

# Ein Optimierungsmodell für Produktions- und Lagerhaltungsnetze

Dipl.-Inf. Michael Kämpf, Professur Modellierung und Simulation  
Prof. Dr. oec. Dr. rer. nat. habil. Peter Köchel, Professur Modellierung und Simulation  
Technische Universität Chemnitz

## 1. Problembeschreibung und Modellentwurf

Mehrere Produzenten eines bestimmten Spektrums von Endprodukten beschließen zu kooperieren, d.h., ein gemeinsames Management (1) der Vorratshaltung der Endprodukte und (2) der Fertigung dieser Endprodukte zu realisieren. Dazu hat das Management zumindest vier Fragen zu beantworten: *Wann ist welches Produkt mit welcher Losgröße durch welchen Produzenten zu fertigen?* Managementziel ist Maximierung des zu erwartenden Gewinns. Zu diesem Zwecke entwickeln wir zuerst ein Modell, in dessen Rahmen Antworten auf diese Fragen gefunden werden können. Die dabei zu lösenden Probleme sind schon für einfache Situationen äußerst komplex. Darum schlagen wir simulationsbasierte Optimierung als Lösungszugang vor und zeigen seine Anwendbarkeit.

Eine schematische Darstellung des zu optimierenden Fertigungs- und Lagersystems ist in Abb.1 enthalten. Man erkennt zwei Hauptelemente des zu entwickelnden Modells - das Lager für  $N$  Endprodukte und das Fertigungssystem. Das Fertigungssystem bestehe aus  $M$  einstufigen Einheiten, welche die einzelnen Unternehmen repräsentieren. Die Unternehmen seien identisch in dem Sinne, dass sie den gleichen Produktmix fertigen können. Ein Unternehmen solle gleichzeitig nur ein Produkt fertigen können. Produktwechsel erzeuge Rüstkosten und Rüstzeiten. Fertiggestelltes Produkt wird im (gemeinsamen) Lager mit endlicher Kapazität gelagert. Aus dem Lagervorrat wird ein zufälliger Bedarf nach Endprodukten befriedigt. Bedarf, der nicht unmittelbar befriedigt werden kann, wird vorgemerkt und in eine Warteschlange mit endlich vielen Warteplätzen eingereiht. Bedarf, der keinen freien Warteplatz findet, geht verloren. Es werden Kosten für Fertigung und Lagerung der Produkte sowie für wartenden und abgewiesenen Bedarf angesetzt. Verkaufte

