GOR

Gesellschaft für Operations Research e.V. (GOR)
Netherlands Society for Operations Research (NGB)

H. Fleuren
D. den Hertog · P. Kort
Editors

Operations Research Proceedings 2004

Springer

Operations Research Proceedings 2004

Selected Papers
of the Annual International Conference
of the German Operations Research Society (GOR).
Jointly Organized with the Netherlands Society
for Operations Research (NGB)

Tilburg, September 1–3, 2004

Hein Fleuren · Dick den Hertog
Peter Kort
Editors

Operations Research Proceedings 2004

Selected Papers
of the Annual International Conference
of the German Operations Research Society (GOR).
Jointly Organized with the Netherlands Society
for Operations Research (NGB)

Tilburg, September 1–3, 2004

With 104 Figures
and 59 Tables

Springer

Professor Dr. Hein Fleuren

Professor Dr. Dick den Hertog

Professor Dr. Peter Kort

Warandelaan 2
5037 AB Tilburg
The Netherlands
E-mail: fleuren@uvt.nl
E-mail: d.denhertog@uvt.nl
E-mail: kort@uvt.nl

Cataloging-in-Publication Data

Library of Congress Control Number: 2005924438

ISBN 3-540-24274-0 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springeronline.com

Cover design: Erich Kirchner
Production: Helmut Petri
Printing: Strauss Offsetdruck

SPIN 11375272 Printed on acid-free paper – 42/3153 – 5 4 3 2 1 0

Preface

This volume contains a selection of papers referring to lectures presented at the symposium “Ŏperations Research 2004” (OR 2004) held at Tilburg University, September 1-3, 2004. This international conference took place under the auspices of the German Operations Research Society (GOR) and the Dutch Operations Research Society (NGB).

The symposium had about 500 participants from more than 30 countries all over the world. It attracted academicians and practitioners working in various fields of Operations Research and provided them with the most recent advances in Operations Research and related areas in Economics, Mathematics, and Computer Science.

The program consisted of 4 plenary and 19 semi-plenary talks and more than 300 contributed presentations, selected by the program committee, to be presented in 20 sections.

Due to a limited number of pages available for the proceedings volume, the length of each article as well as the total number of accepted contributions had to be restricted. Submitted manuscripts have therefore been reviewed and 59 of them have been selected for publication.

We would like to thank the GOR-board for the abundant collaboration, which we always found to be helpful and fruitful. We are grateful to all the reviewers, which were asked to review one or more submitted papers. Finally, we would like to thank our colleague Annemiek Dankers and Barbara Fess from Springer-Verlag for their support in publishing this proceedings volume.

Tilburg, March 2005

Hein Fleuren
Dick den Hertog
Peter Kort

Committees

Organizing Committee

Prof.dr. Hein Fleuren (chair), Tilburg University
Dr. Willem Haemers, Tilburg University
Dr. Herbert Hamers, Tilburg University
Drs. Maaike van Krieken, Tilburg University
Mw. Ilse van de Pol, Tilburg University
Prof.dr. Dolf Talman, Tilburg University

Program Committee

Prof.dr. Micheal Jünger, University of Cologne
Prof.dr. Thomas Spengler, Braunschweig University of Technology
Prof.dr. Brigitte Werners, Ruhr-Universität-Bochum
Prof.dr. Rainer Kolisch, Technische Universität München
Prof.dr. Dick den Hertog, Tilburg University
Prof.dr. Peter Kort, Tilburg University
Prof.dr. Marc Salomon, Tilburg University
Prof.dr. Henk Zijm, University of Twente

Reference Committee

Prof.dr. Frank van der Duyn Schouten, Tilburg University
Prof.dr. Lodewijk Kallenberg, Leiden University
Prof.dr. Alexander Rinnooy Kan, Board ING-group
Prof.dr. Jan Karel Lenstra, National Research Institute for Mathematics and Computer Science
Prof.dr. Lex Schrijver, The National Research Institute for Mathematics and Computer Science in the Netherlands
Prof.dr. Stef Tijs, Tilburg University

Sections and section leaders

Reverse Logistics
Rommert Dekker (Erasmus University Rotterdam)
Karl Inderfurth (Otto-von-Guericke-Universität Magdeburg)

OR in Entertainment and Sports
Gerard Sierksma (University of Groningen)
Andreas Drexl (Christian-Albrechts-Universität zu Kiel)

Service Management
Ger Koole (Free University of Amsterdam)
Stefan Helber (Universität Hannover)

Telecommunication
Stan van Hoesel (University of Maastricht)
Arie Koster (Zuse Zentrum für Informationstechnik Berlin)

Production, Logistics and Supply Chain Management
Jalal Ashayeri (Tilburg University)
Hans-Otto Günther (Technischen Universität Berlin)

Transportation and Traffic
Cees Ruijgrok (Tilburg University)
Dirk Mattfeld (University of Bremen)

Project Management and Scheduling
Han Hoogeveen (Universiteit Utrecht)
Rainer Kolisch (Technische Universität München)

Marketing
Tammo Bijmolt (Tilburg University)
Reinhold Decker (Bielefeld University)

Energy, Environment and Health
Paul van Beek (Wageningen University)
Peter Letmathe (Universität Siegen)

Optimization of Bio Systems
Stephan Pickl (Universität zu Köln)

Finance, Banking and Insurances
Antoon Kolen (University of Maastricht)
Hermann Locarek-Junge (Technische Universität Dresden)

Simulation and Applied Probability
Jack Kleijnen (Tilburg University)
Ulrich Rieder (University of Ulm)

Continuous Optimization
Kees Roos (Delft University of Technology)
Florian Jarre (Heinrich-Heine-Universität Düsseldorf)

Discrete and Combinatorial Optimization
Gerhard Woeginger (University of Twente)
Josef Kallrath (BASF-AG, Ludwigshafen)

A.I., Fuzzy Logic and Multicriteria Decision Making
Heinrich Rommelfanger
(Johann Wolfgang Goethe-Universität Frankfurt am Main)

Econometrics, Game Theory and Mathematical Economics
Dolf Talman (Tilburg University)
Clemens Puppe (Universität Karlsruhe)

Managerial Accounting
Anja De Waegenaere (Tilburg University)
Hans-Ulrich Küpper (Ludwig-Maximilians-Universität München)

Business Informatics
Hennie Daniels (Tilburg University)
Leena Suhl (Universität Paderborn)

OR in Engineering
Dick den Hertog (Tilburg University)
Otto Rentz (Universität Karlsruhe)

Revenue Management
Gerrit Timmer (Free University of Amsterdam)
Alf Kimms (Technische Universität Bergakademie Freiberg)

Contents

GOR Awards

Reverse Logistics

Service Management

Production, Logistics and Supply Chain Management

Transportation and Traffic

Project Management and Scheduling

Marketing

Energy, Environment and Health

Optimization of Bio Systems

Finance, Banking and Insurances

Simulation and Applied Probability

Continuous Optimization

Discrete and Combinatorial Optimization

A.I., Fuzzy Logic and Multicriteria Decision Making

Econometrics, Game Theory and Mathematical Economics

Business Informatics

OR in Engineering

Revenue Management

Koordination von Abruf- und Lieferpolitiken in Supply Chains

Eric Sucky

Seminar für Logistik und Verkehr, Johann Wolfgang Goethe-Universität,
Mertonstr. 17, 60054 Frankfurt am Main, esucky@wiwi.uni-frankfurt.de

1 Einleitung

Der Materialfluss in Supply Chains, von der Rohstoffgewinnung über die einzelnen Veredelungsstufen bis hin zum Endkunden, resultiert aus der Verknüpfung von Beschaffungs-, Produktions- und Transportprozessen einzelner, in der Supply Chain agierender Unternehmen. Die zeitlichen und quantitativen Ausprägungen dieser Wertschöpfungsprozesse werden in individuellen Abruf- und Lieferpolitiken spezifiziert. Aufgrund der bestehenden zeitlichen und quantitativen Interdependenzen ist die zielgerichtete, unternehmensübergreifende Koordination von Abruf- und Lieferpolitiken eine zentrale Aufgabe des Supply Chain Management.

