüsse. In dieser Arbeit wird von der Netzwerktheorie jedoch nur die Flusserhaltung benötigt. In einer der Nebenbedingungen werden später die Lagerbestände betrachtet. Dort muss sichergestellt werden, dass die Lagerbestände der einzelnen Produkte logisch zusammenhängen und nicht etwa ein paar Produkte unterwegs verschwinden oder plötzlich auftauchen.

Möchte man einen Graphen in den Computer eingeben, so eignen sich zwei spezielle Darstellungsformen. Die *Adjazenzmatrix* ist eine quadratische Matrix $A = (a_{ij})$ mit

$$a_{ij} = \begin{cases} 1 & \text{falls } (v_i, v_j) \in E \\ 0 & \text{sonst} \end{cases}$$

Im Falle eines ungerichteten Graphen ist A also eine symmetrische Matrix. Die Adjazenzmatrizen für den ungerichteten und den gerichteten Graphen aus Abbildung 2.1 sehen wie folgt aus:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Sind für die Kanten Kapazitäten vorgegeben, so kann man diese Werte anstelle der Einsen in die Matrix eintragen. Eine weitere Darstellungsform für gerichtete Graphen ist die Knoten-Kanten-Inzidenzmatrix (kurz *Inzidenzmatrix*) $B = b_{ij}$, die in den Zeilen die Knoten i und in den Spalten die Kanten j speichert und die Richtung der Kanten wie folgt wiedergibt:

$$b_{ij} = \begin{cases} 1 & \text{falls } e_j = (v, i) \text{ für ein } v \in V \setminus \{i\}, \\ -1 & \text{falls } e_j = (i, v) \text{ für ein } v \in V \setminus \{i\}, \\ 0 & \text{sonst} \end{cases}$$

Ein Knoten $v \in V$ und eine Kante $e \in E$ heißen also inzident, wenn der Knoten entweder als Ausgangs- oder als Eingangsknoten in der Kante enthalten ist. Wählt man die Kantenreihenfolge wie in der Kantendefinition, sieht die Inzidenzmatrix für den gerichteten Graphen aus Abbildung 2.1 wie folgt aus:

$$\begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 \end{pmatrix}$$

Damit sind alle nötigen mathematischen Grundlagen eingeführt und werden im Folgenden, vor allem im nächsten Kapitel, als gegeben vorausgesetzt. Bevor das Produktionsplanungsproblem modelliert wird, folgt ein Abschnitt über betriebswirtschaftliche Grundlagen, der dieses Kapitel abschließt.

2.2 Betriebswirtschaftliche Grundlagen

Dieser Teil der Arbeit stellt die üblichen Sprechweisen und Definitionen vor, die in der Betriebswirtschaft bei der Produktionsplanung verwendet werden. In diesem Abschnitt werden auf die Werke von Tempelmeier [Tem06], Thommen und Achleitner [TA03] und Oßwald [Oßw79] zurückgegriffen, in denen die hier erwähnten Definitionen nachgelesen werden können.

Der Oberbegriff Produktionsplanung beinhaltet die beiden Teilgebiete *Produktionsprogrammplanung* und *Produktionsablaufplanung*. Unter dem *Produktionsprogramm* versteht man die Gesamtheit aller von einem Unternehmen zu erstellenden Güter und Leistungen. Die Festlegung des Produktionsprogramms trifft somit die Entscheidung, welche Produkte hergestellt werden sollen und hat langfristige Auswirkungen. Das *Absatzprogramm* bildet den Haupteinflussfaktor dieser Entscheidung. In einem markt-orientierten Unternehmen sind dies die von den Kunden nachgefragten Erzeugnisse. Hat man sich für ein Produktionsprogramm entschieden, so folgt die *Ablaufplanung*, deren Ziel es ist, die Gestaltung des Fertigungsprozesses in Bezug auf Auftrag, Zeit und Kapazität festzulegen. Sie hat dafür zu sorgen, dass folgende drei Grundsätze eingehalten werden [TA03]:

-
1. **Prinzip der Termineinhaltung:** Dieser Grundsatz beinhaltet die optimale Abstimmung der Fertigungstermine mit den Auftragsterminen.
 2. **Prinzip der Zeitminimierung:** Dieses Prinzip verlangt, die Durchlaufzeiten des zu bearbeitenden Materials so zu gestalten, dass möglichst keine Wartezeiten entstehen, in denen das Material nicht bearbeitet wird.
 3. **Prinzip der Kapazitätsauslastung:** Dieser Grundsatz fordert eine möglichst hohe Kapazitätsauslastung und damit eine Minimierung der Leerzeiten, in denen Betriebsmittel und Arbeitskräfte nicht genutzt werden.

Überlicherweise lassen sich Grundsatz zwei und drei nur selten gleichzeitig verwirklichen. Dieser Zielkonflikt wird in Fachkreisen auch *Dilemma der Ablaufplanung* genannt. Das eigentliche Ziel der Ablaufplanung besteht somit in einer optimalen Abstimmung dieser beiden Forderungen.

Die für die Produktion notwendigen Materialien lassen sich in folgende Gruppen einteilen:

- **Rohstoffe**, die als Grundmaterial unmittelbar in das Produkt eingehen (z.B. Eisenstangen bei der Rikscha-Produktion),
- **Hilfsstoffe**, die ebenfalls in das Erzeugnis eingehen, aber nur ergänzenden Charakter haben (z.B. Schrauben),
- **Betriebsstoffe**, die bei der Produktion verbraucht werden, aber keinen Bestandteil des Fertigproduktes bilden (z.B. Schmierfette für Maschinen),
- **Halbfabrikate**, die als Teile oder Baugruppen in das Endprodukt eingehen und die sich von den Hilfsstoffen durch einen höheren Reifegrad unterscheiden,
- **Handelswaren**, die als Ergänzung des Produktionsprogramms nicht in den Produktionsprozess eingehen, sondern unbearbeitet weiterverkauft werden.

Mit der Beschaffung und Verarbeitung von Roh-, Hilfs- und Betriebsstoffen befasst sich die *Material-Logistik*. Deren Hauptaufgabe ist es, den räumlichen, zeitlichen und mengenmäßigen Transfer der in der Produktion eingesetzten Verbrauchsfaktoren zu organisieren. Die Planung dieses Güterflusses von den Rohstoffen bis hin zum fertigen Endprodukt ist auch unter dem Begriff *Supply Chain Management* bekannt. Im Laufe dieses Güterflusses sind üblicherweise Lager nötig, um einen Puffer für die Materialien und Produkte anzulegen. Sie lassen sich in drei verschiedene Lagerstufen einteilen [TA03]:

1. **Eingangslager**, welche die Rohstofflieferungen auffangen, wenn der momentane Bedarf in der Fertigung kleiner ist als der Güterzufluss oder der zukünftige Bedarf größer ist als der zukünftige Güterzufluss.
2. **Zwischenlager**, die während des Produktionsprozesses entstehen und als Puffer zwischen den verschiedenen Fertigungsstufen fungieren. Der Bestand der Zwischenlager hängt sehr stark von der Lösung des Dilemmas der Ablaufplanung ab. Je stärker das Prinzip der maximalen Kapazitätsauslastung verfolgt wird, desto eher entstehen Zwischenlager.
3. **Fertigwarenlager**, die nach Beendigung des Produktionsprozesses entstehen. Diese Lager fangen zeitliche Abweichungen zwischen Produktions- und Absatzmenge auf.

Die Höhe der Lagerbestände wird vor allem durch Entscheidungen der *Losgrößenplanung* beeinflusst: Als Fertigungslos bezeichnet man die Menge einer Sorte oder einer Serie, die ohne Unterbrechung des Produktionsprozesses auf einer Maschine hergestellt wird. Losgrößenprobleme treten immer dann auf, wenn mit der Umstellung einer Maschine für den Produktionsprozess Rüstzeiten und/oder Rüstkosten anfallen. Je mehr einer Sorte am Stück produziert wird, desto geringer sind die Rüstkosten pro Einheit, desto größer sind allerdings im Umkehrschluss die Zwischenlager, die für die produzierte Menge bereitgestellt werden müssen. Die Lösung dieser widersprüchlichen Ziele, nämlich einerseits die Rüstkosten durch höhere Produktionszahlen niedrig zu halten und andererseits hohe Lagerkosten zu vermeiden, wird später in der Arbeit diskutiert.

Die Losgrößen- und Materialbedarfsplanung greift auf vier wichtige Informationsquellen zurück [Tem06]:

1. das **Produktionsprogramm**, welches festlegt, was produziert werden soll,
2. den **Erzeugniszusammenhang**, also welche und wie viele Einzelteile und Baugruppen in den einzelnen Produktionsschritten benötigt werden,
3. die **Durchlaufzeiten** bzw. **Beschaffungszeiten** der Erzeugnisse, welche die Produktionsdauer beeinflussen,
4. den **periodenbezogenen Lagerbeständen**, also die Entscheidung, wie viel produziert werden soll.

Auch im später verwendeten Modell sind diese Informationen als Datengrundlage vorgegeben und die Losgröße wird daraufhin optimiert. Zur Veranschaulichung des Erzeugniszusammenhangs der einzelnen Produkte gibt es verschiedene Darstellungsformen. In dieser Arbeit werden dazu *Gozinto-Graphen* verwendet. Der Begriff Gozinto-Graph geht auf einen fiktiven italienischen Mathematiker mit dem Namen „Zepartzat Gozinto“ zurück [Tem06]. Ein lautmalerischer Vergleich mit den englischen Worten „the part that goes into“ beschreibt anschaulich, was mit der Bezeichnung Gozinto-Graph gemeint ist: Die Erzeugnisse werden durch die Knoten repräsentiert. Eine Kante zwischen zwei Knoten beschreibt die Input-Output-Beziehung zwischen den Erzeugnissen. Die Kantenbewertung einer Kante zwischen Knoten i und j gibt an, wie viele Einheiten von Produkt i benötigt werden, um eine Einheit von Produkt j herzustellen. Diese Größe bezeichnet man als *Direktbedarfskoeffizient*.

Hat man das Problem der Losgrößenplanung gelöst, so muss sich das Unternehmen in einem nächsten Schritt überlegen, welche Erlöse erzielt werden müssen, um wirtschaftlich zu produzieren. Der Gewinn einer Unternehmung setzt sich zusammen aus den Erlösen E für den Verkauf eines Produktes abzüglich der Produktionskosten K . Nimmt man an, dass x Einheiten produziert und auch abgesetzt werden können, so lässt sich der Gewinn G als Funktion von x schreiben:

$$G(x) = Ex - Kx$$

Falls die Produktionsmenge x vorgegeben ist, so lässt sich daraus eine Preisuntergrenze bestimmen. Geht man davon aus, dass die Produktionskosten bekannt sind, so muss man lediglich den Gewinn auf null setzen und die Gleichung nach E auflösen, um den Preis zu bestimmen, den man mindestens verlangen muss, damit man keine Verluste erwirtschaftet. Dieses Vorgehen wird *Break-Even-Analyse* oder *Gewinnschwellenanalyse* genannt.

Damit sind alle nötigen Grundlagen eingeführt, um das nachfolgende Modell zu verstehen und im Anschluss daran zu analysieren.

3 Mathematische Modellierung

In diesem Kapitel wird die mathematische Modellierung, mit deren Hilfe die Produktionsplanung von Rikschas gelöst werden soll, vorgestellt. Dabei wird auf ein bereits bestehendes Modell [WS07] zurückgegriffen, das im Laufe dieses Kapitels angepasst und erweitert wird. Im ersten Abschnitt wird das von Sven Grothklags, Ulf Lorenz und Jan Wesemann entwickelte Modell ausführlich vorgestellt. Es wurde aus dem Paper mit dem Titel „A Stochastic Lot-Sizing and Scheduling Model“ entnommen [WS07, S. 491-496]. Im zweiten Abschnitt werden einige notwendige Erweiterungen, ohne die das Modell nicht korrekt funktioniert, Schritt für Schritt eingeführt und ein Ausblick auf Erweiterungsmöglichkeiten gegeben. Abschließend wird das Modell mit den vorgestellten Erweiterungen noch einmal im Ganzen dargestellt.

3.1 Das Grundmodell

Das hier vorgestellte Modell ermöglicht die Bestimmung der optimierten mehrstufigen und kapazitätsbeschränkten Losgrößen- und Ablaufplanung unter folgenden Annahmen:

Verschiedene Produkte mit jeweils bekannter Nachfrage müssen produziert werden. Ein Gozinto-Graph spiegelt die „Ist-ein-Teil-von“-Beziehung zwischen den einzelnen Produkten wieder. Sollte die Nachfrage nicht fristgerecht erfüllt werden können, so tritt entweder Lieferverzögerung oder Lieferausfall auf. Außerdem werden positive Transportzeiten für die Zwischenprodukte zu den einzelnen Prozessen berücksichtigt.

In den einzelnen Produktionsstufen kommen mehrere Maschinen zum Einsatz, die zum Teil verschiedene Produkte herstellen können. In der Zeit, in der ein bestimmtes Zwischen- oder Endprodukt auf einer Maschine produziert wird, kann diese Maschine kein anderes Produkt herstellen. Ein Produkt kann auf verschiedenen Maschinen mit unterschiedlichen Produktionsgeschwindigkeiten gefertigt werden. Auch die Produktionsgeschwindigkeit auf ein und der selben Maschine kann im Zeitverlauf variieren.

Die Produktion eines Produktes kann nur dann erfolgen, wenn die Maschine in einem entsprechenden Zustand ist. Wird zwischen verschiedenen Zuständen gewechselt, so fallen Setupkosten und Setupwechselzeiten an, welche vom alten und neuen Zustand abhängen können. Ein Zustand bleibt so lange erhalten, bis der nächste Setupwechsel erfolgt. Eine zusätzliche Bedingung, die im Modell mit aufgenommen wurde, ist, dass Setupwechsel nur zu Periodenbeginn erlaubt sind. Daraus ergibt sich unmittelbar, dass jede Maschine in jeder Zeiteinheit nur Produkte maximal eines Typs produzieren kann.

Der Planungshorizont ist in eine endliche Anzahl von Zeitschritten unterteilt. Rohstoffe, Halbfabrikate und Endprodukte, die zur Bedarfsdeckung produziert werden, müssen in einem Zwischenlager aufbewahrt werden und verursachen produktspezifische Lagerkosten.