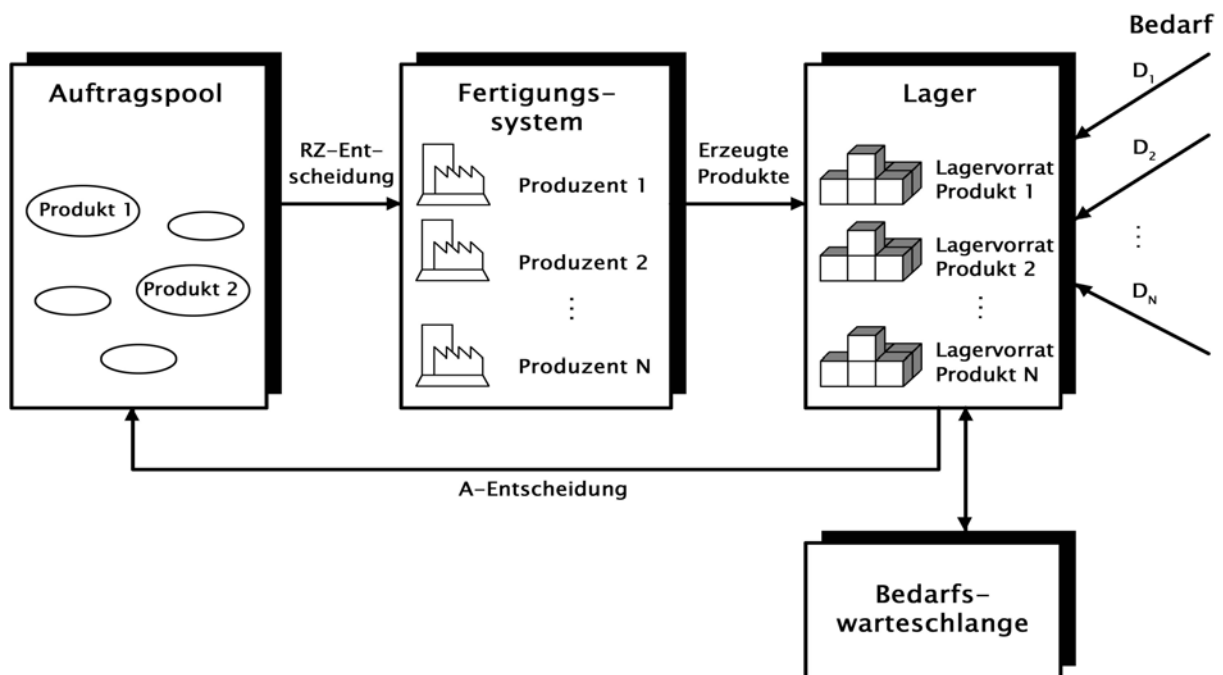


Abbildung 1. Schema des Fertigungs- und Lagersystems

Produkt bringt einen Erlös. Das Entscheidungsproblem besteht nun darin, eine solche Strategie  $\pi$  zu finden, die in Abhängigkeit vom Zustand des Gesamtsystems sowohl Fertigungsaufträge auslöst als auch deren Reihenfolge und Aufteilung auf die Unternehmen so festlegt, dass der je Zeiteinheit zu erwartende Gewinn maximiert wird. Dieses überaus komplexe Entscheidungsproblem zerlegen wir in zwei Teile. Teil 1 bezieht sich auf das Lager und entscheidet in Abhängigkeit vom Lagerzustand welches Produkt in welchem Umfang zu produzieren ist. Dieser Entscheidungsteil heißt Auftragsauslösung (A-Entscheidung). Wir nutzen die aus der Lagerhaltung bekannten (s, Q) und (s, S) Strategien. Sie haben eine einfache Struktur, was für eine Anwendung von Bedeutung ist. Die von der Auftragsauslösung generierten Fertigungsaufträge werden im Auftragspool vor dem Fertigungssystem gespeichert. Ein zweiter Entscheidungsteil legt fest, welcher der wartenden Aufträge als nächster durch welches Unternehmen zu realisieren ist. Die entsprechende Entscheidung wird Reihenfolge-und-Zuordnungs-Entscheidung (RZ-Entscheidung) genannt. In der vorliegenden Version wird die Zuordnungsentscheidung vereinfacht durch die Annahme, dass jeder Produzent jedes Endprodukt fertigen kann. Hat ein Produzent die Fertigung eines Produktes beendet, muss er somit lediglich die Entscheidung treffen, welches Produkt er als nächstes fertigen soll. Wir betrachten dazu statische und dynamische Heuristiken. Zu den statischen Heuristiken gehören *First-Come-First-Served (FCFS)*, die Aufträge werden in der Reihenfolge wie sie vorgemerkt wurden gefertigt, und *Random*, welche den nächsten Auftrag zufällig auswählt. Die dynamischen Heuristiken nehmen Rücksicht auf den aktuellen Systemzustand. So wird bei der *zyklischen Strategie (Cyclic)* ein sich ständig wiederholender Produktionszyklus im Vorfeld festgelegt. Vor der Produktion muss lediglich geprüft werden, ob Bedarf an der Fertigung dieses Produktes besteht, ansonsten muss die Prüfung für das nächste Produkt durchgeführt werden usw. Durch den Einsatz von Warteschlangen für unbefriedigten Bedarf ergeben sich zwei weitere Strategien. Es wird jeweils Rücksicht auf die Anzahl der wartenden Kunden eines Produktes (*Längste Warteschlange (LWS)*) oder die vorgemerkte Menge eines Produktes (*Meiste wartende Menge (MWM)*) genommen.

Zur mathematischen Formulierung des Entscheidungsproblems werden folgende Größen benötigt:

- $P$  - Gesamtkapazität des Lagers,
- $p_i$  - benötigte Lagerkapazität einer Einheit des Endproduktes  $i$ ,
- $h_i$  - Lagerkosten je Zeiteinheit für eine Einheit des Endproduktes  $i$ ,
- $w_i$  - Wartekosten je Zeiteinheit und wartende Bedarfseinheit des Endproduktes  $i$ ,
- $b_i$  - Warteschlangenkapazität für Bedarfsanmeldungen des Endproduktes  $i$ ,
- $r_i$  - Abweiskosten je abgewiesene Bedarfseinheit des Endproduktes  $i$ ,
- $g_i$  - Gewinn je verkaufte Einheit des Endproduktes  $i$ ,
- $f_i$  - Fertigungskosten je Zeiteinheit für eine Einheit des Endproduktes  $i$ ,
- $\mu_i$  - Fertigungsintensität für eine Einheit des Endproduktes  $i$ ,
- $c_{ij}$  - Umrüstkosten bei Wechsel von Endprodukt  $i$  auf Fertigung von Endprodukt  $j$ ,
- $H_i$  - mittlerer Lagervorrat von Endprodukt  $i$ ,
- $W_i$  - mittlerer wartender Bedarf nach Endprodukt  $i$ ,
- $R_i$  - mittlerer je Zeiteinheit abgewiesener Bedarf nach Endprodukt  $i$ ,
- $U_{ij}$  - je Zeiteinheit erfolgte mittlerer Anzahl Umrüstungen von  $i$  auf Endprodukt  $j$ ,
- $O_i(t)$  - Anzahl der zum Zeitpunkt  $t$  wartenden Bedarfsanmeldungen für Endprodukt  $i$ ,
- $M_i(t)$  - Umfang der zum Zeitpunkt  $t$  wartenden Fertigungsaufträge für Endprodukt  $i$ ,
- $I_i^+(t)$  - zum Zeitpunkt  $t$  physisch vorhandener Vorrat von Endprodukt  $i$ .

Kämpf und Köchel (2004) haben gezeigt, dass die Maximierung des je Zeiteinheit zu erwartenden Gewinns äquivalent ist zur Lösung des Minimierungsproblems

$$(MP) \begin{cases} \min_{\pi \in \Pi} C(\pi) \\ \sum_{i=1}^N p_i \cdot I_i^+(t) \leq P \text{ für } t \geq 0; \\ O_i(t) \leq b_i \text{ für } t \geq 0, i = 1, 2, \dots, N \end{cases}$$

mit 
$$C(\pi) = \sum_{i=1}^N \left[ h_i \cdot H_i(\pi) + w_i \cdot W_i(\pi) + (r_i + g_i - f_i / \mu_i) \cdot R_i(\pi) + \sum_{j=1}^N c_{ij} \cdot U_{ij}(\pi) \right].$$

Die Zielfunktion  $C(\pi)$  repräsentiert die je Zeiteinheit unter Strategie  $\pi$  zu erwartenden Kosten, wobei ein modifizierter Abweiskostenfaktor herangezogen wird. Es versteht sich, dass die in (MP) auftretenden Leistungsmaße  $H_i(\pi), W_i(\pi), R_i(\pi)$  und  $U_{ij}(\pi)$  von der gewählten Strategie  $\pi$  abhängen. Durch die Nebenbedingungen werden in (MP) die jeweiligen Kapazitäten für das Lager bzw. die Bedarfswarteschlange beachtet. Offensichtlich kann (MP) selbst unter sehr einschränkenden Annahmen nicht analytisch gelöst werden. Einen Ausweg aus diesem Dilemma bietet die simulationsbasierte Optimierung, auf deren prinzipielle Wirkungsweise im folgenden Abschnitt kurz eingegangen wird.

## 2. Lösungszugang simulationsbasierte Optimierung

Bei der simulationsbasierten Optimierung wird zwischen Simulation einer vorgeschlagenen Lösung und Suche nach einer besseren iteriert (vgl. Abb. 2). Bis zu seinem Abbruch realisiert der Lösungsprozess folgenden Zyklus: Vorschlag einer Lösung, Erzeugung problemrelevanter Daten über ein Simulationsexperiment, Leistungsanalyse auf Basis dieser Daten und Entscheidung über Annahme dieser Lösung oder Fortsetzung der Suche. Um simulationsbasierte Optimierung anwenden zu können, benötigt man zwei Dinge – einen Simulator und einen Optimierer. Als Optimierungswerkzeug bevorzugen wir Genetische Algorithmen (GA), da sie eine Reihe von Vorzügen aufweisen: relative Unabhängigkeit vom Anwendungsgebiet, ausgezeichnete Eignung zur Lösung von sehr allgemeinen Optimierungsproblemen, Robustheit gegenüber zufälliger Störungen. Ein weiterer Grund ist die Tatsache, dass wir GA's in der Vergangenheit erfolgreich zur Lösung der verschiedenen komplexen Optimierungsprobleme angewandt haben (vgl. Köchel et al. 2003, Köchel und Nieländer 2002). Um GA's anwenden zu können, muss ein Optimierungsproblem parametrisiert werden. Das bedeutet, dass wir uns auf Lösungen beschränken, die durch einen Vektor  $x = (x_1, x_2, \dots, x_L)$  mit endlich vielen Komponenten beschrieben werden können. Wenn z.B. die RZ- Strategie fixiert wird als First-Come-First-Served (FCFS) und die A-Entscheidung für alle Produkte durch  $(s, Q)$  – Strategien definiert wird, dann beschreibt  $x = (s_1, s_2, \dots, s_N; Q_1, Q_2, \dots, Q_N)$  eine solche Lösung. Ein Vektor  $x$  heißt Individuum, eine gegebene Komponente Gen. Unser GA arbeitet mit einer Populationsgröße von 50 Individuen. Durch Anwendung von Selektions-, Mutations- und Crossover-Operatoren werden Eltern-Individuen in Kinder-Individuen transformiert. Der oben beschriebene Zyklus wird so lange durchlaufen, bis ein