Sowohl in der relevanten Literatur als auch in kommerziellen Softwaresystemen werden Planungsansätze des Supply Chain Management weitgehend auf der Basis zentraler Koordinationsprinzipien diskutiert. In real existierenden Supply Chains sind i.d.R. jedoch nicht die Voraussetzungen für eine zentrale Koordination gegeben: Eine zentrale Koordination bedingt, dass die Unternehmen in der Supply Chain 1.) ihre Planungsautonomie aufgeben, 2.) alle planungsrelevanten Informationen (d.h. auch sensible Kosten- und Kapazitätsinformationen) einer zentralen Koordinationsinstanz zur Verfügung stellen und 3.) ihre Handlungen an übergeordneten Supply Chain-Zielen ausrichten [8]. Darin ist der Grund zu sehen, warum kommerzielle Softwaresysteme des Supply Chain Management (Advanced Planning Systems), bisher lediglich unternehmensintern eingesetzt werden, obwohl sie für einen unternehmensübergreifenden Einsatz entwickelt wurden [9].

Zentrale Planungsansätze bilden relevante Problemstellungen des Supply Chain Management nicht adäquat ab. Praxisrelevante Fragestellungen der Allokation von Supply Chain-Gewinnen in Win-Win-Situationen oder der Kompensation von Akteuren in Win-Lose-Situationen sowie relevante Rahmenbedingungen, insbesondere Informationsbedingungen und Machtverhältnisse, werden nur bedingt oder gar nicht berücksichtigt. Zur Unterstützung der

Koordination des zielgerichteten Zusammenwirkens der verteilten Leistungserstellung in Supply Chains liefern zentrale Planungsansätze daher nur einen geringen Beitrag. Während produktionswirtschaftliche und logistische Ansätze zur Planung integrierter Materialflüsse durch eine weitgehende Vernachlässigung von Wettbewerbsaspekten und der Dominanz hierarchischer Koordinationsprinzipien gekennzeichnet sind, eignen sich spieltheoretische Ansätze, die strategische Interaktion zwischen Unternehmen zu analysieren und eine dezentral abgestimmte Planung zu unterstützen [4].

2 Das Koordinationsproblem

Dem zu analysierenden Koordinationsproblem liegen folgende Annahmen zu Grunde: Es wird ein Abnehmer (A) und ein Lieferant (P) eines bestimmten Produkts betrachtet, für das eine im Zeitverlauf gleichbleibende Nachfrage in konstanter Höhe besteht. Für die Phasen des Materialflusses werden auf Lieferantenseite Produktion und Warenausgangslager und auf Abnehmerseite Beschaffung und Wareneingangslager, wie in Abbildung 1 dargestellt, unterschieden.

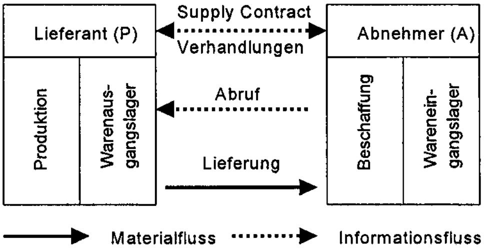


Abb. 1. Betrachtete Lieferanten-Abnehmer-Beziehung

Ist die vom Lieferanten an den Abnehmer zu liefernde Periodenquantität gegeben, muss festgelegt werden, zu welchen Lieferzeitpunkten welche Lieferquantitäten erfolgen, d.h. es muss eine gemeinsame Abruf- und Lieferpolitik ermittelt werden. Die Abrufpolitik des Abnehmers lässt sich durch die Anzahl der Abrufe je Periode, den Abrufzyklus, d.h. der Zeitspanne zwischen zwei Abrufen sowie der Quantität je Abruf charakterisieren. Der Lieferant legt simultan die Produktions- und Lieferpolitik fest. Während sich die Lieferpolitik an der Anzahl der Lieferungen je Periode, dem Lieferzyklus sowie der Quantität je Lieferung konkretisiert, ist die Produktionspolitik durch die Anzahl der Losauflagen je Periode, dem Produktionszyklus sowie der Losgröße je Losauflage gekennzeichnet.

2.1 Individuelle Planungsprobleme der Supply Chain-Partner

Die bekannte Periodenquantität des vom Lieferanten an den Abnehmer zu liefernden Produkts, gemessen in Mengeneinheiten je Periode, beträgt b [ME/PE]. Der Lieferant, der das vom Abnehmer nachgefragte Produkt im Rahmen einer Serien- bzw. Sortenfertigung erzeugt, plant simultan seine Liefer- und Produktionspolitik unter dem Ziel der Minimierung seiner relevanten Periodenkosten. Dabei ist zu beachten, zu welchem Zeitpunkt fertiggestellte Produkteinheiten eines Produktionsloses zur Weitergabe an den Abnehmer zur Verfügung stehen. Im Folgenden wird angenommen, dass der Lieferant seine Liefer- und Produktionspolitik auf der Basis einer Lot-for-Lot-Politik disponiert, d.h. direkt nach Abschluss der Bearbeitung eines Produktionsloses wird das gesamte Produktionslos vollständig an den Abnehmer weitergegeben; die Lieferquantität entspricht der Produktionslosgröße.

Die Periodenkapazität des Lieferanten beträgt d [ME/PE]. Es gilt $d > b$ und die Nachfrage des Abnehmers wird durch d/x Lieferungen identischer Quantität x [ME] gedeckt. Die variablen Produktions- und Transportkosten je Mengeneinheit werden als im Planungszeitraum konstant angenommen und sind daher nicht entscheidungsrelevant. Die Summe der relevanten Rüst- und Lagerkosten je Periode stellt somit die Zielgröße dar; als Höhenpräferenzrelation gilt „Minimierung“. Die Rüstkosten, die bei jeder Losauflage in gleicher Höhe anfallen, betragen R [GE]. Der Lagerhaltungskostensatz sei mit h_P $[GE/ME \cdot PE]$ bezeichnet. Die relevanten Periodenkosten, die individuell optimale Produktions- und Lieferpolitik sowie die resultierenden minimalen Periodenkosten des Lieferanten lauten [1]:

$$z_P(x) = R \cdot \frac{b}{x} + \frac{x}{2} \cdot \frac{b}{d} \cdot h_P, \; x_P^* = \sqrt{\frac{2 \cdot B \cdot d}{h_P}}, \; z_P(x_P^*) = b \cdot \sqrt{\frac{2 \cdot R \cdot h_P}{d}}. \qquad (1)$$