All diese Bedingungen werden im nun folgenden Optimierungsproblem berücksichtigt. In Tabelle 3.1 sind alle Variablen beschrieben, die zur Modellierung benötigt werden. Des Weiteren werden in Tabelle 3.2 die benötigten Parameter eingeführt. Im Anschluss daran lässt sich mit diesen Variablen und Parametern das MIP formulieren, welches das Losgrößen- und Ablaufplan-Problem löst.

Tabelle 3.1: Benötigte Variablen

Symbol	Definition
$q_{i,t}$	Produktionsmenge (Quantity) von Produkt P_i durch Prozess i in Periode t
$s_{p,t}$	Lagerbestand (Stock) von Produkt p am Ende von Periode t
$b_{p,t}$	Verzugsmenge/Lieferrückstand (Backlog) von Produkt p am Ende von Periode t
$o_{p,t}$	Fehlmenge/Lieferausfall (Shortfall) von Produkt p in Periode t
$x_{i,j,t}$	Binärvariable, die anzeigt, ob ein Setupwechsel zwischen Prozess i und j am Anfang von Periode t auftritt ($x_{i,j,t} = 1$), oder nicht ($x_{i,j,t} = 0$)
$y_{i,t}$	Binärvariable, die anzeigt, ob Maschine M_i bereit ist für Prozess i am Ende von Periode t ($y_{i,t} = 1$) oder nicht ($y_{i,t} = 0$)

Tabelle 3.2: Benötigte Mengen und Parameter

Symbol	Definition
P	Menge von Produkten
M	Menge von Maschinen
I	Menge von Prozessen (Arbeitsschritten)
T	Anzahl von Perioden ($1 \dots T$)
M_i	Maschine, die für Prozess i benötigt wird
P_i	Das von Prozess i produzierte Produkt
I_i	$= \{j \in I : M_j = M_i, j \neq i\}$; Menge von Prozessen, die die gleiche Maschine wie Prozess i benutzen
S_p	Lagerkosten (Stockcosts) für eine Einheit von Produkt p pro Zeitschritt
B_p	Kosten für den Lieferverzug (Backlog) einer Einheit von Produkt p pro Zeiteinheit
O_p	Strafkosten für den Lieferausfall (Shortfall) einer Einheit von Produkt p
$X_{i,j}$	Setupkosten, um zwischen Prozess i und j zu wechseln (auf Maschine $M_i = M_j$)
Q_i	Produktionskosten einer Einheit von Produkt P_i durch Prozess i
$D_{p,t}$	externe Nachfrage (Demand) nach Produkt p in Periode t
$A_{p,q}$	Gozinto-Faktor: Null, falls Produkt q kein unmittelbarer Nachfolger von Produkt p ist, ansonsten die Anzahl von Produkt p , die benötigt wird, um eine Einheit von Produkt q zu produzieren
$\Delta_{p,i}$	(ganzzahlige) Anzahl von Zeiteinheiten, um Produkt p zu Maschine M_i zu transportieren (für Prozess i)
$C_{i,t}$	(Maximale) Produktionsmenge von Produkt P_i mit Prozess i in Periode t
$T_{i,j}$	(gebrochenrationale) Anzahl Perioden um zwischen Prozess i und j zu wechseln
$R_{i,j}$	$= \lfloor T_{i,j} \rfloor$; Anzahl kompletter Perioden, die für den Setupwechsel benötigt werden
$R_{i,j}^f$	$= T_{i,j} - R_{i,j}$; Rest der $(R_{i,j} + 1)$ -ten Periode nach dem Wechseln von Prozess i zu Prozess j

$$\min \sum_{t=1}^T \left(\sum_{p \in P} (S_p s_{p,t} + B_p b_{p,t} + O_p o_{p,t}) + \sum_{\substack{i \in I \\ j \in I_i}} X_{i,j} x_{i,j,t} + \sum_{i \in I} Q_i q_{i,t} \right) \quad (3.1)$$

$$\text{s. t. } s_{p,t-1} + \sum_{i:P_i=p} q_{i,t} - D_{p,t} - \sum_{\substack{q \in P \\ j:P_j=q}} A_{p,q} q_{j,t+\Delta_{p,j}} + b_{p,t} + o_{p,t} - b_{p,t-1} = s_{p,t} \quad \forall p \in P, \\ t = 1 \dots T \quad (3.2)$$

$$b_{p,t} + o_{p,t} - b_{p,t-1} \leq D_{p,t} \quad \forall p \in P, \\ t = 1 \dots T \quad (3.3)$$

$$\sum_{i:M_i=m} y_{i,t} = 1 \quad \forall m \in M, \\ t = 1 \dots T \quad (3.4)$$

$$y_{i,t-1} + y_{j,t} - 1 \leq x_{i,j,t} \quad \forall i \in I, j \in I_j, \\ t = 1 \dots T \quad (3.5)$$

$$C_{i,t} \left(y_{i,t} - \sum_{\substack{j \in I_i \\ t-R_{j,i} < t' \leq t}} x_{j,i,t'} - \sum_{j \in I_i} R_{j,i}^f x_{j,i,t-R_{j,i}} \right) \geq q_{i,t} \quad \forall i \in I, \\ t = 1 \dots T \quad (3.6)$$

$$q_{i,t} \geq 0; \quad s_{p,t}, b_{p,t}, o_{p,t} \geq 0 \quad \forall i \in I, p \in P, \\ t = 1 \dots T \quad (3.7)$$

$$x_{i,j,t} \in \{0, 1\} \quad \forall i \in I, j \in I_i, \\ t = 1 \dots T \quad (3.8)$$

$$y_{i,t} \in \{0, 1\} \quad \forall i \in I, \\ t = 1 \dots T \quad (3.9)$$

Die Zielfunktion (3.1) minimiert die Summe aller Kosten. Im Einzelnen sind dies die Lagerkosten, Kosten für Lieferverzögerung und -ausfall, Kosten für Setupänderungen und die Produktionskosten.

Gleichung (3.2) stellt das Lagergleichgewicht dar. Der Lagerbestand jedes Produktes zum Zeitpunkt t setzt sich zusammen aus dem Lagerbestand der Vorperiode ($s_{p,t-1}$) plus der Produktionsmenge des jeweiligen Produktes, abzüglich des Fremd- und Eigenbedarfs. Weiterhin müssen die Lieferrückstände und -ausfälle hinzuaddiert werden, da sie in der aktuellen Periode den externen Bedarf verringern. Die Lieferrückstände der Vorperiode ($b_{p,t-1}$) müssen abgezogen werden. Diese Produkte müssen nachträglich produziert werden – entweder in der aktuellen oder in einer späteren Periode. Erwähnenswert ist noch, dass für den internen Bedarf an Zwischenprodukten positive Transportzeiten $\Delta_{p,j}$ berücksichtigt werden. Soll ein Nachfolgeprodukt durch Prozess j produziert werden, so müssen alle Vorprodukte $p \in P$, die unmittelbar in das Nachfolgeprodukt eingehen, $\Delta_{p,j}$ Perioden zuvor aus dem Lager geholt werden. In diesen $\Delta_{p,j}$ Perioden treten für Produkt p im Modell keine Lagerkosten mehr auf, allerdings können die Transportkosten den Produktionskosten von Prozess j aufgeschlagen werden.

Die Ungleichung (3.3) besagt, dass die Veränderung der Verzugsmenge und der Fehlmenge zu keinem Zeitpunkt den externen Bedarf übersteigen darf. Zusammen mit der Gleichung (3.2) stellt diese Ungleichung sicher, dass der interne Bedarf in jeder Periode gedeckt werden kann und somit die Produktion garantiert ist. Sollte der Bedarf in einer Periode nicht gedeckt werden können, so kann das Modell zwischen zwei

Varianten, die unterschiedliche Kosten verursachen, entscheiden: Entweder wird der Bedarf später noch gedeckt (in diesem Falle treten in der Zielfunktion für jede weitere Periode, in der nicht geliefert werden kann, Strafkosten $B_p b_{p,t}$ für den Lieferverzug auf), oder der Auftrag wird storniert (einmalige, sehr hohe Strafkosten $O_p o_{p,t}$ für den Vertragsbruch). Der Vorteil dieser Modellierung ist, dass es für jede beliebige Problem Instanz auf jeden Fall zulässige Lösungen gibt, selbst wenn der Bedarf unrealistisch hoch wird.

Die Gleichung (3.4) zusammen mit (3.9) stellt sicher, dass jede Maschine m am Ende eines jeden Zeitpunktes in genau einem Zustand ist. Dafür wird für jeden Zeitpunkt über alle Prozesse summiert, die auf derselben Maschine bearbeitet werden. Die Gleichung fordert, dass von diesen Prozessen genau eine der $y_{i,t}$ -Variablen eins gesetzt wird. Dies bedeutet, dass pro Zeiteinheit nur einer der Prozesse, die um dieselbe Maschine konkurrieren, ablaufen kann.

Ungleichung (3.5) zusammen mit den Binärbeschränkungen (3.8) und (3.9) stellt Setupänderungen fest. Wenn eine Maschine am Ende von Periode $t - 1$ im Zustand i war ($y_{i,t} = 1$) und ein Setupwechsel von Prozess i zu j zu Beginn von Periode t stattgefunden hat ($y_{j,t} = 1$), dann fixiert Ungleichung (3.5) $x_{i,j,t}$ zu eins. In allen anderen Fällen kann $x_{i,j,t}$ null werden und da Setupwechsel Kosten verursachen, wird die Zielfunktion (3.1) dies auch anstreben. Aus diesem Grunde kann die Nebenbedingung (3.8) auch durch $x_{i,j,t} \geq 0$ ersetzt werden, ohne den optimalen Zielfunktionswert zu verändern.

Bleiben schließlich noch die Produktionsbeschränkungen, die durch Ungleichung (3.6) festgelegt werden. Zunächst bleibt festzuhalten, dass ein Prozess i nur dann Produkt P_i produzieren kann, wenn sich Maschine M_i in Periode t im entsprechenden Zustand befindet ($y_{i,t} = 1$). Wenn alle auftretenden $x_{i,j,t}$ Variablen null sind, also keine Setupänderung zum Zeitpunkt t stattfindet, können in dieser Periode bis zu $C_{i,t}$ Produkte hergestellt werden. Da im Modell Setupänderungszeiten berücksichtigt werden, muss sichergestellt werden, dass der Setupwechsel vom vorherigen Prozess j zu i bereits vor Zeitpunkt t abgeschlossen ist oder in der entsprechenden Periode abgeschlossen wird. Die erste Summe reduziert die Produktionsmenge in Perioden, die vollständig für Setupwechsel vorgesehen sind, auf null. Sollte ein Setupwechsel nicht zufällig eine ganzzahlige Periodenanzahl benötigen, so wird ein Wechsel typischerweise mitten in einer Periode fertig. Im verbleibenden Teil dieser Periode kann schon mit der Produktion des neuen Produktes begonnen werden. Dies ermöglicht die zweite Summe, die den Teil der Produktionszeit (und damit der Produktionsmenge) wegschneidet, der in der Periode t noch für den Setupwechsel $x_{j,i,t}$ benötigt wird. Ungleichung (3.7) erzwingt, dass die linke Seite von (3.6) nicht negativ wird. Deshalb müssen alle $x_{i,j,t}$ Variablen null sein, falls $y_{i,t}$ null ist und höchstens eine $x_{i,j,t}$ Variable kann eins werden, falls $y_{i,t}$ eins ist. Dies stellt sicher, dass eine Maschine während der gesamten Setupwechselzeit in ihrem Zustand j bleibt und erst nach Beendigung der Rüstzeit in den Zustand i wechselt. Weiterhin stellt sie sicher, dass sich keine zwei Setupwechsel überschneiden.

Die Ungleichungen in (3.7) sind die Nichtnegativitätsbedingungen für Produktionsmengen, Lagerbestände, Lieferrückstände und -ausfälle. Die Bedingungen in (3.8) und (3.9) definieren die Binärvariablen für die Setupvariablen und -wechsel. Die etwas unglückliche Zusammenfassung von Prozessen und Produkten in (3.7) wurde vom Originalmodell übernommen. Im Folgenden werden diese Ungleichungen der Übersichtlichkeit halber getrennt.

Im Laufe der Arbeit hat sich herausgestellt, dass dieses Modell noch ein paar zusätzliche Bedingungen benötigt, ohne die es in der Praxis nicht lauffähig ist. Im nächsten Abschnitt werden diese Nebenbedingungen eingeführt und auf zusätzliche interessante Erweiterungen eingegangen, die auch im nächsten Kapitel mit einfließen.

3.2 Modellerweiterungen

Möchte man das obige Modell implementieren, so stellt sich heraus, dass ein paar Erweiterungen notwendig sind, damit die Modellierung praktisch umgesetzt werden kann¹. Im Folgenden werden diese zusätzlichen Nebenbedingungen Schritt für Schritt eingeführt und deren Wirkungsweise diskutiert.

Zunächst sind einige Initialisierungsbedingungen notwendig, da in manchen Nebenbedingungen mit Indexverschiebungen wie $t - 1$ gearbeitet wird. Damit die Nebenbedingungen nicht künstlich individuell angepasst werden müssen, wird dem Modell ein Ausgangszeitpunkt $t = 0$ hinzugefügt. Dies ist keine wirkliche Einschränkung und hat auch wirtschaftlich gesehen durchaus seine Berechtigung: Lagerbestände der Zwischen- und Endprodukte zum Zeitpunkt $t = 0$ kann man sich als Endlagerbestände vom Vortag vorstellen; Lagerbestände von Rohstoffen zum Zeitpunkt $t = 0$ sind Bestellmengen für den Planungshorizont. Da im praktischen Teil dieser Arbeit über den Planungshorizont von 24 Stunden optimiert wird, werden die Anfangslagerbestände der Zwischen- und Endprodukte mit null initialisiert. Ansonsten könnte sich das Modell diese Lagerbestände frei wählen, was nicht der Idee einer mehrstufigen Produktion entspricht und damit nicht besonders realistisch erscheint. Die Anfangslagerbestände der Rohstoffe hingegen darf das Modell frei wählen, da ohne die nötige Reserve die Produktion nicht anlaufen kann. Da jedes einzelne Produkt – und damit auch Rohstoffe – Lagerkosten verursacht, sind in der Optimallösung die Anfangslagerbestände der Rohstoffe so gering wie möglich, aber so hoch wie für die Produktion erforderlich gewählt. Bezeichnet man die Menge der Rohstoffe mit Q , so kann man die zusätzlichen Bedingungen wie folgt definieren:

$$s_{p,0} = 0 \quad \forall p \in P \setminus Q$$

Diese Modellierung erlaubt es nicht, nach Beginn der Produktion weitere Rohstoffe zu kaufen. Diese Einschränkung erscheint allerdings auch sinnvoll, schließlich kann ein Unternehmen üblicherweise die Rohstoffe nicht stündlich, sondern lediglich für einen gewissen Planungshorizont im Voraus bestellen. Es sei noch darauf hingewiesen, dass diese Modellierung die Anschaffungskosten für die Rohstoffe vernachlässigt. Man kann sie allerdings entweder unmittelbar vom erzielbaren Erlös abziehen oder den Produktionskosten für die Prozesse aufschlagen.