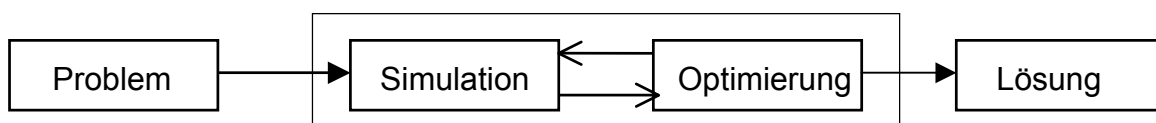


Abbildung 2. Prinzip der simulationsbasierten Optimierung

Abbruchkriterium erfüllt ist. Es ist zu bemerken, dass dieser Suchprozess automatisch abläuft ohne dass ein Nutzer irgendwelche Handlungen ausführen muss. Nachdem der Suchprozess abgebrochen ist, wird die beste gefundene Lösung zurückgegeben. Weitere Ausgaben wie die zweitbeste Lösung oder Informationen über den Optimierungsprozess sind möglich.

Als Simulator für das zu untersuchende System wurde FLSIM entwickelt. FLSIM ist ein *ereignisorientierter* Simulator. Im vorliegenden Fall sind zwei Ereignisgruppen von Bedeutung – Ankunft einer Bedarfsmeldung (eines Kunden) und Vollendung eines Fertigungsauftrages. Um ein Simulationsexperiment realisieren zu können, werden zwei weitere Ereignisse benötigt – die Ereignisse „Ende der Einschwingphase“ und „Ende der Simulation“. Ersteres gibt den Zeitpunkt an, ab dem Information aus der Simulation zu sammeln ist. Letzteres stoppt die Simulation für eine gegebene Lösung.

Beide Werkzeuge, das Simulationsmodell und der GA, sind in der Programmiersprache C# implementiert und an der Professur Modellierung und Simulation verfügbar.

### **3. Ausblick**

Unsere bisherigen Ergebnisse verstehen wir als ersten Meilenstein der Arbeiten zur optimalen Steuerung von Produktions- und Lagerhaltungsnetzen. Weitere Arbeiten werden sich in mehrere Richtungen bewegen. Eine Richtung betrifft die Entwicklung von Modellen und die Formulierung von Steuerungsproblemen, die praxisnah sind und nicht die hier getroffenen vereinfachenden Einschränkungen aufweisen. Eine zweite Richtung wird sich mit der Erweiterung der Strategieklassen und Untersuchungen befassen, unter welchen Bedingungen sich einzelne Klassen dominieren. Das wird nur auf der Basis umfangreicher empirischer Rechnungen möglich sein. Bei einer dritten Richtung geht es vor allem darum, ein nutzerfreundliches und effizientes Softwareprodukt zur Steuerung der betrachteten Systeme zu entwickeln. Da mit enormen Rechenzeiten kalkuliert werden muss, sind vor allem Verfahren der verteilten bzw. parallelen Verarbeitung gefragt. Erfahrungen dazu liegen an der Professur Modellierung und Simulation vor. Beispielsweise werden parallele GA's zur Optimierung eingesetzt. Gleichzeitig liegen mit dem Chemnitzer Linux Cluster (CLiC) beste technische Voraussetzungen an der TU Chemnitz vor.

### **Literatur**

Köchel, P.; Nieländer, U. (2002). Kanban optimization by simulation and evolution. *Production Planning & Control*, v.13, 725-734

Köchel, P.; Kunze, S.; Nieländer, U. (2003). Optimal control of a distributed service system with moving resources: Application to the fleet sizing and allocation problem. *Internat. Journal of Production Economics*, v.81-82, 443-459

Kämpf, M.; Köchel, P. (2004). Simulation-based sequencing and lot size optimisation for a production-and-inventory system with multiple items. *Proceedings of the 13th International Working Seminar on Production Economics, Igls/Innsbruck, Austria, February 16-20, 2004*, pp. 175-184