Der Abnehmer plant seine Abrufpolitik ebenfalls unter dem Ziel der Minimierung seiner eigenen entscheidungsrelevanten Periodenkosten, welche die Abruf- und Lagerhaltungskosten umfassen. Die variablen Beschaffungskosten je Stück werden als im Planungszeitraum konstant angenommen und sind daher nicht entscheidungsrelevant. Unter der Annahme einer unendlich schnellen Lieferzeit sowie eines kontinuierlichen Lagerabgangs mit konstanter Nachfragerate ist der Gesamtbedarf b [ME/PE] durch b/x Abrufe identischer Quantität x [ME] zu decken. Die Abrufkosten, die bei jedem Abruf in gleicher Höhe anfallen, betragen B [GE]. Der Lagerhaltungskostensatz beträgt h_A $[GE/ME \cdot PE]$. Die Funktion der relevanten Periodenkosten, die individuell optimale Abrufpolitik sowie die daraus resultierenden minimalen Periodenkosten des Abnehmers ergeben sich mit:

$$z_A(x) = B \cdot \frac{b}{x} + \frac{x}{2} \cdot h_A, \; x_A^* = \sqrt{\frac{2 \cdot B \cdot b}{h_A}}, \; z_A(x_A^*) = \sqrt{2 \cdot B \cdot b \cdot h_A}\,. \tag{2}$$

2.2 Die integrierte Liefer- und Abrufpolitik der Supply Chain-Partner

Die Koordinationsaufgabe des Supply Chain Management besteht in der hier betrachteten Problemstellung in der zielgerichteten Abstimmung der dezentral ermittelten Teilpläne der Supply Chain-Partner, d.h. in der Koordination der Abruf- und Lieferpolitik des Abnehmers und des Lieferanten. Eine integrierte Abruf- und Lieferpolitik ist durch die Übereinstimmung von Abruf- und Liefermenge gekennzeichnet. Das Problem der Ermittlung einer integrierten Abruf- und Lieferpolitik kann als Vektorminimumproblem formuliert werden:

$$\text{"min"}\left\{ z = \begin{pmatrix} z_A(x) \\ z_P(x) \end{pmatrix} = \begin{pmatrix} B \cdot b & \frac{1}{2} \cdot h_A \\ R \cdot b & \frac{1}{2} \cdot \frac{b}{d} \cdot h_P \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{x} \\ x \end{pmatrix} \middle| \, x \in R_+ \right\}. \tag{3}$$

Existiert eine Abruf- und Lieferpolitik $\hat{x}$, für die beide Zielfunktionen $z_A(x)$ und $z_P(x)$ ihre individuellen Minima erreichen, so wird diese Lösung als perfekte Lösung von (3) bzw. als perfekte Abruf- und Lieferpolitik bezeichnet. Eine perfekte Lösung existiert jedoch nur, wenn die individuell optimale Abrufmenge und die individuell optimale Produktions- und Lieferlosgröße identisch sind. In diesem Fall gilt $\hat{x} = x_A^* = x_P^*$ und es ergibt sich der ideale Zielfunktionswertvektor:

$$\hat{z}' = (z_A(\hat{x}), z_P(\hat{x})) = \left(\sqrt{2 \cdot B \cdot b \cdot h_A}, b \cdot \sqrt{\frac{2 \cdot R \cdot h_P}{d}} \right). \tag{4}$$

In realen Planungssituationen wird die individuell optimale Abrufpolitik nicht der individuell optimalen Lieferpolitik entsprechen. Im Allgemeinen gilt $x_A^* \neq x_P^*$ und das Vektorminimumproblem (3) besitzt keine perfekte Lösung. Die Akteure müssen eine Kompromisslösung finden, d.h. sie müssen sich z.B. im Rahmen bilateraler Verhandlungen über die zu realisierende Abruf- und Lieferpolitik einigen. Kandidaten für eine solche Verhandlungslösung sind Abruf- und Lieferpolitiken, die bezüglich der Zielfunktionen $z_A(x)$ und $z_P(x)$ funktional-effizient sind. Die Menge der funktional-effizienten Abruf- und Lieferpolitiken ist [10]:

$$X^{\text{eff}} = \left\{ x \in R_+ \middle| x \in [x_A^*, x_P^*] \right\}. \tag{5}$$

Ausgehend von einer funktional-effizienten Abruf- und Lieferpolitik $x \in X^{eff}$ kann kein Akteur besser gestellt werden (im Sinne der Realisierung niedrigerer Periodenkosten), ohne dass der jeweils andere Akteur schlechter gestellt wird. Für jede Politik $x \notin X^{eff}$ hingegen, kann mindestens eine Politik $x \in X^{eff}$ angegeben werden, für die beide Akteure niedrigere Periodenkosten realisieren. Nicht funktional-effiziente Alternativen sind daher keine potenziellen integrierten Abruf- und Lieferpolitiken, ihre Wahl verstößt gegen den Grundsatz formaler Rationalität [5]. Die Ermittlung der Menge funktional-effizienter Politiken (5) reduziert die Komplexität des dem Vektorminimumproblem (3) zu Grunde liegenden Koordinationsproblems, eine endgültige Auswahl der integrierten Abruf- und Lieferpolitik gelingt aber nicht. Es existiert für $x_A^* \neq x_P^*$ folgender Zielkonflikt:

$$z_A(x_P^*) > z_A(x) > z_A(x_A^*) \text{ und } z_P(x_A^*) > z_P(x) > z_P(x_P^*) \, \forall x \in]x_A^*, x_P^*[\tag{6}$$

Ausgehend von einer funktional-effizienten Abruf- und Lieferpolitik führt die Verringerung des Zielfunktionswertes eines Akteurs zur Erhöhung des Zielfunktionswertes des jeweils anderen Akteurs. In Abbildung 2 sind für $x_A^* < x_P^*$ die streng konvexen Verläufe der relevanten Periodenkosten der Akteure dargestellt. Verhalten sich die Supply Chain-Partner rational im Sinne ihrer Zielsetzungen, so strebt der Lieferant die Realisierung seiner individuell optimalen Produktions- und Lieferpolitik an, während der Abnehmer seine individuell optimale Abrufpolitik realisieren möchte. Lediglich in dem unrealistischen Fall $x_A^* = x_P^*$ besteht somit kein Interessenkonflikt und das Koordinationsproblem ist gelöst.

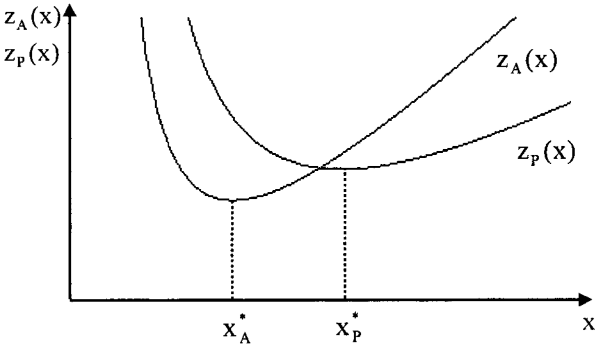


Abb. 2. Verläufe der relevanten Periodenkosten

Im praxisrelevanten Fall $x_A^* \neq x_P^*$ müssen sich die Akteure z.B. im Rahmen bilateraler Verhandlungen über die zu realisierende Abruf- und Lieferpolitik einigen. In der relevanten betriebswirtschaftlichen Literatur wird vorgeschlagen, die integrierte Abruf- und Lieferpolitik so zu bestimmen, dass die Summe der

Periodenkosten der Akteure ihr Minimum annimmt (vgl. z.B. [1], [2] und [3]). Dies impliziert jedoch einen zentralen Planungsansatz auf der Basis einer kompensatorischen Kompromisszielfunktion mit identischer Gewichtung der individuellen Zielfunktionswerte. Neben den in Abschnitt 1 genannten Hemmnissen einer zentralen Koordination bleiben insbesondere die Machtverhältnisse der Supply Chain-Partner unberücksichtigt.