Aufgrund der Nebenbedingung (3.3) müssen die Lieferrückstände zum Zeitpunkt $t = 0$ gesetzt werden. Auch hier erscheint es sinnvoll, diese null zu wählen, da die erste Nachfrage zum Zeitpunkt $t = 1$ erfolgen kann:

$$b_{p,0} = 0 \quad \forall p \in P$$

Analog müssen wegen (3.5) die Binärvariablen $y_{i,t}$ zum Zeitpunkt $t = 0$ gesetzt werden. Dazu gibt es zwei Möglichkeiten: Setzt man $y_{i,0} = 1$ für ein $i \in I_i$, so findet zum ersten Zeitpunkt aufgrund von (3.5) auf jeden Fall ein Setupwechsel statt, sofern man nicht zufällig den richtigen Prozess i ausgewählt hat. Dies könnte man als Anschaltkosten der Maschinen interpretieren, bzw. als den Endzustand des Vortages. Die zweite Möglichkeit,

$$y_{i,0} = 0 \quad \forall i \in I$$

hat die Bedeutung, dass die Maschinen in der ersten Periode schon betriebsbereit (und auf den richtigen Prozess eingestellt) sind und kein Setupwechsel mehr nötig ist. Da sich beide Versionen lediglich um

¹ Entsprechende Beispiele sind auf der CD im Anhang im Ordner „Kapitel3“ zu finden. Für das Ausgangsmodell siehe dort unter Beispiel1.

einen konstanten Term im Zielfunktionswert unterscheiden, und ein Setupwechsel in der ersten Periode eine gewisse Zeit $T_{i,j}$ dauert, wird im Folgenden die zweite Version verwendet. So kann zum Zeitpunkt $t = 1$ direkt mit der Produktion begonnen werden.

Da auch einige Indexverschiebungen über mehr als eine Periode erfolgen, muss hier sichergestellt werden, dass man dabei nicht über den Planungshorizont hinausläuft. In Gleichung (3.2) betrifft dies die Produktionsmenge $q_{i,t+\Delta_{p,j}}$ und in Ungleichung (3.6) die Binärvariable $x_{j,i,t-R_{j,i}}$. Möchte man das Hinausschreiten über den Endzeitpunkt $t = T$ verhindern, so lauten die neuen Nebenbedingungen:

$$s_{p,t-1} + \sum_{i:P_i=p} q_{i,t} - D_{p,t} - \sum_{\substack{q \in P \\ j \in I: P_j=q \wedge t+\Delta_{p,j} \leq T}} A_{p,q} q_{j,t+\Delta_{p,j}} + b_{p,t} + o_{p,t} - b_{p,t-1} = s_{p,t} \quad \forall p \in P, t = 1 \dots T$$

$$C_{i,t} \left(y_{i,t} - \sum_{\substack{j \in I_i \\ 0 < t-R_{j,i} < t' \leq t}} x_{j,i,t'} - \sum_{\substack{j \in I_i \\ t-R_{j,i} \geq 1}} R_{j,i}^f x_{j,i,t-R_{j,i}} \right) \geq q_{i,t} \quad \forall i \in I, t = 1 \dots T$$

Mit diesen notwendigen Änderungen erhält man die erste lauffähige Version des Modells². Allerdings ist leicht ersichtlich, dass das Modell noch nicht das gewünschte Ergebnis liefert. Zum einen werden einige Produkte früher produziert, als dies die Prozesskette zulässt, zum anderen tritt zum Endzeitpunkt $t = T$ u.U. noch ein Lieferrückstand auf. Das macht es für das Modell sehr lukrativ, nicht rechtzeitig fertig produzierte Produkte als Verzugsmenge in $b_{p,t}$ zu speichern und nicht als Fehlmenge $s_{p,t}$, da die Ausfallkosten üblicherweise deutlich höher als die Kosten für Lieferverzug sind. Da das Modell zum Zeitpunkt $t = T$ endet, erscheint es sinnvoll, bis dahin nicht befriedigte Nachfrage als Lieferausfall, und nicht als Lieferverzug anzusehen. Die entsprechende Nebenbedingung lautet damit

$$b_{p,T} = 0 \quad \forall p \in P, i \in I$$

und zwingt das Modell dazu, frühzeitig abzuwägen, ob die Nachfrage nachträglich befriedigt wird, oder ob Strafkosten für den Lieferausfall in Kauf genommen werden. Um das Problem mit den zu früh gestarteten Prozessen zu lösen, ist hingegen eine komplett neue Nebenbedingung erforderlich. Das Problem hat seinen Ursprung in der Nebenbedingung (3.2), da die Sichtweise dort in die Zukunft blickt: „Was ich zum Zeitpunkt $t + \Delta_{p,j}$ produzieren möchte, muss ich zum Zeitpunkt t aus dem Lager nehmen“. Dies vernachlässigt aber alle Zeitpunkte $t : 1 \leq t \leq 1 + \Delta_{p,j}$. Um auch hier die Bedingung einzuhalten, dass ein Produkt nur dann produziert werden kann, falls alle Vorprodukte ausreichend auf Lager liegen, ist folgende Nebenbedingung nötig:

$$A_{p,q} q_{i,t} \leq s_{p,t-\Delta_{p,i}} \quad \forall p, q \in P : A_{p,q} \neq 0, i \in I : P_i = q, t : \Delta_{p,i} \leq t \leq 2 \cdot \Delta_{p,i}$$

Sie drückt aus, dass die Produktionsmenge zum Zeitpunkt t geringer sein muss, als der Lagerbestand der Vorprodukte zu dem Zeitpunkt, zu dem die Vorprodukte aus dem Lager genommen werden, um sie zum nächsten Prozess zu bringen.

Nimmt man diese beiden zusätzlichen Bedingungen mit ins Modell auf und fordert, dass die Losgrößen ganzzahlig sind

$$q_{i,t} \in \mathbb{N} \quad \forall i \in I, t = 1 \dots T$$

² siehe CD unter „Beispiel2“

so liefert dies das gewünschte Ergebnis³. Im Folgenden werden noch zwei zusätzliche Faktoren betrachtet, die das Modell im Rahmen der Verwendbarkeit realistischer machen.

In den bisherigen Ausführungen wurden die Produktionskosten minimiert. Alternativ kann man natürlich auch den Gewinn betrachten und diesen maximieren. Da das Modell deterministisch ist, sind beide Sichtweisen äquivalent. Man muss lediglich in der Zielfunktion Erlöse für die abverkauften Produkte einführen und davon die Produktionskosten des Ursprungsmodells abziehen. Dabei sollte man beachten, dass die Strafkosten für Fehlmengen höher sind als die Erlöse, da ansonsten künstlich selbst dann Gewinn erwirtschaftet wird, wenn man jeden Lieferauftrag sofort als Fehlmenge meldet. Definiert man für den Erlös von Produkt p einen Parameter E_p , so ergibt sich als neue Zielfunktion:

$$\max \sum_{t=1}^T \left(\sum_{p \in P} E_p D_{p,t} - \left((S_p s_{p,t} + B_p b_{p,t} + O_p o_{p,t}) + \sum_{\substack{i \in I \\ j \in I_i}} X_{i,j} x_{i,j,t} + \sum_{i \in I} Q_i q_{i,t} \right) \right)$$

Abhängig von der Nachfrage nach Produkt p liefert das Grundmodell die minimalen Kosten. Setzt man nun Preise E_p für eine Einheit von Produkt p fest, so kann man an der Lösung ablesen, ob und wie viel Gewinn der Betrieb über den Planungshorizont erwirtschaftet. Auch die Break-Even-Analyse lässt sich mit diesem Modell berechnen, also die Preisuntergrenze, ab der die Produktion wirtschaftlich Sinn macht. Dazu bestimmt man über die Zielfunktion die minimalen Kosten, fügt statt einem fest vorgegebenen Preis E_p eine Variable e_p ein und fordert in einer Nebenbedingung, dass die Kosten gleich dem Erlös sein sollen. Eine Beispielrechnung zu diesem Modell findet man auf der beigefügten CD⁴.

Eine weitere interessante Erweiterung betrifft die Nachfrage $D_{p,t}$. Das Modell geht davon aus, dass man zum Anfangszeitpunkt bereits weiß, wie groß die Nachfrage zu den einzelnen Zeitpunkten sein wird. Die Lösung dieses Problems ist im Voraus bekannt und kann deterministisch bestimmt werden; der erzielbare Gewinn steht bereits zum Anfangszeitpunkt sicher fest. Interessanter wird das Modell, wenn man bei der Nachfrage stochastische Einflüsse betrachtet. Kennt man den genauen Bedarf nicht, sondern muss sich auf Erfahrungswerte oder Schätzungen verlassen, so weiß man zu Beginn nicht, wie groß die Nachfrage tatsächlich sein wird. Das Modell wird nun um genau diese Unsicherheit erweitert.

Es gibt nun eine Menge von Szenarien S , die jeweils mit einer gewissen Wahrscheinlichkeit ω^s eintreten können. Weiterhin wird angenommen, dass es einen Zeitpunkt \tilde{t} gibt, ab dem bekannt ist, welches der Szenarien eintritt. Bis zu diesem Zeitpunkt muss vorausschauend und kompromissbereit für alle möglichen Szenarien produziert werden. Wählt man $\tilde{t} = 0$, so berechnet das Modell für alle Szenarien unabhängig den optimalen Produktionsplan und als Zielfunktionswert erhält man den Erwartungswert des Gewinns. Wählt man $\tilde{t} = T$, also den Endzeitpunkt, so rechnet man zweimal dasselbe Modell und bekommt daher als Zielfunktionswert die Lösung des ursprünglichen deterministischen Modells.

Es gibt für jedes Szenario einen Satz Variablen, der beschreibt, wie die Variablen in den einzelnen Szenarien belegt werden. Im Modell wird dies durch ein hochgestelltes s , wie z.B. in $q_{i,t}^s$ verwirklicht. Ebenso legt man für jedes Szenario einen Parametersatz $D_{p,t}^s$ an, der die mögliche Nachfrage der einzelnen Szenarien darstellt. Optimiert wird nun über den mit Wahrscheinlichkeiten gewichteten Erwartungswert der einzelnen Szenarien. Schließlich braucht man noch ein paar weitere Nebenbedingungen, die sicherstellen, dass

³ siehe CD unter „Beispiel3“

⁴ unter „Beispiel4“

die einzelnen Szenarien nicht unabhängig voneinander gerechnet werden, sondern bis zum Zeitpunkt \tilde{t} die Variablen der einzelnen Szenarien miteinander verknüpft sind. Diese *Nichtantizipativitätsbedingungen* gibt es für jedes Variablenpaar und sind hier beispielhaft für die Produktionsmenge dargestellt:

$$q_{i,t}^{s_1} = q_{i,t}^{s_2} \quad \forall i \in I, t = 1 \dots \tilde{t}, s_1, s_2 \in S : s_1 = 1, s_2 \neq s_1$$

Mit diesen Nebenbedingungen werden die Produktionsmengen der einzelnen Szenarien s bis zum Zeitpunkt \tilde{t} gekoppelt. Die Einschränkung an s_1 und s_2 verhindert redundante Nebenbedingungen.

Nimmt man all diese Änderungen zum Ursprungsmodell hinzu, so erhält man das im folgenden Abschnitt noch einmal im Ganzen dargestellte Modell.

3.3 Modellübersicht

In Tabelle 3.3 sind alle Parameter und Variablen aufgeführt, die neu zum Modell hinzugekommen oder um stochastische Einflüsse ergänzt worden sind. Mit diesen Variablen und allen Änderungen aus dem vorangegangenen Abschnitt ergibt sich nun abschließend folgendes Modell. Es wird im nächsten Kapitel mithilfe von mathematischer Software implementiert und auf seine Praxistauglichkeit hin überprüft.