Im Weiteren wird angenommen, dass sich der Abnehmer in der Machtposition befindet, seine individuell optimale Abrufpolitik auch gegen den Widerstand des Lieferanten durchzusetzen (zur Analyse alternativer Machtkonstellationen in Lieferanten-Abnehmer-Beziehungen vgl. [10]). Der Abnehmer hat nur dann einen Anreiz, einer anderen als seiner individuell optimalen Abrufpolitik zuzustimmen, wenn für ihn daraus eine so genannte Win-Situation resultiert. Da für jede integrierte Abruf- und Lieferpolitik $x \neq x_A^*$ gilt $z_A(x) > z_A(x_A^*)$, ergibt sich für ihn jedoch eine Lose-Situation. Der Abnehmer hat somit zunächst keinen Anreiz, eine andere als seine individuell optimale Abrufpolitik zu wählen und durchzusetzen. Ist die Abrufpolitik mit x_A^* gegeben, so ergibt sich aus der Perspektive des Lieferanten eine Lose-Situation, da gilt: $z_P(x_A^*) > z_P(x_P^*)\ \forall\ x_A^*, x_P^* > 0, x_A^* \neq x_P^*$.

Im Rahmen von bilateralen Verhandlungen möchte der Lieferant daher eine Einigung über eine gemeinsame Abruf- und Lieferpolitik x_G erzielen, für die gilt: $z_P(x_P^*) \leq z_P(x_G) < z_P(x_A^*)$. Damit der Abnehmer eine Abruf- und Lieferpolitik $x_G \neq x_A^*$ akzeptiert, muss mindestens die dadurch bei ihm induzierte Kostenerhöhung durch eine Transferzahlung s [GE/PE] vom Lieferanten kompensiert werden. Das Planungsproblem des Lieferanten besteht somit in der simultanen Ermittlung der anzubietenden Abruf- und Lieferpolitik sowie der damit korrespondierenden Transferzahlung. Ein Angebot des Lieferanten bezeichnet daher eine Kombination aus Quantität je Lieferung x_G und Transferzahlung s.

3 Die Verhandlungslösung

In der relevanten Literatur wird analysiert, wie der Lieferant die Abrufpolitik des Abnehmers durch Preisnachlässe beeinflussen kann (vgl. z.B. [7] und [6]). Diese Planungsansätze bilden das zu Grunde liegende Verhandlungsproblem jedoch nicht adäquat ab. Insbesondere sind die bestehenden Planungsansätze durch die Annahme der vollständigen Information des Lieferanten bezüglich der Ausprägung der Kostenfunktion des Abnehmers gekennzeichnet. Praxisrelevante Verhandlungssituationen werden jedoch dadurch gekennzeichnet sein, dass der Lieferant nur über unvollständige Informationen verfügt, d.h. die Kostenstruktur des Abnehmers nicht mit Sicherheit kennt. Da der Abnehmer keinen Anreiz hat, die wahre Ausprägung seiner Kostenfunktion preiszugeben – der Abnehmer hat im Gegenteil einen Anreiz, für jede angebotene Abruf- und Lieferpolitik überhöhte Kosten anzugeben, um eine möglichst hohe Transferzahlung zu erzielen – muss der Lieferant die Ausprägung der Kostenfunktion des Abnehmers bei der Angebotsgestaltung schätzen.

Aus der Perspektive des Lieferanten sind mehrere Ausprägungen der Kostenfunktion des Abnehmers denkbar. Der Lieferant muss dann ein ganzes Set von Angeboten ermitteln, d.h. auf Basis jeder angenommenen Ausprägung der Kostenfunktion des Abnehmers ist ein spezielles Angebot zu generieren. Die einzelnen Angebote müssen einerseits individuell rational sein, d.h. es muss sich für den Abnehmer lohnen, ein Angebot auch anzunehmen. Andererseits müssen die Angebote anreizkompatibel sein: Der Abnehmer muss einen Anreiz haben, genau das Angebot anzunehmen, welches auf der Basis seiner tatsächlichen Kostenfunktion generiert wurde. Da dem Abnehmer ein Set von Angeboten vorgelegt wird, aus dem er selbst ein Angebot auswählen kann, wird diese Form der Angebotsgestaltung auch als Selbstwahlmechanismus oder Self-Selection bezeichnet. Im Weiteren bezeichnet $z_i^A(x_G)$ ($i \in \{1,2,...,m\}$) die i-te potenzielle Ausprägung der Kostenfunktion des Abnehmers. Es sei w_i die (subjektive) Wahrscheinlichkeit, mit der der Lieferant annimmt, dass die i-te Ausprägung der Kostenfunktion des Abnehmers die tatsächliche ist, mit $w_i > 0 \ \forall i, \sum_{i=1}^{m} w_i = 1$. Schließlich sei $(x_{G,i}, s_i)$ die angebotene Kombination aus Lieferquantität und Transferzahlung, für die i-te angenommene Ausprägung der Kostenfunktion des Abnehmers. Wird angenommen, der Lieferant sei risikoneutral, so lässt sich zur Lösung des stochastischen Planungsproblems ein Ersatzmodell auf der Basis des Erwartungswertes der relevanten Periodenkosten des Lieferanten heranziehen:

$$\min \ E[z_P(x_{G,1}, x_{G,2}, ..., x_{G,m}, s_1, s_2, ..., s_m)] = \sum_{i=1}^{m} w_i \cdot \left(z_P(x_{G,i}) + s_i\right) \tag{7}$$

unter Beachtung der Restriktionen

$$z_i^A(x_{G,i}) - s_i \le z_i^A(x_i^*) \qquad \forall i \in \{1,2,...,m\} \tag{8}$$

$$z_i^A(x_{G,i}) - s_i \le z_i^A(x_{G,j}) - s_j \qquad \forall \ i, j \in \{1,2,..., m\}, \ i \neq j \tag{9}$$

$$x_{G,i} \ge 0 \,, \ s_i \ge 0 \qquad \forall i \in \{1,2,...,m\} \tag{10}$$

Die Zielfunktion (7) gibt den Erwartungswert der relevanten Periodenkosten des Lieferanten an. Die Restriktionen (8) sind die Teilnahmebedingungen (individual-rationality constraints) und bei den Restriktionen (9) handelt es sich um die Anreizkompatibilitätsbedingungen (self-selection constraints). Insgesamt müssen $m + m \cdot (m-1)$ Teilnahme- und Anreizkompatibilitätsbedingungen beachtet werden. Aufgrund der strengen Konvexität der Kostenfunktionen $z_P(x)$ und $z_i^A(x) \ \forall i \in \{1,...,m\}$ stellen die *Karush-Kuhn-Tucker*-Bedingungen sowohl die notwendigen als auch die hinreichenden Optimalitätsbedingungen dar.

Speziell im Fall $i \in \{1,2\}$, d.h. wenn der Lieferant zwei potenzielle Ausprägungen der Kostenfunktion des Abnehmers für möglich hält, kann das den Erwartungswert der relevanten Periodenkosten des Lieferanten minimierende Angebotsset direkt aus den *Karush-Kuhn-Tucker*-Bedingungen ermittelt werden [10].

4 Fazit

Die Bestimmung integrierter Abruf- und Lieferpolitiken stellt ein Verhandlungsproblem dar, das dadurch gekennzeichnet ist, dass die Supply Chain-Partner zwar das gemeinsame Interesse an einer Einigung bezüglich des Verhandlungsgegenstands haben, sie aber individuell sehr unterschiedliche Einigungsresultate herbeiführen wollen. Verhalten sich beide Akteure individuell rational im Sinne ihrer Zielsetzungen, so strebt der Abnehmer die Realisierung seiner individuell optimalen Abrufpolitik an, während der Zulieferer seine individuell optimale Liefer- und Produktionspolitik realisieren möchte. Dieses interdependente Planungsproblem mit interpersonellen Zielkonflikten kann als ein kooperatives Zwei-Personen-Nichtkonstantsummenspiel identifiziert werden. In realistischen Verhandlungssituation mit asymmetrischen Machtverhältnissen und beiderseitiger unvollständiger Information besteht für den stärkeren Supply Chain-Partner zunächst kein Anreiz, eine andere als die individuell optimale Lösung zu wählen und diese gegenüber dem schwachen Akteur durchzusetzen. Eine andere als die individuell optimale Lösung des starken Akteurs ergibt sich nur, wenn der starke Akteur für sein Abweichen „belohnt“ bzw. kompensiert wird. Es wurden daher Transferzahlungen berücksichtigt, die für den Fall verbindlich zugesagt werden können, dass starke Akteure von ihren individuell optimalen Lösungen abweichen. Die Ermittlung der anzubietenden Abruf- und Lieferpolitik sowie der korrespondierenden Transferzahlung, stellt für den schwachen Supply Chain-Partner ein Planungsproblem unter Risiko dar. Der schwache Akteur muss sich eine Wahrscheinlichkeitsverteilung über die möglichen Ausprägungen der Funktion der entscheidungsrelevanten Periodenkosten des starken Akteurs bilden auf deren Basis er ein umfassendes Angebotsset ermitteln muss. Die einzelnen Angebote müssen sowohl individuell rational als auch anreizkompatibel sein. Aufgrund der Komplexität dieses Verhandlungsproblems bei Risiko ist eine softwaretechnische Unterstützung des Entscheidungsträgers anzustreben. Es wurde daher ein prototypisches Verhandlungsunterstützungssystem entwickelt und implementiert, das den Entscheidungsträger bei der Generierung und Bewertung von Angeboten interaktiv unterstützt. Sowohl Aufbau und Funktionalität dieser Software-Applikation als auch ihr Einsatzpotenzial sowie die Integration in bestehende Supply Chain Management-Systeme werden in [11] dargestellt.

Literatur

[1] Banerjee, A. (1986a) A Joint Economic-Lot-Size Model for Purchaser and Vendor. In: Decision Science, Vol. 17, S. 292-311.

[2] Goyal, S. K. (1988) „A Joint Economic-Lot-Size Model for Purchaser and Vendor“: A Comment. In: Decision Science, Vol. 19, S. 236-241.

[3] Goyal, S. K.; Nebebe, F. (2000) Determination of economic production – shipment policy for a single-vendor-single-buyer system. In: European Journal of Operational Research, Vol. 121, S. 175-178.

[4] Inderfurth, K.; Minner, S. (2001) Produktion und Logistik. In: Jost, P. J. (Hrsg.), Die Spieltheorie in der Betriebswirtschaftslehre. Stuttgart, S. 307-349.

[5] Isermann, H. (1974) Lineare Vektoroptimierung. Dissertation, Regensburg.

[6] Miller, P. A.; Kelle, P. (1998) Quantitative Support for Buyer-Supplier Negotiation in Just-In-Time Purchasing. In: International Journal of Purchasing and Materials Management, Spring 1998, S. 25-29.

[7] Monahan, J. P. (1984) A Quantity Discount Pricing Model to increase Vendor Profits. In: Management Science, Vol. 30, No. 6, S. 720-726.

[8] Pibernik, R.; Sucky, E. (2004) Zentrales und dezentrales Supply Chain Planning. In: WiSt Wirtschaftswissenschaftliches Studium, Heft 1/2004, S. 25-33.

[9] Steven, M.; Krüger, R. (2002) Advanced Planning Systems – Grundlagen, Funktionalitäten, Anwendungen. In: Busch, A.; Dangelmaier, W. (Hrsg.), Integriertes Supply Chain Management. Wiesbaden, S. 169-186.

[10] Sucky, E. (2004a): Koordination in Supply Chains: Spieltheoretische Ansätze zur Ermittlung integrierter Bestell- und Produktionspolitiken, Wiesbaden.

[11] Sucky, E. (2004b): Softwaregestützte Gestaltung von Supply Contracts. In: Suhl, L., Voß, S. (Hrsg.), Quantitative Methoden in ERP und SCM, DSOR Beiträge zur Wirtschaftsinformatik 2, Paderborn, S. 77-98.

Ein Entscheidungsunterstützungssystem zur Verschnittoptimierung von Rollenstahl

Ingmar Steinzen

Decision Support & Operations Research Lab, Universität Paderborn,
Warburger Str. 100, 33098 Paderborn, Email: steinzen@uni-paderborn.de

Abstract. Im Rahmen einer Zusammenarbeit mit der Firma Stahlwerk Ergste-Westig GmbH, einem Unternehmen der ZAPP AG, wurde ein Verfahren zur Lösung eines 1.5-dimensionalen Verschnittproblems entwickelt und prototypisch implementiert. Das Problem zeichnet sich durch einen begrenzten Lagerbestand mit sehr heterogenem Sortiment und zusätzlichen Nebenbedingungen zur Materialverwendung aus. Es wird durch ein lineares ganzzahliges Modell abgebildet und mittels einer Branch-and-Cut Strategie unter Einsatz der MIP-Solver MOPS und CPLEX gelöst. Dabei wird durch ein um die Berücksichtigung von Range-Bedingungen erweitertes MOPS IP-Preprocessing eine entscheidende Verbesserung der Lösungsqualität und Verkürzung der Rechenzeit erreicht. Das implementierte Entscheidungsunterstützungssystem konnte die realen Probleminstanzen effizient und im Sinne des Unternehmens lösen und erzielte in den durchgeführten Vergleichsrechnungen ein Reduzierungspotential für den Verschnitt von etwa 30% zum bisherigen Vorgehen. Unter Verwendung des Systems ist neben der Ersparnis durch die Verschnittreduzierung auch ein deutliches Einsparpotential an personellen Ressourcen durch den im Vergleich zur manuellen Disposition geringen Zeitbedarf vorhanden.

1 Einführung und Problemstellung

Das Service-Center (SC) der Stahlwerk Ergste-Westig GmbH vertreibt hochwertigen rostfreien Federbandstahl. Federbandstahl ist auf Rollen gewickeltes Präzisband und wird dem Service-Center als Halbfabrikat geliefert. Aufgabe des Service-Centers ist es, Kundenaufträge durch Zuschneiden der gewünschten Breite zu bedienen. Im Lager des Service-Centers sind begrenzte Mengen verschiedener Ausgangsbreiten (Chargen) vorrätig. Beim Zuschnitt werden jährlich mehrere hundert Tonnen Material in kleinere Breiten (längs) zerschnitten, wobei das eingesetzte Material oft nicht komplett verbraucht wird, so dass nicht weiter verwertbare Reststücke entstehen können. Ziel der Verschnittoptimierung war es, Schnittmuster zu finden, die die Kundenaufträge erfüllen und den Verschnitt unter Berücksichtigung der Herstellkosten minimieren. Weitere Ziele aus Sicht des Unternehmens waren eine reproduzierbare Dispositionsstrategie und eine (halb)automatische Materialsuche für Aufträge über alle Lagerorte und Bestandsstufen.

Das Service-Center fertigt und liefert ausschließlich nach Kundenbestellungen, die im Wesentlichen die Materialanforderungen, die Auftragsmengen in Kilogramm und die Breite der Streifen definieren. In der Logistik werden für die Kundenaufträge zunächst alle zulässigen Chargen im Lager gesucht und daraufhin Schnittmuster (siehe Abbildung 1) bestimmt, die möglichst wenig Verschnitt erzeugen. Zur Zeit wird eine manuelle Verschnittoptimierung durch Bündelung von Aufträgen durchgeführt.