Tabelle 3.3: Alle neu eingeführten oder veränderten Variablen und Parameter

Symbol	Definition
Q	$\subseteq P$; Menge der Rohstoffe
S	Menge der möglichen Szenarien
\tilde{t}	Zeitpunkt, ab dem der Bedarf bekannt ist ($1 \leq \tilde{t} \leq T$)
ω^s	Eintrittswahrscheinlichkeit von Szenario s ($\sum_{s \in S} \omega^s = 1$)
E_p	Erlös (Exchange) für Produkt p
$D_{p,t}^s$	Die Nachfrage hängt nun auch von den Szenarien ab
$q_{i,t}^s$	Produktionsmenge von Produkt P_i durch Prozess i in Periode t in Szenario s
$s_{p,t}^s$	Lagerbestand von Produkt p am Ende von Periode t in Szenario s
$b_{p,t}^s$	Lieferrückstand von Produkt p am Ende von Periode t in Szenario s
$o_{p,t}^s$	Lieferausfall von Produkt p in Periode t in Szenario s
$x_{i,j,t}^s$	Binärvariable, die anzeigt, ob ein Setupwechsel zwischen Prozess i und j am Anfang von Periode t in Szenario s auftritt ($x_{i,j,t}^s = 1$), oder nicht ($x_{i,j,t}^s = 0$)
$y_{i,t}^s$	Binärvariable, die anzeigt, ob Maschine M_i bereit ist für Prozess i am Ende von Periode t in Szenario s ($y_{i,t}^s = 1$) oder nicht ($y_{i,t}^s = 0$)

$$\max \sum_{s \in S} \omega^s \cdot \left(\sum_{t=1}^T \left(\sum_{p \in P} E_p D_{p,t}^s - \left((S_p s_{p,t}^s + B_p b_{p,t}^s + O_p o_{p,t}^s) + \sum_{\substack{i \in I \\ j \in I_i}} X_{i,j} x_{i,j,t}^s + \sum_{i \in I} Q_i q_{i,t}^s \right) \right) \right)$$

$$\text{s. t.} \quad s_{p,t-1}^s + \sum_{i:P_i=p} q_{i,t}^s - D_{p,t}^s - \sum_{\substack{q \in P \\ j \in I:P_j=q \wedge t+\Delta_{p,j} \leq T}} A_{p,q} q_{j,t+\Delta_{p,j}}^s \quad \forall p \in P, s \in S, \\ t = 1 \dots T \\ + b_{p,t}^s + o_{p,t}^s - b_{p,t-1}^s = s_{p,t}^s$$

$$A_{p,q} q_{i,t}^s \leq s_{p,t-\Delta_{p,i}}^s \quad \forall p, q \in P : A_{p,q} \neq 0, \\ i \in I : P_i = q, s \in S, \\ t : \Delta_{p,i} \leq t \leq 2 \cdot \Delta_{p,i}$$

$$b_{p,t}^s + o_{p,t}^s - b_{p,t-1}^s \leq D_{p,t}^s \quad \forall p \in P, s \in S, \\ t = 1 \dots T$$

$$\sum_{i:M_i=m} y_{i,t}^s = 1 \quad \forall m \in M, s \in S, \\ t = 1 \dots T$$

$$y_{i,t-1}^s + y_{j,t}^s - 1 \leq x_{i,j,t}^s \quad \forall i \in I, j \in I_j, s \in S, \\ t = 1 \dots T$$

$$C_{i,t} \left(y_{i,t}^s - \sum_{\substack{j \in I_i \\ 0 < t - R_{j,i} < t' \leq t}} x_{j,i,t'}^s - \sum_{\substack{j \in I_i \\ t - R_{j,i} \geq 1}} R_{j,i}^f x_{j,i,t-R_{j,i}}^s \right) \geq q_{i,t}^s \quad \forall i \in I, s \in S, \\ t = 1 \dots T$$

$$q_{i,t}^{s_1} = q_{i,t}^{s_2}, \quad y_{i,t}^{s_1} = y_{i,t}^{s_2} \quad \forall i \in I, t = 1 \dots \tilde{t}, \\ s_1, s_2 \in S : s_1 = 1, s_2 \neq s_1$$

$$s_{p,t}^{s_1} = s_{p,t}^{s_2}, \quad b_{p,t}^{s_1} = b_{p,t}^{s_2}, \quad o_{p,t}^{s_1} = o_{p,t}^{s_2} \quad \forall p \in P, t = 1 \dots \tilde{t}, \\ s_1, s_2 \in S : s_1 = 1, s_2 \neq s_1$$

$$x_{i,j,t}^{s_1} = x_{i,j,t}^{s_2} \quad \forall i \in I, j \in I_i, \\ t = 1 \dots \tilde{t}, \\ s_1, s_2 \in S : s_1 = 1, s_2 \neq s_1$$

$$q_{i,t}^s \geq 0 \quad \forall i \in I, s \in S, t = 1 \dots T$$

$$s_{p,t}^s, b_{p,t}^s, o_{p,t}^s \geq 0 \quad \forall p \in P, s \in S, t = 1 \dots T$$

$$x_{i,j,t}^s \in \{0, 1\} \quad \forall i \in I, j \in I_i, s \in S, \\ t = 1 \dots T$$

$$y_{i,t}^s \in \{0, 1\} \quad \forall i \in I, s \in S, t = 1 \dots T$$

$$q_{i,t}^s \in \mathbb{N} \quad \forall i \in I, s \in S, t = 1 \dots T$$

$$b_{p,0}^s, b_{p,T}^s = 0 \quad \forall p \in P, s \in S$$

$$y_{i,0}^s = 0 \quad \forall i \in I, s \in S$$

$$s_{p,0}^s = 0 \quad \forall p \in P \setminus Q, s \in S$$

Um diese Modellierung ggf. mit anderen Modellen vergleichen zu können, ist die Gesamtvariablenanzahl interessant. Sie ist abschließend in folgender Tabelle zusammengefasst. Es sei darauf hingewiesen, dass T die Periodenanzahl und keine Menge ist. Insgesamt ergeben sich $|S| \cdot T \cdot (|I|^2 + 2|I| + 3|P|)$ viele Variablen. Für das im nächsten Kapitel gerechnete stochastische Szenario werden nach dieser Formel 11520 Variablen benötigt.

Tabelle 3.4: Variablenanzahl der Modellierung.

Variable	Bezeichnung	Anzahl
$q_{i,t}^s$	Produktionsmenge	$ I \cdot T \cdot S $
$s_{p,t}^s$	Lagerbestand	$ P \cdot T \cdot S $
$b_{p,t}^s$	Lieferverzug	$ P \cdot T \cdot S $
$o_{p,t}^s$	Lieferausfall	$ P \cdot T \cdot S $
$x_{i,j,t}^s$	Setupwechsel	$ I ^2 \cdot T \cdot S $
$y_{i,t}^s$	Bereitschaft	$ I \cdot T \cdot S $

4 Praktische Umsetzung des Modells anhand eines Beispiels

Nachdem das Modell eingeführt und seine Funktionsweise erläutert wurde, ist es das Ziel dieses Kapitels, die Praxistauglichkeit des Modells anhand einer Beispielinstantz zu überprüfen. Zunächst werden die Instanz und die verwendete Software vorgestellt. Im Anschluss daran werden verschiedene Rechnungen durchgeführt und analysiert. Um das Nachvollziehen der Untersuchungen zu erleichtern, findet man auf der CD im Anhang im Ordner „Kapitel4“ die zu diesen Rechnungen gehörenden Datensätze.

4.1 Die Beispielinstantz

Wie bereits zu Beginn erwähnt, wird zur Veranschaulichung des Modells der Produktionsplanung im Folgenden von einem fiktiven Unternehmen ausgegangen, das Rikschas in verschiedenen Ausfertigungen herstellt. Dieser Abschnitt beschreibt den Produktionsablauf von den Rohstoffen bis zu den fertigen Rikschas und stellt die Verbindung zur mathematischen Modellierung aus dem vorherigen Kapitel her.

Der Bau einer Rikscha besteht aus zwei gekoppelten Teilbereichen: Zunächst werden die Stäbe produziert und bearbeitet, bevor sie in der Fertigung zu Rikschas zusammengebaut werden können. Als Rohstoffe sind lange Stäbe (im Folgenden mit Produkt0 bezeichnet), Reifen (Produkt9) und Sitze (Produkt10) zu nennen, die auf dem Markt beschafft werden. Da für die Fertigung verschiedenartige Stangen benötigt werden, müssen sie vor dem Zusammenbau in verschiedene Formen gebracht werden. Eine erste Maschine kann aus den Ursprungsstäben entweder kurze oder lange Stäbe schneiden. Im nächsten Bearbeitungsschritt werden die Stäbe auf einer weiteren Maschine an den Enden zusammengedrückt. Im letzten Bearbeitungsschritt werden von der dritten Maschine Löcher in die Stäbe gebohrt. Dabei kann die Maschine entweder eins, zwei oder drei Löcher bohren. In den verwendeten Datensätzen werden die Maschinen mit Maschine1, Maschine2 und Maschine3 bezeichnet. Da jede dieser Maschinen mehrere Aufgaben erledigen kann, werden die entsprechenden Prozesse (im späteren Modell Tasks genannt) gemäß der folgenden Tabelle definiert.

Tabelle 4.1: Zusammenhang zwischen Maschinen und Prozessen für die Stäbe in der Beispielinstantz

Maschine	Task	Erzeugter Stab	Produktbezeichnung
Maschine1	Task1	kurze Stäbe	Produkt1
Maschine1	Task2	lange Stäbe	Produkt2
Maschine2	Task3	kurze Stäbe platt	Produkt3
Maschine2	Task4	lange Stäbe platt	Produkt4
Maschine3	Task5	kurze Stäbe 3 Löcher	Produkt5
Maschine3	Task6	lange Stäbe 1 Loch	Produkt6
Maschine3	Task7	kurze Stäbe 2 Löcher	Produkt7
Maschine3	Task8	lange Stäbe 2 Löcher	Produkt8

Die auf den ersten Blick etwas verwirrend erscheinende Nummerierung der Tasks hat ihren Ursprung in der Art der Stäbe: Die Prozesse, die die kurzen Stäbe bearbeiten, haben eine ungerade Nummer und entsprechend diejenigen, die lange Stäbe bearbeiten, gerade Nummern.

Sind genug Stäbe auf Lager, so kann anschließend der Fertigungsprozess der Rikschas beginnen. Insgesamt gibt es zwei Typen von Rikschas, wobei eine der beiden Ausfertigungen (Produkt12) optional mit einem Anhänger ausgeliefert wird (Produkt15) oder der Anhänger separat bestellt werden kann. Der Anhänger besteht im Wesentlichen aus dem anderen Rikscha-Typ, sodass vor der Fertigstellung ein Zwischenprodukt (Produkt11) erstellt wird, und dann je nach Bedarf eine Rikscha (Produkt13) oder ein Anhänger (Produkt14) hergestellt werden kann. Hierfür sind zwei weitere Maschinen nötig, die wie folgt mit den Prozessen verbunden sind:

Tabelle 4.2: Zusammenhang zwischen Maschinen und Prozessen für den Zusammenbau der Rikschas in der Beispielinstantz

Maschine	Task	Erzeugte Rikscha	Produktbezeichnung
Maschine4	Task9	Zwischenprodukt	Produkt11
Maschine4	Task10	Rikscha Typ2	Produkt12
Maschine5	Task11	Rikscha Typ1	Produkt13
Maschine5	Task12	Anhänger	Produkt14
Maschine5	Task13	Rikscha Typ2 mit Anhänger	Produkt15

Eine graphische Veranschaulichung, wie aus den Stäben die Rikschas zusammengebaut werden, findet man in Abbildung 4.1. Der gesamte Produktionsplan einschließlich der Gozinto-Faktoren ist noch einmal in Abbildung 4.2 dargestellt. Es sei noch erwähnt, dass im Modell von ein paar Details abstrahiert wird. So werden sicherlich Schrauben für den Zusammenbau von Rikschas benötigt, und ob die Fertigung von Maschinen oder Menschen vorgenommen wird, bleibt ebenfalls offen. Allerdings kann man dafür ggf. anfallende Zusatzkosten direkt den entsprechenden Produktionskosten für den jeweiligen Prozess aufschlagen.

Damit ist der grundsätzliche Produktionsablauf des fiktiven Rikschaunternehmens vollständig vorgestellt. Diejenigen Parameter, die später für die Modellanalyse benötigt und verändert werden, werden an der jeweiligen Stelle genauer beleuchtet. Es folgen jedoch noch ein paar grundsätzliche Hinweise zu der sinnvollen Größenordnung einiger Parametern, damit das Modell nachvollziehbare Ergebnisse liefert:

- Die Kosten für die Ausfallmengen sollten mindestens doppelt so hoch sein als die erzielbaren Gewinne. Sind sie sogar niedriger, so würde auch ein Produktionsstillstand Gewinn einbringen. Sind sie nicht wesentlich höher als die Gewinne, so wird sich das Modell häufiger dazu entschließen, bei Lieferverzug sofort Ausfallkosten zu zahlen, anstatt die Nachproduktion anzustreben und Verzugskosten in Kauf zu nehmen.
- Die Verzugskosten sollten etwas höher sein als die Lagerkosten für fertige Produkte, da das Modell ansonsten auf Vorrat produzierte Erzeugnisse als Lieferverzug zwischenspeichert.
- Da zum Anfangszeitpunkt keine fertigen Rikschas auf Lager sind, und mit der Produktion der Stangen ebenfalls erst zum Anfangszeitpunkt begonnen werden kann, sollte man dem Modell etwas Vorlaufzeit geben, ehe die erste Nachfrage eintritt. Dies wird sicher auch in der Praxis so gehandhabt, da ein Unternehmen keine Aufträge annehmen wird, wenn es im Voraus weiß, dass es nicht fristgerecht – oder zumindest zeitnah – liefern kann.
- Die Summe aus Produktions- und Lagerkosten für die einzelnen Bestandteile einer Rikscha sollten weitaus niedriger sein als die Strafkosten für Fehlmengen. Da sich die Produktionszeit einer Rik-

scha über mehrere Perioden erstreckt und die Lagerkosten pro Periode anfallen, ist es ansonsten profitabler die Produktion einzustellen.

Alle weiteren Ergebnisse werden in Abschnitt 4.4 diskutiert und mit konkreten Daten belegt. Zuvor folgt ein Abschnitt über die Software, die zur Implementierung verwendet wurde und eine ausführliche Beispielrechnung.