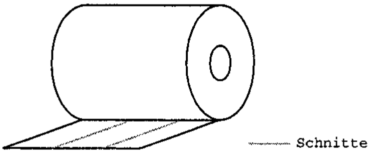


Fig. 1. Schematische Darstellung eines Schnittmusters

Ein Kundenauftrag legt eine Reihe von Materialmerkmalen fest, anhand derer alle Chargen aus dem Lager ausgewählt werden können, die ein zulässiges Einsatzmaterial für den Auftrag darstellen. Die Merkmale sind entweder durch das Material eindeutig festgelegt oder können durch den Fertigungsprozess verändert werden.

Sind alle gültigen Einsatzmaterialien für eine Menge von Aufträgen gefunden, so können daraus Schnittmuster aufgezählt werden. Ein Schnittmuster ist eine Schneidanweisung einer Charge auf einer Maschine, die die Anzahl und die Breite der zu schneidenden Streifen festlegt. Jeder Streifen wird genau einem Kundenauftrag zugeordnet. Ein Schnittmuster kann prinzipiell aus vielen Streifen verschiedener Breiten bestehen, wobei alle fertigungstechnischen Beschränkungen der ausgewählten Maschine zu beachten sind. Eine Charge kann quer zur normalen Schneidrichtung geteilt werden. Eine Querteilung kann nur über die gesamte Breite eines Coils erfolgen (Guillotinenschnitt).

Nach der Generierung der zulässigen Schnittmuster sind Restriktionen und Ziele zu definieren, unter denen die Optimierung durchgeführt werden soll, um aus der Gesamtmenge der Schnittmuster die optimalen auswählen zu können:

Anforderung 1 (Auftragsmengenrestriktion) *Jeder durch den Lagerbestand erfüllbare Kundenauftrag muss durch die ausgewählten Schnittmuster in ausreichender Menge beliefert werden. Eine begrenzte, anteilige Unter- oder Überlieferung eines Auftrags ist zulässig. Die Unter- und Überlieferquoten sind grundsätzlich als konstant anzusehen.*

Anforderung 2 (Lagerbestandsrestriktion) *Für eine Menge von Schnittmustern dürfen nur die tatsächlich im Lager befindlichen Chargen eingesetzt werden.*

Jede Charge entstammt genau einer Schmelze. Mehrere Chargen können aus derselben Schmelze vorgefertigt worden sein.

Anforderung 3 (Schmelzenrestriktion) *In Abhängigkeit von seiner Auftragsmenge darf ein Auftrag nur durch Verwendung von Chargen aus einer begrenzten Anzahl verschiedener Schmelzen erfüllt werden.*

Anforderung 4 (Zielsetzung) *Zielsetzung der Verschnittoptimierung sind Minimierung des Verschnitts und der Herstellkosten. Die Vergleichbarkeit beider Zielsetzungen muss sichergestellt werden, daher wird der Verschnitt eines Schnittmusters mit Kosten bewertet. Jedem Schnittmuster wird die Summe aus Herstellkosten und Verschnittkosten als Gesamtkosten zugewiesen.*

Der Gilmore-Gomory Ansatz ([3], [4]) löst die kontinuierliche Relaxation für Verschnittprobleme mit verschiedenen Breiten der großen Objekte durch Column-Generation Techniken und rundet die im Allgemeinen nicht ganzzahlige Lösung. Dieser heuristische Ansatz führt bei einer Standardbreite oft zu einem Optimum, da der abgerundete optimale Zielfunktionswert der LP-Relaxation oft dem des ganzzahligen Problems entspricht ([5], [8]).

Für allgemeine eindimensionale Verschnittprobleme (1-CSP) mit großen Auftragsmengen konnten in den vergangenen Jahren LP-basierte exakte Methoden erfolgreich zur Lösung eingesetzt werden ([2], [7]). Es wurde dabei eine Kombination von Branch-and-Bound und Column-Generation, auch Branch-and-Price genannt, zur Lösung großer Modelle eingesetzt. Valério de Carvalho greift dabei auf eine Arc-Flow Formulierung zurück, die übrigen Modelle auf die Gilmore-Gomory Formulierung. De Carvalho verweist in seiner Übersicht [6] auf die Erweiterbarkeit der genannten Modelle auf verschiedene Standardbreiten der großen Objekte. Ein Cutting-Plane-Algorithmus wurde von Belov und Scheithauer [1] ebenfalls mit verschiedenen Standardbreiten getestet.

2 Mathematisches Modell

Das im vorhergehenden Abschnitt beschriebene Verschnittproblem wird analog zum Gilmore-Gomory Modell formuliert und um einen begrenzten Lagerbestand mit sehr heterogenem Sortiment sowie zusätzliche Nebenbedingungen zur Materialverwendung erweitert.

Das eindimensionale Verschnittproblem wird durch die Daten $(p, P, b = (b_1, \ldots, b_{p+P}), g = (g_1, \ldots, g_{p+P}))$ charakterisiert. Hierbei sind p die Anzahl der Aufträge, P die Anzahl der zur Verfügung stehenden Chargen, $b_1, \ldots, b_p$ die Auftragsbreiten in mm, $b_{p+1}, \ldots, b_P$ die Breiten der Chargen in mm, $g_1, \ldots, g_p$ die Auftragsgewichte in kg und $g_{p+1}, \ldots, g_P$ die Chargengewichte in kg. Sei $q = p + P$ und seien $(B_1, \ldots, B_P) = (b_{p+1}, \ldots, b_q)$, $(G_1, \ldots, G_P) = (g_{p+1}, \ldots, g_q)$ die Teilvektoren der Chargen.

Ein Vektor $a = (a_1, \ldots, a_q)^T \in \mathbb{Z}_+^q$ beschreibt ein zulässiges Schnittmuster, falls $\sum_{i=1}^p a_i b_i \leq \sum_{j=p+1}^q a_j b_j$ (Rucksack-Bedingung) und $\sum_{j=p+1}^q a_j = 1$. Die Elemente a_i mit $1 \leq i \leq p$ legen fest, wie oft die Auftragsbreite b_i geschnitten wird, ohne die Breite der Charge zu überschreiten. Die Komponente a_{p+j^*} bestimmt die für das Schnittmuster benutzte Charge j^* und wird

Eins gesetzt, alle anderen Elemente a_{p+j} für $1 \leq j \leq P, j \neq j^*$ sind Null. Das spezifische Ringgewicht $\rho_j = G_j/B_j$ bestimmt das Gewicht eines 1mm breiten Streifens der Charge j. Für das Ringgewicht eines Schnittmusters h gilt $\rho_h = \rho_{j^*}$.

Sei n die Anzahl aller Schnittmuster. Die Matrix $A = (a_{ih}) \in \mathbb{Z}_+^{q \times n}$ mit den Schnittmustern als Spalten wird Schnittmustermatrix genannt. Einer Schnittmustermatrix A ist ein Kostenvektor $c := (c(a))_{a \in A}$ zugeordnet. Die Schnittmustermatrix und der Kostenvektor werden explizit definiert und müssen während der Initialisierung des Modells generiert werden.

Einem Schnittmuster ist über die Charge genau eine von l Schmelzen zugeordnet. Mehrere Chargen und somit auch Schnittmuster können dieselbe Schmelze besitzen. Ein Vektor $s^i = (s_1, \ldots, s_l)^T \in \{0,1\}^l$ stellt eine zulässige Schmelzenzuordnung von Auftrag i und einem Schnittmuster dar, falls die Ungleichung $\sum_{k=1}^{l} s_k \leq 1$ gilt. Das Element s_{k^*} mit $1 \leq k^* \leq l$ wird Eins gesetzt, wenn die Charge des Schnittmusters zu Schmelze k^* gehört, alle anderen Komponenten s_k mit $1 \leq k \leq l, k \neq k^*$ sind Null. Es sind genau dann alle Elemente des Vektors s^i Null, wenn das Schnittmuster nicht für Auftrag i eingesetzt wird.

Die Matrix $S^i = (s^i_{kh}) \in \{0,1\}^{l \times n}$ mit den Schmelzenzuordnungen als Spalten wird als Schmelzenzuordnungsmatrix von Auftrag i bezeichnet. Ein Auftrag i darf mit maximal d^i verschiedenen Schmelzen beliefert werden.

Jeder Auftrag kann um die konstanten Faktoren $u \in [0,1]$ unterliefert und $o \geq 1$ überliefert werden.

Man nehme an, die Schnittmustermatrix, Schnittmusterkosten und Schmelzenzuordnungsmatrizen seien bekannt, dann lautet das ganzzahlige lineare Modell wie folgt:

$$\sum_{h=1}^{n} c_h x_h \rightarrow \min \tag{1}$$

$$s.t. \quad g_i u \leq \sum_{h=1}^{n} a_{ih} \rho_h b_i x_h \leq g_i o \qquad (1 \leq i \leq p) \tag{2}$$

$$\sum_{h=1}^{n} a_{jh} x_h \leq 1 \qquad (p < j \leq q) \tag{3}$$

$$\sum_{h=1}^{n} s^i_{kh} x_h \leq (P+1) y^i_k \qquad (1 \leq k \leq l, 1 \leq i \leq p) \tag{4}$$

$$\sum_{k=1}^{l} y^i_k \leq d^i \qquad (1 \leq i \leq p) \tag{5}$$

$$y^i_k \in \{0,1\} \tag{6}$$

$$x_h \in \{0,1\} \tag{7}$$

Die Optimierungsaufgabe ist demnach, den Schnittintensitätsvektor $x = (x_1, \ldots, x_n)^T$ zu bestimmen, so dass die Kosten minimiert (1), alle Aufträge erfüllt (2), jede Charge maximal einmal verwendet (3) und für jeden Auftrag die maximalen Schmelzenanzahlen eingehalten werden, siehe (4) und (5).

3 Lösungsverfahren

Für das in den vorangehenden Abschnitten beschriebene Problem und formalisierte Modell wurde ein Entscheidungsunterstützungssystem implementiert. Das System erhält als Eingabe die Auftrags- und Bestandsdaten aus dem im Unternehmen eingesetzten SAP R/3 System. Zusätzlich werden die Eigenschaften der vorhandenen Maschinen, wie z.B. die maximal zulässige Streifenanzahl, sowie eine Reihe weiterer Parameter übergeben. Die Parameter umfassen bspw. die Unter- und Überlieferungsquoten und die Strategie zur Generierung der Schnittmuster. Die Ausgabe des Systems umfasst eine Beschreibung der ausgewählten Schnittmuster mit allen fertigungsrelevanten Informationen und den benötigten Verschnitt.

Das in Abbildung 2 dargestellte Lösungsverfahren kann in drei Phasen unterteilt werden, die im Folgenden kurz beschrieben werden.

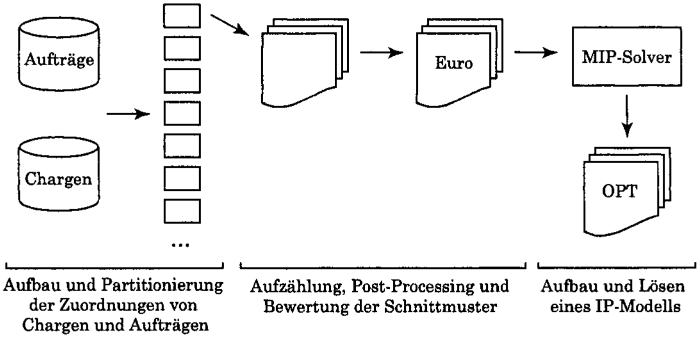

Aufträge
Chargen
Euro
MIP-Solver
OPT
...
Aufbau und Partitionierung der Zuordnungen von Chargen und Aufträgen
Aufzählung, Post-Processing und Bewertung der Schnittmuster
Aufbau und Lösen eines IP-Modells

Fig. 2. Lösungsverfahren

In Phase eins wird ein Zuordnungsgraph erstellt, der jeder Charge alle im Sinne der Anforderungen zulässigen Aufträge zuordnet. Ist nur das spezifische Ringgewicht einer Charge nicht in der Spezifikation eines Kundenauftrags, kann durch Querteilungen versucht werden, die Anforderung zu erfüllen und eine Zuordnung zu erstellen. Die Zuordnungen zwischen Aufträgen und Chargen sind bidirektional. Jede Zuordnung wird im Anschluss validiert, um spätere Unzulässigkeiten im ganzzahligen linearen Modell abzufangen. Es wird u.a. für jeden Auftrag überprüft, ob die Auftragsmenge theoretisch durch alle zugeordneten Chargen aufgebracht werden kann. Alle unzulässigen Zuordnungen werden aus dem Graph entfernt. Der Zuordnungsgraph wird anschließend in disjunkte Teilgraphen partitioniert.

Für jede Charge und ihre zugeordneten Aufträge innerhalb eines Teilproblems werden in Phase zwei mit Hilfe eines Backtracking-Algorithmus alle möglichen Schnittmuster aufgezählt. Optional können neben den die Chargenbreite bestmöglich ausnutzenden Schnittmustern zusätzlich auch nichtmaximale Schnittmuster aufgezählt werden. Jedes einzelne Schnittmuster wird durch ein gesondertes Post-Processing auf fertigungstechnische Machbarkeit überprüft, ggf. angepasst und mit den fertigungsrelevanten Informationen versehen. Für jedes Schnittmuster wird die günstigste Maschine bestimmt. Aus den Herstell- und Verschnittkosten werden die Gesamtkosten eines Schnittmusters berechnet und dem Schnittmuster zugewiesen.

In der letzten Phase werden für jedes Teilproblem die Schnittmustermatrix, der Kostenvektor und die Schmelzenzuordnungsmatrizen aus den Aufträgen und den bewerteten Schnittmustern aufgebaut und das entsprechende ganzzahlige lineare Modell initialisiert.

Das ganzzahlige Optimierungsmodell jedes Teilproblems wird durch ein Branch-and-Cut-Verfahren iterativ gelöst. Für den ersten Lösungsversuch werden dem Modell in der Regel nicht alle Restriktionen hinzugefügt. Wird in der Lösung eine der ausgelassenen Restriktionen verletzt, so wird sie dem Modell angefügt und ein erneuter Lösungsversuch gestartet.

Kann in einem so initialisierten Modell beim ersten Lösungsversuch mit einem Standard-MIP-Solver wie MOPS oder CPLEX keine ganzzahlige Lösung gefunden werden, werden den Auftragsmengenrestriktionen kontinuierliche Schlupfvariablen hinzugefügt.

Weiterhin wird durch ein um die Berücksichtigung von Range-Bedingungen erweitertes MOPS IP-Preprocessing eine entscheidende Verbesserung der Lösungsqualität und Verkürzung der Rechenzeit erreicht.