Bill of Materials

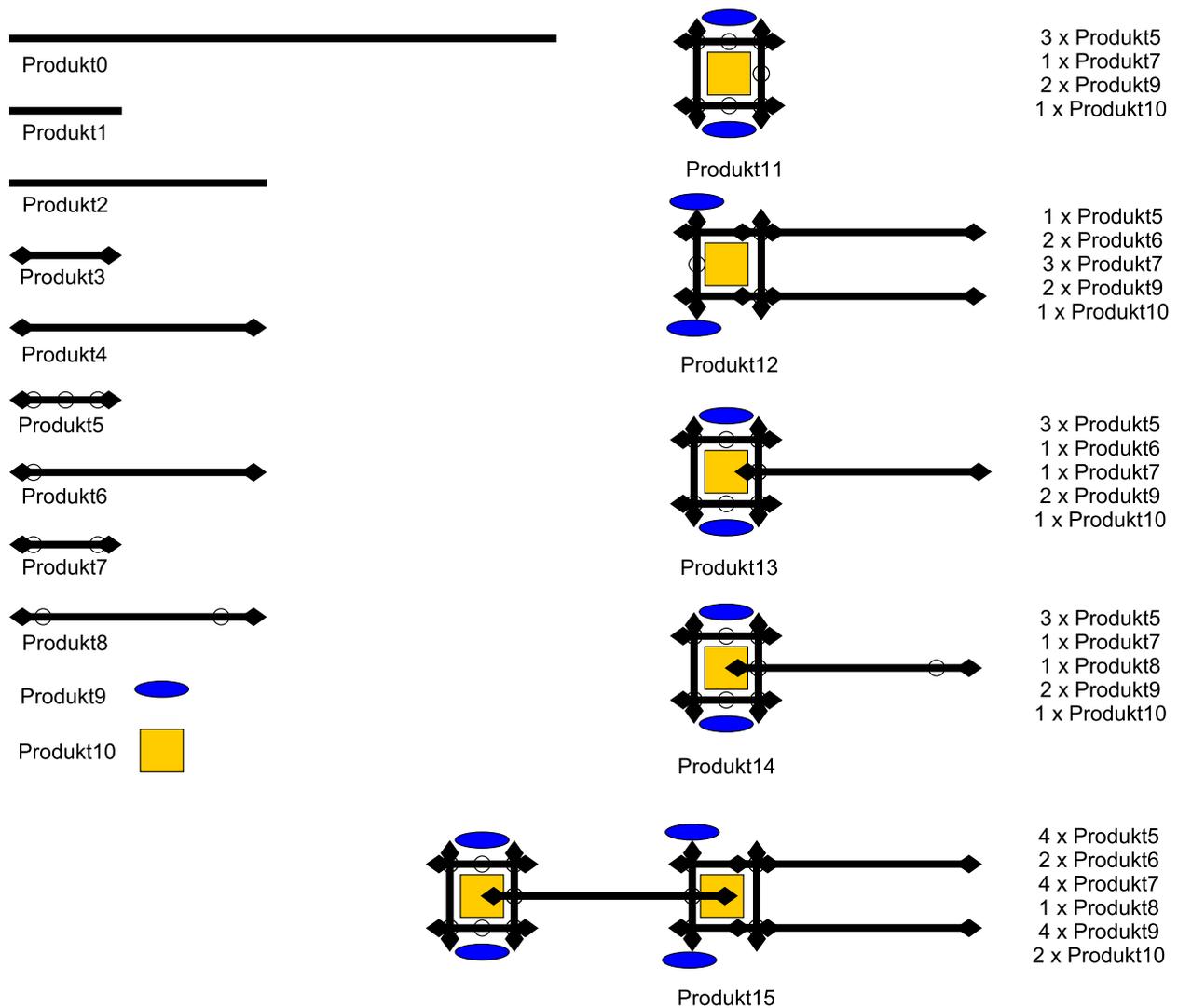


Abbildung 4.1: Produktionsstufen der Rikscha. Dargestellt sind die einzelnen Bearbeitungsstufen der Stäbe und schließlich der Zusammenbau der Rikscha. Angegeben sind zusätzlich noch die Stücklisten (Bill of Materials, BOM), also die Menge der einzelnen Vorprodukte, die für eine fertige Rikscha benötigt werden.

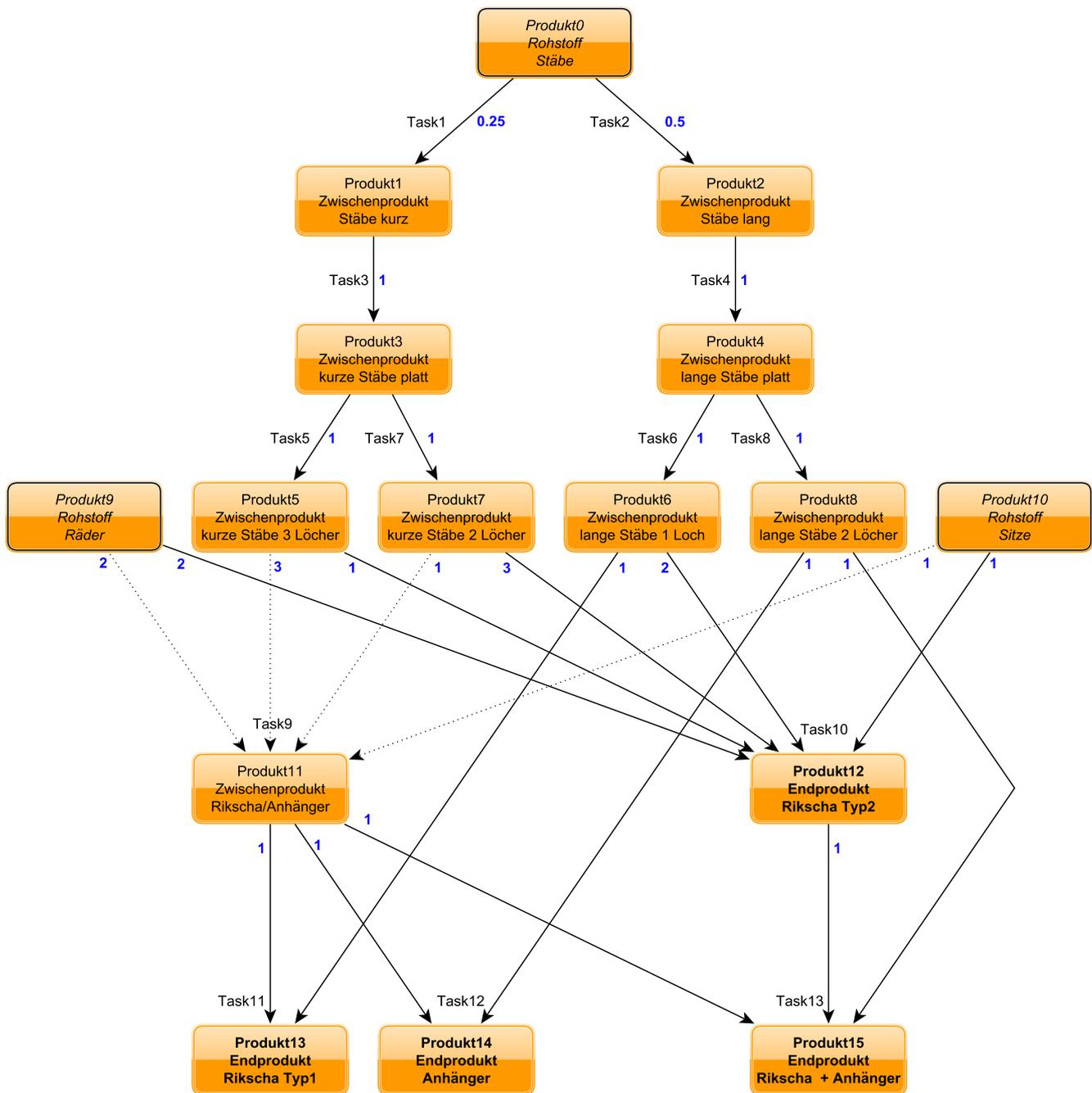


Abbildung 4.2: Gozinto-Graph des Produktionsprozesses. Kursiv dargestellt sind die Rohstoffe, die möglichen Endprodukte fett. Die blauen Kantenbewertungen stellen den Gozinto-Faktor dar, also die Anzahl Vorprodukte, die benötigt wird, um ein Nachfolgeprodukt zu produzieren. Des Weiteren sind die Tasks angegeben, die das entsprechende Produkt produzieren.

4.2 Mathematische Software

Um Instanzen des mathematischen Modells aus Kapitel drei mit Hilfe von mathematischer Software zu lösen, wurden die Programme GLPK und CPLEX verwendet. Das GNU Linear Programming Kit (GLPK) ist ein Softwarepaket, mit dessen Hilfe sich mathematische Modelle formulieren lassen. GLPK ist quelloffen (Open Source), kostenlos und kann aus dem Internet heruntergeladen werden [GLP11]. Die Modellierungssprache von GLPK heißt GMPL¹ und ist an AMPL² angelehnt. Daher ist die Verwendung für Mathematiker intuitiv.

Für das mathematische Modell wird eine Datei mit der Endung `.mod` angelegt. Darin enthalten sind alle Variablen und Parameterdefinitionen, sowie die Zielfunktion und alle benötigten Nebenbedingungen. Möchte man eine Instanz des Problems rechnen, werden die konkreten Daten entweder direkt in der `.mod` Datei eingefügt oder in einer eigenen Datei mit der Endung `.dat` abgespeichert. GLPK besitzt einen Kommandozeilen-Solver namens `glpsol`, der mit der Simplexmethode, Inneren-Punkte-Verfahren und einem Branch&Cut Algorithmus arbeitet.

Werden die Instanzen zu groß, so eignet sich `glpsol` zum Lösen nur noch bedingt, da die Laufzeit nicht mehr vertretbar ist. Daher empfiehlt es sich, auf einen kommerziellen Solver, wie z.B. den IBM ILOG CPLEX Optimizer umzusteigen [CPL11]. Dieser Solver wird für die weiteren Rechnungen verwendet. Als Eingabe erwartet der CPLEX Solver eine `.lp` Datei, welche sich einfach aus GLPK heraus erzeugen lässt. Man kann also problemlos von `glpsol` auf CPLEX umsteigen.

Die Lösungen und Laufzeiten, die im Folgenden vorgestellt werden, wurden alle (sofern nicht anders angegeben) auf dem TU-Fachgebietsrechner fb04554 der AG Optimierung erzielt. Im Anhang findet man die Datei „ModellDeterministisch.mod“ zu dem Ausgangsmodell, dessen Ergebnisse nun vorgestellt werden. Alle weiteren Dateien, die diskutiert werden, findet man in den jeweiligen Ordnern auf der beigefügten CD. Im Einzelnen sind dies beispielsweise für Instanz1:

- Instanz1.dat: Hier werden alle Daten und Parameter für die Instanz definiert
- Instanz1.lp: Die aus GLPK heraus erzeugte `.lp` Datei, welche mit CPLEX geladen werden kann
- Instanz1.sol: Die Optimallösung als CPLEX File gespeichert
- Instanz1sol.txt: Die Optimallösung als Textdatei
- ModellDeterministisch.mod: Das verwendete Modell

Um in GLPK die Übersicht zu behalten, wurden die Variablen und Parameter mit anderen Bezeichnungen versehen als in der mathematischen Modellierung. Zum besseren Verständnis folgt eine Tabelle, die diese einführt.

¹ GNU Mathematical Programming Language bzw. GNU MathProg

² A Mathematical Programming Language

Tabelle 4.3: Parameter- und Variablenbezeichnung der mathematischen Modellierung und der verwendeten Software im Vergleich.

Parameter	GLPK	Variable	GLPK
P	Produkte	$q_{i,t}^s$	prodmenge[i,t,s]
Q	Rohstoffe	$s_{p,t}^s$	lagerbestand[p,t,s]
M	Maschinen	$b_{p,t}^s$	backlog[p,t,s]
I	Tasks	$o_{p,t}^s$	shortfall[p,t,s]
T	Perioden	$x_{i,j,t}^s$	changesetup[i,j,t,s]
S	Szenarien	$y_{i,t}^s$	ready[i,t,s]
\tilde{t}	Info		
ω^s	Prob[s]		
E_p	Erloes[p]		
M_i	Benoetigte_Maschine[i]		
P_i	Produziertes_Produkt[i]		
I_i	Gleiche_Maschine[i,j]		
S_p	Lagerkosten[p]		
B_p	Backlogkosten[p]		
O_p	Shortfallkosten[p]		
$X_{i,j}$	Setupkosten[i,j]		
Q_i	Produktionskosten[i]		
$D_{p,t}^s$	Demand[p,t,s]		
$A_{p,q}$	Gozinto_Faktor[p,q]		
$\Delta_{p,i}$	Transportzeit[p,i]		
$C_{i,t}$	Kapazitaetsbeschraenkung[i,t]		
$T_{i,j}$	Changesetupzeit[i,j]		
$R_{i,j}$	Changesetupzeitinteger[i,j]		
$R_{i,j}^f$	Changesetupzeitfraction[i,j]		

4.3 Eine Beispielrechnung

Im Folgenden wird die Funktionsweise des Modells mit einem konkreten Datensatz vorgestellt, der für die weitere Analyse als Ausgangslage dienen soll.

Als Planungshorizont wird ein kompletter Tag betrachtet, wobei die Stunden die Perioden darstellen. Lager-, Lieferrückstands- und Fehlmengenkosten sind gemäß den Hinweisen in Abschnitt 4.1 gewählt. Es gibt keine Transportzeiten für die Produktion der Stangen (Task1 bis Task8), da hier von einer Fließbandfertigung ausgegangen wird. Für alle anderen Tasks wurde eine Transportzeit von einer Stunde gewählt, um das Produkt zur Fertigungsstelle zu bringen. Um zwischen den Tasks zu wechseln, fallen unterschiedlich hohe Setupkosten und -zeiten an, je nachdem wie aufwendig es ist, die Maschinen umzurüsten. Die Nachfrage findet im Ausgangsmodell ausschließlich zum Endzeitpunkt $t = T = 24$ statt, so dass der ganze Tag zur Produktion genutzt werden kann. Die Nachfrage ist aber so hoch gewählt, dass das Modell schon zu Beginn mit der Produktion beginnen muss. Die genauen Daten und Parameter findet man auf der CD im Ordner „Instanz1“.

Bevor das Ergebnis dieser Instanz diskutiert wird, wird zunächst die Funktionsweise des Modells untersucht. In Abbildung 4.3 kann man sehen, wie sich der Lagerbestand der Rohstoff-Stäbe im Laufe der Zeit verändert. Da die Stäbe ein Rohstoff sind, befindet sich zu Planungsbeginn bereits die für den Planungshorizont benötigte Menge im Lager. Im ersten Bearbeitungsschritt werden aus diesen Stäben entweder kurze oder lange Stäbe geschnitten. Der Gozinto-Faktor für die kurzen Stäbe beträgt $\frac{1}{4}$, d.h. aus einem Stab werden vier kleine Stäbe gesägt. Analog beträgt der Gozinto-Faktor für die langen Stangen $\frac{1}{2}$, es können also zwei große Stäbe aus den Rohstoff-Stäben gesägt werden. Beispielsweise werden in der ersten Periode 120 lange Stangen gesägt, also reduziert sich der Lagerbestand der Rohstoff-Stäbe um 60. In Abbildung 4.3 fällt auf, dass der Lagerbestand der Rohstoffe bei der Produktion von langen Stäben schneller abnimmt als bei den kurzen. Dies lässt sich mit der maximalen Produktionsmenge begründen, die für die langen Stäbe 120 und für die kurzen 80 beträgt und den unterschiedlichen Gozinto-Faktoren für kurze und lange Stäbe. Am Ende des Planungshorizontes ist der Lagerbestand der Stäbe leer. Der Grund dafür liegt an den Kosten, die das Einlagern von zusätzlichen Stäben verursachen würde. Da die Zielfunktion die Kosten minimiert, sind in der Optimallösung zu jedem Zeitpunkt genau so viele Stangen im Lager, wie für den weiteren Verlauf benötigt werden.

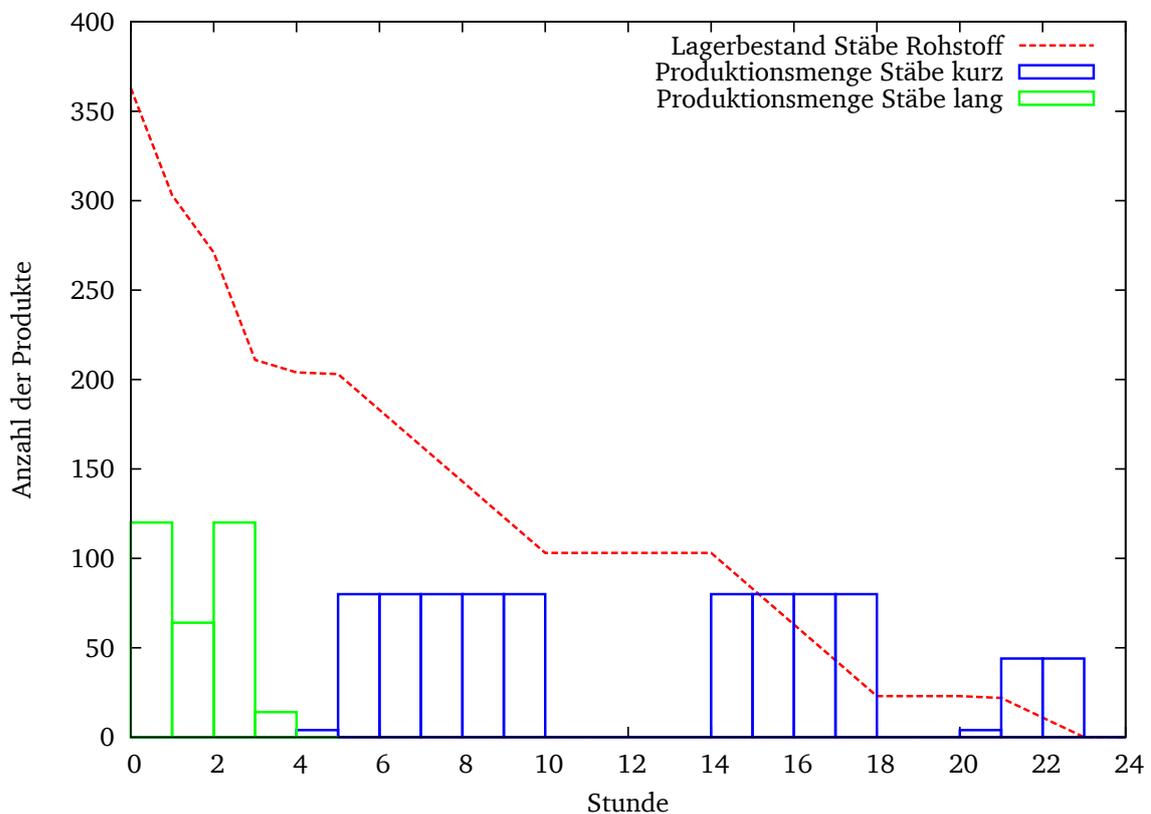


Abbildung 4.3: Lagerbestand der Stäbe und Produktionsmenge der Zwischenprodukte in Instanz 1

Etwas spannender zu betrachten ist der Lagerbestand von Produkt 6, den langen Stäben mit einem Loch. Diese liegen zu Planungsbeginn noch nicht im Lager und müssen erst von Task 6 produziert werden. Wie man in Abbildung 4.4 sehen kann, werden alle nötigen langen Stäbe gleich in den ersten Stunden produziert. Dies hat seinen Ursprung in den niedrigeren Lagerkosten der bearbeiteten Stäbe im Vergleich zu den Rohstoff-Stäben. Der Lagerbestand bleibt solange unverändert, bis eines der Nachfolgeprodukte, in diesem Fall Rikscha Typ 1 oder Rikscha Typ 2, produziert wird. Hierbei ist auffällig, dass sich der Lagerbestand schon eine Periode vor dem Produktionsbeginn anfängt zu leeren und dafür eine Periode

früher aufhört, was die Transportzeiten im Modell bedingen. Um die nötigen Produkte zum Zusammenbau der Rikscha zu transportieren, ist im Datenfile eine Zeiteinheit vorgesehen. Während dieser einen Periode ist das Produkt nicht mehr im Lager, kann aber auch noch nicht produziert werden. Dies ist ein Nachteil der Modellierung, denn diese Transportmenge erscheint in keiner Variablen. Auch in Abbildung 4.4 ist der Lagerbestand am Tagesende aus dem gleichen Grund wie oben auf null gesunken.

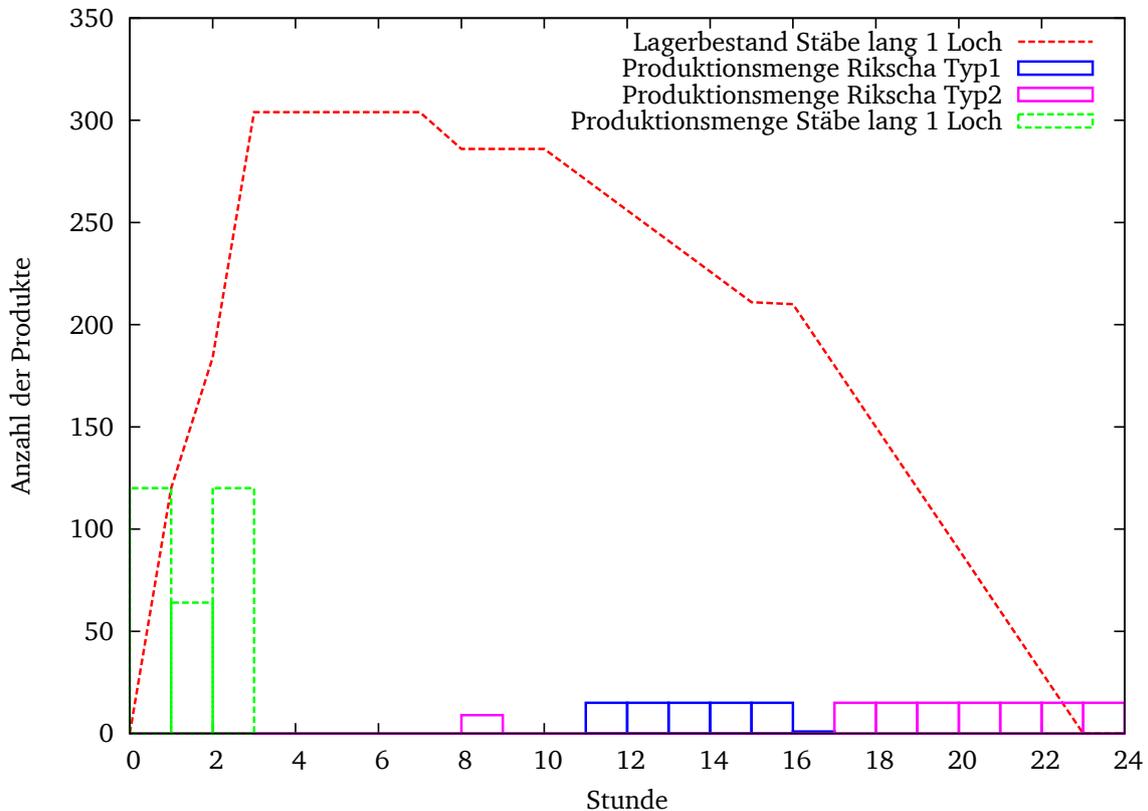


Abbildung 4.4: Lagerbestand der der langen Stäbe in Instanz1

In Abbildung 4.5 ist der Lagerbestand der Räder und Sitze und die Produktionsmengen des Zwischenproduktes und von Rikscha Typ2 im Zeitverlauf abgetragen. Da die Räder und Sitze, wie die unbearbeiteten Stäbe Rohstoffe darstellen, befinden sich bereits zu Beginn die notwendige Anzahl Räder und Sitze im Lager. Da jede Rikscha mit zwei Rädern und einem Sitz ausgestattet ist, ist der Lagerbestand der Räder immer doppelt so groß wie der Bestand der Sitze. Auch hier sind für die Transportzeiten eine Zeiteinheit gewählt, weshalb die Produktion immer eine Periode später stattfindet, als Sitze und Räder aus dem Lager geholt werden.

Anhand dieser ausgewählten Produkte erkennt man, dass das Modell mit allen nötigen Erweiterungen aus Kapitel drei ein sinnvolles Ergebnis liefert. Es bleibt abschließend das Ergebnis an sich zu bewerten. Die Nachfrage nach Rikscha Typ2 in Höhe von 100 Stück kann komplett gedeckt werden, während von den 80 nachgefragten Rikschas von Typ1 vier nicht produziert werden können und zum Zeitpunkt $t = 24$ ein Lieferausfall eintritt. Außerdem kann die Nachfrage nach losen Anhängern und nach Rikscha Typ2 mit Anhänger nicht gedeckt werden und es tritt auch dort ein Lieferausfall auf. Insgesamt ist die Zusammensetzung der Nachfrage also etwas höher gewählt, als die maximal mögliche Produktionsmenge. Dennoch wird in dieser Instanz ein Gewinn von 1230,80 € erzielt. Für diese Optimallösung hat CPLEX ca. 241 Sekunden benötigt, was für einen Planungshorizont von 24 Stunden durchaus akzeptabel ist.

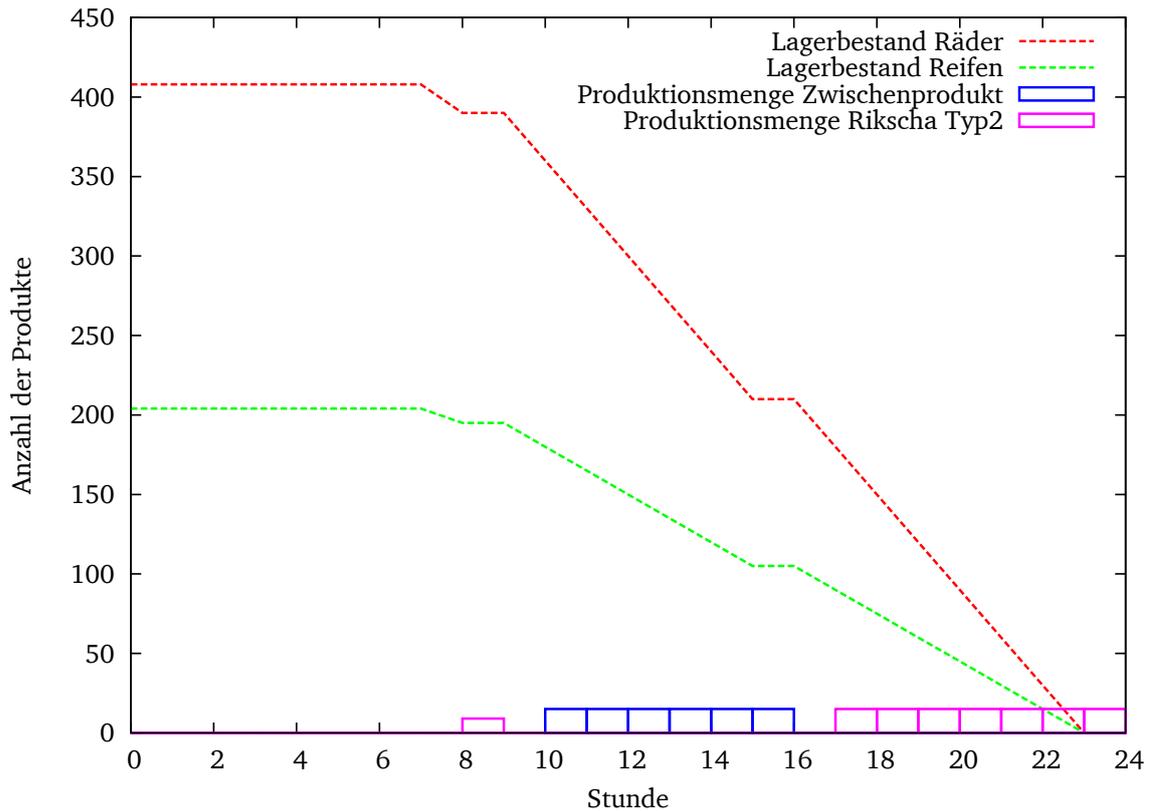


Abbildung 4.5: Lagerbestand und Verbrauch der Räder und Sitze in Instanz1

4.4 Verhaltensanalyse

Nun werden einige Parameter, insbesondere die Nachfrage, verändert und jeweils das erhaltene Ergebnis hinsichtlich Güte und Laufzeit analysiert. Alle angegebenen Instanzen und deren Lösungen sind auch auf der beigefügten CD enthalten und können im entsprechenden Unterordner nachvollzogen werden. Als Referenzwert dient immer Instanz1, welche im vorherigen Abschnitt ausführlich diskutiert wurde. Instanz2 bis Instanz7 rechnen ebenfalls mit dem deterministischen Modell, während Instanz8 bis Instanz10 das stochastische Modell aus Kapitel 3 verwenden.

In **Instanz2** wurde zusätzlich zur Nachfrage am Endzeitpunkt eine weitere Nachfrage zum Zeitpunkt $t = 12$ in Höhe von zehn Einheiten für jedes Produkt eingefügt. Da in Instanz1 schon am oberen Limit produziert wurde, kann man davon ausgehen, dass eine zusätzliche Nachfrage den Gewinn schmälert, weil die Instanz die Nachfrage nicht mehr decken kann. Es tritt in Instanz2 sogar ein Verlust in Höhe von $-870,00 \text{ €}$ auf. Die zusätzliche Nachfrage am Mittag bringt das Modell dazu, früher mit der Produktion der Endprodukte zu beginnen. Außerdem tritt nun auch ein Lieferverzug von Produkt13 zum Zeitpunkt $t = 12$ auf. Das Modell entscheidet sich also dazu, die Nachfrage eine Periode später noch zu decken. Die zusätzliche Nachfrage bedingt einen höheren Lieferausfall am Endzeitpunkt. Konnten in Instanz1 nur 30 Produkte nicht produziert werden, so sind es in Instanz2 66 Produkte. Dies ist der Hauptgrund für das negative Ergebnis. Die längere Laufzeit in Höhe von 986 Sekunden lässt darauf schließen, dass es schwieriger ist, den optimalen Produktionsplan zu bestimmen, wenn die Nachfrage an mehreren Zeitpunkten auftritt.

Im Vergleich zu Instanz1 wurde in **Instanz3** die identische Nachfrage gleichmäßig über den Nachmittag verteilt. Das Ergebnis von Instanz3 in Höhe von 4373,30 € erscheint im ersten Moment zu hoch zu sein, zumal zwei Produkte mehr als in der Ursprungsinstanz nicht produziert werden können. Das Ergebnis lässt sich aber recht plausibel durch die Lagerkosten erklären: Je früher die ersten Produkte verkauft werden, desto kürzer verharren die Rohstoffe und Zwischenprodukte in ihren Lagern. Da die Produkte im Ursprungsmodell bis zum Endzeitpunkt Lagerkosten verursachen, ist das Ergebnis dort schlechter. Auch die Laufzeit von Instanz3 liegt mit 835 Sekunden absolut im Rahmen. Ähnlich wie Instanz2 braucht diese Instanz länger im Vergleich zur Ursprungsinstanz, um den optimalen Produktionsplan zu finden.

Das Phänomen der Lagerkosten tritt auch in **Instanz4** auf. Dort wurde der Planungshorizont auf 48 Stunden erweitert und zum Zeitpunkt 24 und 48 fallen jeweils die Nachfrage von Instanz1 an. Lässt man die Lagerkosten außer acht, so könnte man meinen, dass das Ergebnis genau doppelt so hoch sein müsste wie in Instanz1, da die Instanz3 nur eine Aneinanderreihung von Instanz1 ist. Die Lagerkosten fallen allerdings so sehr ins Gewicht, dass ein Verlust in Höhe von -1852,00 € auftritt. Außerdem ist die Laufzeit von Instanz4 mit fast vier Tagen länger als der Planungshorizont und damit nicht mehr vertretbar. Dies hängt einerseits mit der größeren Variablenanzahl zusammen, die daraus resultiert, dass jede Variable für jeden Zeitschritt existiert, andererseits entstehen mit Erweiterung des Planungshorizontes immer mehr zulässige Lösungen für das Optimierungsproblem. Letzteres bedeutet anschaulich, dass automatisch mehr Möglichkeiten entstehen, in welcher Reihenfolge welche Produkte produziert werden können. Aufgrund der großen Komplexität dieser Instanz wurde das Ergebnis auf einem leistungsstarken Server der Arbeitsgruppe Optimierung³ erzielt. Die Laufzeit ist somit nicht direkt vergleichbar mit den anderen Instanzen.

In **Instanz5** wurden alle Transportzeiten auf null gesetzt. Es wird also davon ausgegangen, dass alle Produkte sofort für den nächsten Prozess bereit stehen. Dadurch sinkt die Produktionszeit einer Rikscha erheblich. Zusätzlich wurde eine komplett andere Nachfrage eingeführt: Die Nachfrage nach Rikschas von Typ1 steigt bis zum Mittag an und fällt dann wieder ab. Ab 11 Uhr bis 18 Uhr werden pro Stunde zehn Rikschas vom Typ2 nachgefragt und ab 18 Uhr bis 24 Uhr in der gleichen Höhe Rikschas mit Anhänger. Diese komplett andere Nachfrage macht es natürlich schwierig, das Ergebnis von Instanz5 mit Instanz1 zu vergleichen. Es zeigt allerdings, dass durch den Wegfall der Transportzeiten trotz relativ großer und abwechslungsreicher Nachfrage noch ein Gewinn von 650,30 € erzielt werden kann. Einerseits treten relativ viele Lieferausfälle und damit verbunden Strafzahlungen auf, andererseits verringern sich durch die frühe und recht hohe Nachfrage die Lagerkosten erheblich. Dieses Zusammenspiel ermöglicht das positive Ergebnis. Die Laufzeit liegt mit 6182 Sekunden, das sind etwas mehr als 100 Minuten, noch im akzeptablen Bereich.

Instanz6 und **Instanz7** werden hauptsächlich für Instanz9 benötigt. Dennoch werden die jeweiligen Ergebnisse an dieser Stelle kurz vorgestellt. In Instanz6 werden ausschließlich 200 Rikschas von Typ1 zum Endzeitpunkt nachgefragt. Die Lösung dieses Problems beansprucht 647 Sekunden, da die Produktion lediglich auf diese Nachfrage eingestellt werden muss. Der Zielfunktionswert liegt bei 5501,90 €. Ein gutes Ergebnis, das seinen Ursprung in den geringeren Rüstkosten und -zeiten hat. Es können lediglich drei Rikschas nicht produziert werden. Analog werden in Instanz7 nur 200 Anhänger zum Endzeitpunkt nachgefragt. Die Laufzeit dieser Instanz beträgt 51 Sekunden und der Zielfunktionswert 408,00 €. Die Differenz zu Instanz6 entsteht durch den niedrigeren Erlös, der für die Anhänger erzielt werden kann und durch einen Lieferausfall in Höhe von 20 Einheiten zum Endzeitpunkt.

³ numopt1 mit 140 GB RAM und 16 Kernen

Instanzen 8 verwendet das stochastische Modell. Sie rechnet mit zwei Szenarien, die jeweils mit einer Wahrscheinlichkeit von 50% auftreten. In diesem ersten stochastischen Modell wurde nur scheinbar stochastisch gerechnet, denn beide Szenarien bestehen aus der ursprünglichen Nachfrage von Instanz 1. In diesem Fall gibt es keine Unsicherheit in Instanz 8 und der Zielfunktionswert stimmt mit Instanz 1 überein. Diese Rechnung wurde durchgeführt, um auf die unterschiedlichen Laufzeiten hinzuweisen. Während Instanz 1 nur 241 Sekunden benötigte, braucht Instanz 8 mit 6024 Sekunden fast 25 mal so lange. Dies hängt mit der Modellerweiterung bezüglich der Szenarien zusammen. Da es jede Variable für jedes Szenario gibt, verdoppelt sich bei zwei Szenarien die Variablenanzahl und macht das Optimierungsproblem damit schwieriger.

Instanzen 9 rechnet zwei wirklich verschiedene Szenarien gegeneinander. Wie bereits oben erwähnt sind dies mit einer Wahrscheinlichkeit von jeweils 50% Instanz 6 und Instanz 7. erinnert man sich an Abbildung 4.2, so erkennt man, dass sich Rikschas Typ 1 und der Anhänger kaum unterscheiden und aus dem gleichen Zwischenprodukt bestehen. Damit hat Instanz 9 die folgende Bedeutung: Es ist bekannt, dass das Zwischenprodukt nachgefragt wird, aber nicht in welcher Ausfertigung. Würde man den Erwartungswert der beiden deterministischen Ergebnisse bilden, so ergäbe sich $50\% \cdot 5501,90 \text{ €} + 50\% \cdot 408,00 \text{ €} = 2954,95 \text{ €}$. Das Ergebnis von Instanz 9 liegt mit 2153,30 € darunter. Daran lässt sich sehr gut erkennen, was der Hauptunterschied zwischen Determinismus und Unsicherheit ist. Wenn sicher bekannt ist, welche Mengen der Produkte verkauft werden können, so kann die Produktion gezielt die benötigten Produkte herstellen. Weiß man allerdings nicht genau, welche Produkte nachgefragt werden, so muss man auf Vorrat produzieren und bleibt unter Umständen auf den Produkten sitzen. Dies ist der Grund, warum in beiden Szenarien zum Endzeitpunkt noch vereinzelt Produkte im Lager liegen. In Instanz 9 hat das Modell noch den Vorteil, dass beide Szenarien das gleiche Zwischenprodukt benötigen. So kann dieses zunächst vorproduziert und nach Bekanntwerden der tatsächlichen Nachfrage entsprechend weiterverarbeitet werden. Daher liegt in diesem Fall der Gewinn von Instanz 9 relativ nahe am deterministisch erwarteten Gewinn. Die Laufzeit von Instanz 9 beträgt 2353 Sekunden und ist damit besser als Instanz 8.

Mithilfe des stochastischen Modells sind viele verschiedene Beispielrechnungen denkbar. In **Instanzen 10** wird eine mögliche davon betrachtet, die bewusst relativ komplex gewählt wurde: Während in beiden Szenarien die Nachfrage nach den Rikschas konstant und über den Planungshorizont identisch bleibt, wird die Einführung des Anhängers ohne Rikschas diskutiert. Im ersten Szenario wird der Anhänger nicht eingeführt, die Nachfrage nach Produkt 15, Rikschas mit Anhänger, bleibt unverändert. Im zweiten Szenario wird der Anhänger lose eingeführt. Es wird angenommen, dass die Nachfrage nach dem Kombiprodukt zurückgeht und die Kunden stattdessen Rikschas und Anhänger einzeln kaufen.

Dieses Szenario wurde ebenfalls auf einem großen Server der AG Optimierung⁴ gerechnet. Die Komplexität von Instanz 10 wird durch die Rechenzeit deutlich: Es wurde nach 250000 Sekunden, das sind knapp drei Tage, noch kein Optimalergebnis gefunden und die Rechnung abgebrochen. Daher lässt sich für diese Instanz keine Optimallösung, sondern nur Schranken für den Zielfunktionswert angeben. Die amtierend beste gefundene Lösung beträgt -322,89 €, die Schranke für das relaxierte Problem beträgt ca. 1870,00 €. Dieses Ergebnis zeigt, dass je nach Instanz das Finden einer Optimallösung sehr aufwendig sein kann. Allerdings werden innerhalb weniger Minuten „gute“ Lösungen gefunden. So lautet für Instanz 10 der Zielfunktionswert nach zehn Minuten bereits -1439,00 €. Dieses Szenario soll noch einmal verdeutlichen, dass der Anwender abwägen muss, wie viel Zeit er im Verhältnis zur Optimallösung investieren möchte. Angesichts der Tatsache, dass keine Optimallösung erreicht wurde, erscheint es müßig, die gefundene

⁴ numopt1 mit 140 GB RAM und 16 Kernen

Lösung zu analysieren. Daher wird an dieser Stelle darauf verzichtet und auf die Lösungsdatei auf der CD verwiesen.

Es sind natürlich viele weitere Instanzen vorstellbar, aber die Grundfunktionen des Modelles sind mit den obigen Szenarien weitgehend erläutert worden. Die wichtigsten Hinweise zu der Wahl der Parameter wurden schon in Abschnitt 4.1 gegeben. Abschließend sind in der folgenden Tabelle die Ergebnisse der einzelnen Instanzen noch einmal übersichtlich zusammengefasst.

Tabelle 4.4: Änderungen und Laufzeiten der einzelnen Instanzen. Für Instanz10 wurde keine Optimallösung gefunden und daher nur die beste gefundene zulässige Lösung angegeben.

Instanz	Änderung	Laufzeit in Sek	Gewinn in €
Instanz1	Grundmodell	241	1230,80
Instanz2	Zusätzliche Nachfrage am Mittag	986	-870,00
Instanz3	Nachfrage verteilt	835	4373,30
Instanz4	Planungshorizont 48 Stunden	316812	-1852,00
Instanz5	Transportzeiten null und andere Nachfrage	6182	650,30
Instanz6	Nur Rikscha Typ1 Nachfrage	647	5501,90
Instanz7	Nur Anhänger Nachfrage	51	408,00
Instanz8	Stochastisch Instanz1 gerechnet	6024	1230,80
Instanz9	Stochastisch Instanz6 und Instanz7 als Szenarien	2353	2153,30
Instanz10	Stochastisch Produkteinführungsszenario	> 250000	-322,89

5 Fazit

Ziel dieser Arbeit war es, ein Optimierungsmodell für die innerbetriebliche Losgrößen- und Produktionsplanung zu entwickeln und zu analysieren. Dazu wurde ein bereits bekanntes Modell weiterentwickelt und mithilfe von mathematischer Software anhand verschiedener Beispielszenarien getestet.

Ausgehend von der Nachfrage versucht das Modell, die optimale Produktionsreihenfolge und -menge der einzelnen Produkte zu ermitteln. Dabei werden Restriktionen der Maschinen, des Güterflusses im Modell und der Produktionsmengen berücksichtigt. Ein großer Vorteil des Modells ist die Verwendung von Lieferverzug und -ausfall. Dadurch ist garantiert, dass es für jede beliebige Instanz zulässige Lösungen gibt. In der Realität beendet eine Firma auch nicht sofort die Produktion, nur weil sie nicht fristgerecht liefern kann. Dies ist ein wesentlicher Unterschied gegenüber anderen Optimierungsmodellen, bei denen es u.U. vorkommen kann, dass zu viele Einschränkungen Unlösbarkeit zur Folge haben.

Ein wesentliches Problem der Losgrößenplanung ist, dass es viele unterschiedliche Produktionspläne gibt, die alle einen ähnlichen Zielfunktionswert haben. Dieses Phänomen tritt auch bei der in dieser Arbeit verwendeten Modellierung auf, die sehr viele zulässige Lösungen bedingt. Das macht es selbst für Software schwierig, die optimale Lösung zu finden. Daher kann es bei ungünstiger Wahl der verschiedenen Parameter vorkommen, dass die Rechenzeit nicht mehr akzeptabel ist.

Mithilfe von mathematischer Software wurde das Verhalten des Modells bei Änderung einiger Parameter untersucht. Es wurde ein fiktives Unternehmen betrachtet, welches zwei unterschiedliche Typen von Rikschas, optional mit Anhänger, produziert. Im Laufe der Rechnungen hat es sich als sehr wichtig erwiesen darauf zu achten, dass die Parameter in einem sinnvollem Verhältnis zueinander stehen. Im Ausgangsmodell wurde ein Planungshorizont von 24 Stunden betrachtet und ausschließlich Nachfrage zum Endzeitpunkt betrachtet.

Die Veränderung der Nachfrage stand bei der Untersuchung im Vordergrund. Im deterministischen Fall lässt sich feststellen, dass eine über den Planungshorizont verteilte Nachfrage einen höheren Gewinn erzielt als eine einmalige Nachfrage am Planungsende, da die Lagerkosten geringer ausfallen. Anhand der in der Arbeit getesteten Szenarien kann man schlussfolgern, dass über einen Planungshorizont von 24 Stunden im deterministischen Fall die Optimallösung in wenigen Minuten bestimmt werden kann. Möchte man mit wesentlich mehr Zeitschritten oder stochastischen Szenarien rechnen, so steigt die Laufzeit an. Es empfiehlt sich daher in diesem Fall abzuwägen, wie viel Zeit man im Verhältnis zur Qualität der Lösung aufbringen möchte.

Je nach Planungshorizont könnte man überdenken, ob die Annahme, dass Rohstoffe ausschließlich zu Beginn bestellt werden können, sinnvoll ist. Unter Umständen erscheint es realistischer, dem Modell auch zwischenzeitlich zu erlauben, Rohstoffe nachzukaufen, damit die Lagerkosten für die Rohstoffe nicht ins Unermessliche steigen. Eine weitere Möglichkeit, das Modell zu verbessern, betrifft die Transportzeiten. Die Annahme von Transportzeiten an sich erscheint durchaus sinnvoll, da es Zeit erfordern kann, Produkte aus dem Lager zu holen und zur Weiterverarbeitung vorzubereiten. Allerdings bietet das Modell in seiner jetzigen Form keine Möglichkeit die Transportmenge der einzelnen Produkte festzustellen. Hier könnte man über die Einführung einer Hilfsvariablen nachdenken.

Es bieten sich außerdem zahlreiche Erweiterungsmöglichkeiten des Modells an. Vor allem stochastische Einflüsse, wie sie in dieser Arbeit teilweise schon betrachtet wurden, machen das Modell realitätsnah. Denkt man an die Wirklichkeit, so wird es kaum so sein, dass die Nachfrage im Voraus exakt bestimmt werden kann. Vielmehr wird hier auf Erfahrungen aus der Vergangenheit oder Schätzungen aus Umfragen zurückgegriffen, die nur eine grobe Orientierung vorgeben. Wie man mit der Produktion auf diese Prognosen reagiert, könnte man in Zukunft untersuchen. Sicherlich wäre es für die weitere Analyse auch interessant, das Modell mit realen Daten aus der Praxis zu testen.

Literaturverzeichnis

- [Aig06] AIGNER, Martin: *Diskrete Mathematik*. 6., korr. Aufl. Vieweg, 2006 (Vieweg Studium Aufbaukurs Mathematik). – ISBN 978-3-528-07268-1
- [BDMU11] BORNDÖRFER, Ralf ; DÜR, Mirjam ; MARTIN, Alexander ; ULBRICH, Stefan: *Einführung in die Optimierung*. <https://www3.mathematik.tu-darmstadt.de/index.php?id=84&evsid=23&evsver=932&evsdir=886>. Version: 27.09.2011
- [GPL11] *IBM ILOG CPLEX Optimizer*. <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>. Version: 27.09.2011
- [GLP11] *GNU Linear Programming Kit (GLPK)*. <http://www.gnu.org/s/glpk/>. Version: 27.09.2011
- [Hüc11] HÜCKELHEIM, Arne: *KolkataRickshaw.JPG*. <http://commons.wikimedia.org/wiki/File:KolkataRickshaw.JPG>. Version: 27.09.2011
- [Mar11] MARTIN, Alexander: *Diskrete Optimierung*. <https://www3.mathematik.tu-darmstadt.de/index.php?id=84&evsid=23&evsver=944&evsdir=955>. Version: 27.09.2011
- [Oßw79] OSSWALD, Jürgen: *Produktionsplanung bei losweiser Fertigung*. Wiesbaden : Gabler, 1979 (Beiträge zur industriellen Unternehmensforschung 7). – ISBN 3409342516
- [TA03] THOMMEN, Jean-Paul ; ACHLEITNER, Ann-Kristin: *Allgemeine Betriebswirtschaftslehre*. 4., überarb. und erw. Aufl. Wiesbaden : Gabler, 2003. – ISBN 3409430164
- [Tem06] TEMPELMEIER, Horst: *Material-Logistik*. 6., neubearb. Aufl. Berlin [u.a.] : Springer, 2006. – ISBN 978-3-540-28425-3
- [WS07] WALDMANN, Karl-Heinz (Hrsg.) ; STOCKER, Ulrike M. (Hrsg.): *Operations Research, Proceedings 2006, Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Jointly Organized with the Austrian Society of Operations Research (ÖGOR) and the Swiss Society of Operations Research (SVOR), Karlsruhe, Germany, September 6-8, 2006*. 2007. – ISBN 978-3-540-69994-1



Anhang: CD und Quellcode

Inhalt der CD

Folgende Inhalte befinden sich auf der beigefügten CD:

- Die verwendeten GLPK .mod Dateien ModellDeterministisch.mod und ModellStochastisch.mod
- Die in den jeweiligen Instanzen verwendeten .dat Dateien
- Die aus GLPK heraus erzeugten .lp Dateien
- Die CPLEX Solution-Dateien
- Die CPLEX Lösungen als .txt Datei
- Diese Arbeit als .pdf Datei
- Skripte zur Vorlesung „Einführung in die Optimierung“ [BDMU11] und „Diskrete Optimierung“ [Mar11]

Quellcode

Es folgt die Datei ModellDeterministisch.mod, welche das Ausgangsmodell für Kapitel 4 bildet. Alle weiteren Quellcodes findet man auf der beigefügten CD.

```
#Ausgangsmodell
#Zur besseren Unterscheidung werden Variablen mit Kleinbuchstaben und Parameter mit
    Großbuchstaben gekennzeichnet

#Mengen- und Parameterdefinition

#Menge von Produkten
set Produkte;

#Menge von Rohstoffen (Teilmenge der Produkte)
set Rohstoffe within Produkte;

#Menge an Maschinen
set Maschinen;

#Menge von Prozessen
set Tasks;

#Anzahl der Perioden (0 wird zur Initialisierung benötigt)
param AnzPerioden integer >=1;
set Perioden := 0..AnzPerioden;
```

```

#Maschine, die für Prozess i benötigt wird und das Produkt, dass von Prozess i produziert
  wird
#wird beides aus einer CSV-Datei eingelesen
param Benoetigte_Maschine{i in Tasks};
param Produziertes_Produkt{i in Tasks};
table Tasksettingstable IN "CSV" "Tasksettings.csv" : Tasks <- [Tasks],
  Benoetigte_Maschine ~ Benoetigte_Maschine, Produziertes_Produkt ~
  Produziertes_Produkt;

#Menge von Prozessen, die die gleiche Maschine benutzen wie Prozess i
#wird aus einer CSV-Datei eingelesen
set Tasks_2 dimen 2;
param Gleiche_Maschine{i in Tasks, j in Tasks} binary;
table Gleiche_Maschinetable IN "CSV" "Gleiche_Maschine.csv": Tasks_2 <- [Tasks_i,
  Tasks_j], Gleiche_Maschine ~ Gleiche_Maschine;

#Erlös für eine Einheit von Produkt p
param Erloes{p in Produkte} >=0;

#Lagerkosten für eine Einheit von Produkt p pro Periode
param Lagerkosten{p in Produkte} >=0;

#Backlogkosten für eine Einheit von Produkt p pro Periode
param Backlogkosten{p in Produkte} >=0;

#Fehlmengen-Strafkosten für eine Einheit von Produkt p
param Shortfallkosten{p in Produkte} >=0;

#Setupkosten um zwischen zwei Prozessen zu wechseln
param Setupkosten{i in Tasks, j in Tasks} >=0;

#Produktionskosten einer Einheit von Prozess i
param Produktionskosten{i in Tasks} >=0;

#Demand für Produkt p in Periode t
param Demand{p in Produkte, t in Perioden} >=0 default 0;

#Gozinto-Faktor: 0, falls Produkt q kein unmittelbarer Nachfolger von p ist,
#sonst die Anzahl an Produkt p, die unmittelbar in Produkt Q eingeht.
#wird aus einer CSV-Datei eingelesen
set Produkte_2 dimen 2;
param Gozinto_Faktor{p in Produkte, q in Produkte} >=0;
table Gozinto_Faktortable IN "CSV" "Gozinto_Faktor.csv": Produkte_2 <- [Produkte1,
  Produkte2], Gozinto_Faktor ~ Gozinto_Faktor;

#Transportzeit um Produkt p zum nächsten Prozess zu bringen
param Transportzeit{p in Produkte, i in Tasks};

#maximale Produktionsmenge von Produziertes_Produkt[i] durch Prozess i in Periode t
param Kapazitaetsbeschraenkung{i in Tasks, t in Perioden} >=0;

#Zeit, die benötigt wird, um von Setup i zu Setup j zu wechseln
param Changesetupzeit{i in Tasks, j in Tasks} >=0;

```

```

#ganzzahlige Anzahl Perioden, die für den Setupwechsel benötigt werden
param Changesetupzeitinteger{i in Tasks, j in Tasks} := floor(Changesetupzeit[i,j]);

#reeller Rest für den Wechsel eines Setups
param Changesetupzeitfraction{i in Tasks, j in Tasks} := (Changesetupzeit[i,j] -
    Changesetupzeitinteger[i,j]);

#Variablendeklaration

#Produktionsanzahl von Produziertes_Produkt[i] durch Prozess i in Periode t
var prodmenge{i in Tasks, t in Perioden} >=0, integer;

#Lagerbestand von Produkt p am Ende von Periode t
var lagerbestand{p in Produkte, t in Perioden} >=0, integer;

#Lieferrückstand von Produkt p für externen Bedarf in Periode t
var backlog{p in Produkte, t in Perioden} >=0, integer;

#Fehlmenge von Produkt p in Periode t
var shortfall{p in Produkte, t in Perioden} >=0, integer;

#Binärvariable die anzeigt, ob am Anfang von Periode t ein Setupwechsel zwischen Prozess
    i und j stattfindet (changesetup[i,j,t]=1) oder nicht (changesetup[i,j,t]=0)
var changesetup{i in Tasks, j in Tasks, t in Perioden} binary;

#Binärvariable, die anzeigt, ob Benötigte_Maschine im Zustand von Prozess i ist in
    Periode t (y=1) oder nicht (y=0)
var ready{i in Tasks, t in Perioden} binary;

#Hilfsvariable, die benötigt wird, da GLPK sonst den Erlös nicht berücksichtigt
var one;

#Modellbeginn

#Zielfunktion: maximiere Erlös - (Produktionskosten + (Lagerkosten + Backlogkosten +
    Strafkosten für Nichtbedarfserfüllung) + Setupkosten)
maximize Gewinn:
sum{t in Perioden}{
sum{p in Produkte} Demand[p,t]*Erloes[p]*one
-
    (sum {i in Tasks} prodmenge[i,t]*Produktionskosten[i]
+ sum {p in Produkte} (lagerbestand[p,t]*Lagerkosten[p] + backlog[p,t]*Backlogkosten[p] +
    shortfall[p,t]*Shortfallkosten[p])
+ sum{i in Tasks, j in Tasks: Gleiche_Maschine[i,j]=1 and i != j}
    changesetup[i,j,t]*Setupkosten[i,j]
    )
);

#Nebenbedingung, damit GLPK den Erlös berücksichtigt
s.t. One: one =1;

#Anfangslagerbestände von den Halbfabrikaten und Endprodukten

```

```

s.t. InitProdukte{p in Produkte diff Rohstoffe}: lagerbestand[p,0] = 0;

#Lagerbestandsveränderungen
s.t. Bestandsveraenderung{p in Produkte, t in Perioden:t>=1}:
  sum {i in Tasks: Produziertes_Produkt[i]=p} prodmenge[i,t]
  - sum{q in Produkte, j in Tasks: Produziertes_Produkt[j] = q and
    t+Transportzeit[p,j]<=AnzPerioden}
    prodmenge[j,t+Transportzeit[p,j]]*Gozinto_Faktor[p,q]
  - Demand[p,t] + backlog[p,t] - backlog[p,t-1] + shortfall[p,t]
  = lagerbestand[p,t]-lagerbestand[p,t-1];

#Zusätzliche Nebenbedingungen, die im Originalmodell fehlen
s.t. Produktionsverbot {p in Produkte, q in Produkte, j in Tasks, t in Perioden :
  Produziertes_Produkt[j] = q and Gozinto_Faktor[p,q] > 0 and t-Transportzeit[p,j]>=0
  and t <= 2*Transportzeit[p,j]}:
  prodmenge[j,t]*Gozinto_Faktor[p,q] <= lagerbestand[p,t-Transportzeit[p,j]];
s.t. BacklogAnfang{p in Produkte}: backlog[p,0]=0;
s.t. BacklogEnde{p in Produkte}: backlog[p,AnzPerioden]=0;

#Einschränkung der Lieferrückstände und -ausfälle
s.t. Lagerbeschraenkung{p in Produkte, t in Perioden: t>=1}:
  backlog[p,t]-backlog[p,t-1] + shortfall[p,t] <= Demand[p,t];

#Nebenbedingung, damit sich jede Maschine am Ende jeder Periode in genau einem Zustand
  befindet
s.t. Setupbeschraenkung{m in Maschinen, t in Perioden}:
  sum{i in Tasks: Benoetigte_Maschine[i]=m} ready[i,t] = 1;

#Nebenbedingung, die Setupwechsel identifiziert und die Binärvariablen aneinander koppelt
s.t. Setupaenderung{i in Tasks, j in Tasks, t in Perioden: Gleiche_Maschine[i,j]=1 and i
  != j and t>=1}: ready[i,t-1] + ready[j,t] -1 <= changesetup[i,j,t];

#Kapazitätsbeschränkungen für die Produktionsmengen
s.t. Produktionsbeschraenkung{i in Tasks, t in Perioden}:
  prodmenge[i,t] <= Kapazitaetsbeschraenkung[i,t] *
  (ready[i,t]
  -sum{j in Tasks, z in Perioden: Gleiche_Maschine[i,j]=1 and i != j and
    t-Changesetupzeitinteger[j,i] < z and z<=t and z>=0} changesetup[j,i,z]
  -sum{j in Tasks: Gleiche_Maschine[i,j]=1 and i != j and
    t-Changesetupzeitinteger[j,i]>=0}
    changesetup[j,i,t-Changesetupzeitinteger[j,i]]*Changesetupzeitfraction[j,i]);

solve;

end;

```

ModellDeterministisch.mod