4 Ergebnisse

Die Anzahl der generierten Schnittmuster eines Problems hängt von den vorhandenen Chargen- und Auftragsbreiten ab, insbesondere aber auch von der Anzahl der gefundenen Auftragszuordnungen einer Charge. Große Chargenbreiten, kleine Auftragsbreiten sowie eine hohe Anzahl von Zuordnungen pro Charge lassen die Schnittmusteranzahl zum Teil sehr stark wachsen. Umspulquerschnitte in Kombination mit einer moderaten Anzahl von Querteilungsfaktoren stellen ein wirksames Instrument zur Minimierung der nicht zuzuordnenden Aufträge dar. Diese Minimierung ist gerade aus Sicht des Unternehmens als sehr wichtig anzusehen. Mit dieser Minimierung geht eine zum Teil deutliche Erhöhung der Schnittmusteranzahl einher.

Bei einem Vergleich der MIP-Solver MOPS 6.37 und ILOG CPLEX 8.0 anhand kleiner, mittlerer und großer Probleminstanzen wurde deutlich, dass sich ein Großteil der kleinen und mittleren Instanzen mit beiden Solvern in kurzer Zeit lösen lassen. Bei schwierigen Problemen konnten allerdings mit

CPLEX deutlich bessere Ergebnisse in Bezug auf Zeitbedarf und Lösungsqualität erzielt werden.

Des Weiteren wurde die Wirksamkeit des implementierten Branch-and-Cut Verfahrens zur Reduzierung der Lösungszeit verifiziert, wobei insbesondere kontinuierliche Schlupfvariablen in den Auftragsmengenrestriktionen zu einer drastischen Verschlechterung des IP-Preprocessings sowie der Ableitung von Cuts führten. Mit dieser Verschlechterung ging eine signifikante Erhöhung der Lösungszeit einher. Eine Reduktion der Zeilenanzahl durch das iterative Hinzufügen verletzter Schmelzenrestriktionen verringerte den Zeitbedarf zur Lösung kleiner und mittlerer Instanzen.

Aus wirtschaftlicher Sicht ist eine Maximierung der Lieferquote sinnvoll. Diese Maximierung kann durch das implementierte System nicht vollautomatisch erreicht werden, sondern erfordert den Eingriff des Bedieners. Die vorgeschlagene iterative Vorgehensweise, durch Variation der Parameter eine Erhöhung der Lieferquote zu erreichen, ist auch aus Sicht des Unternehmens wünschenswert.

In einer Vergleichsrechnung anhand zweier realer Eingabesätze zur Quantifizierung des Reduzierungspotentials konnte ein realistisches Einsparpotential von mindestens 30% identifiziert werden (siehe Abbildung 3), was einer monetären Einsparung eines deutlich sechsstelligen Eurobetrags pro Jahr entspricht. Durch den geringen Zeitbedarf des Systems im Vergleich zur manuellen Disposition ist weiterhin ein eindeutiges Einsparpotential an personellen Ressourcen erkennbar.

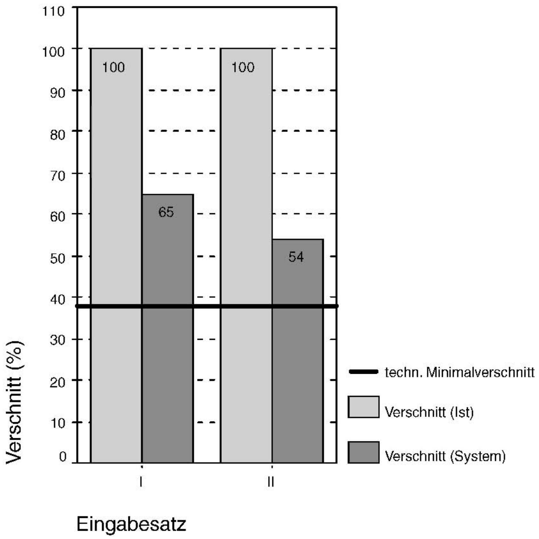


Fig. 3. Verschnittvergleich von Ist und implementiertem System

Nach der Minimierung der Verschnitte über den gesamten Horizont war die Betrachtung geringerer Vorgriffszeiträume sinnvoll, da mit steigendem Vorgriff das Risiko einer Auftragsdatenänderung steigt. Eine solche Änderung kann dazu führen, dass schon gefertigte Streifen unter Umständen nicht mehr ausgeliefert werden können. Ein geringer Vorgriffszeitraum bei der Auftragsselektion von ein bis zwei Perioden führte bei den Eingabesätzen zwar zu sehr geringen Verschnitten, diese gingen allerdings mit der Entstehung vieler breiter Reststücke einher. Vor dem Hintergrund der Ergebnisse scheint ein möglichst weiter Vorgriff sinnvoll, da sich durch die hohe Anzahl an Kombinationsmöglichkeiten Schnittmuster mit geringen Restbreiten und Verschnitten finden lassen.

Bei der Vergleichsrechnung traten sehr große Instanzen mit bis zu 700000 Spalten auf. Die Skalierbarkeit des Systems ist auch bei sehr großen Probleminstanzen durch Strategien wie das manuelle Splitting oder die Auftragsmengenbeschränkung gewährleistet. Beide Strategien reduzierten die Problemgröße und die Lösungszeit durch die systematische Deaktivierung von Aufträgen. Es konnte gezeigt werden, dass diese Verringerung der Problemgröße nicht zu einer Erhöhung der entstehenden Verschnitte führt.

Das implementierte Entscheidungsunterstützungssystem wurde vom Unternehmen als valide und effizient akzeptiert und wird zur Zeit von einer internen Projektgruppe in das SAP R/3 System des Unternehmens zur operativen Nutzung integriert.

References

1. Belov, B., Scheithauer, G. (2002): A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths. European Journal of Operational Research **141[2]**, 274–294
2. Degraeve, Z., Peeters, M. (2003): Optimal integer solutions to industrial cutting stock problems: Part 2, benchmark results. INFORMS Journal on Computing **15**, 58–81
3. Gilmore, P. C., Gomory, R. E. (1961): A linear programming approach to the cutting-stock problem. Operations Research **9**, 849–859
4. Gilmore, P. C., Gomory, R. E. (1963): A linear programming approach to the cutting-stock problem, Part II. Operations Research **11**, 863–888
5. Scheithauer, G., Terno, J. (1995): The Modified-Integer-Round-Up Property of the one-dimensional cutting stock problem. European Journal of Operational Research **84(3)**, 562–571
6. Valério de Carvalho, J. M. (2002): LP models for bin-packing and cutting stock problems. European Journal of Operational Research **141[2]**, 253–273
7. Vanderbeck, F. (1999): Computational study of a column generation algorithm for bin packing and cutting stock problems, Mathematical Programming A**86** **[3]**, 565–594
8. Wäscher, G., Gau, T. (1995): Heuristics for the one-dimensional cutting stock problem: A computational study. OR Spektrum **18(3)**, 131–144

Conflict-free Real-time AGV Routing

Rolf H. Möhring, Ekkehard Köhler, Ewgenij Gawrilow, and
Björn Stenzel

Technische Universität Berlin, Institut für Mathematik, MA 6-1,
Straße des 17. Juni 136, 10623 Berlin, Germany
{moehring,ekoehler,gawrilow,stenzel}@math.tu-berlin.de

Abstract. We present an algorithm for the problem of routing Automated Guided Vehicles (AGVs) in an automated logistic system. The algorithm avoids collisions, deadlocks and livelocks already at the time of route computation (conflict-free routing). After a preprocessing step the real-time computation for each request consists of the determination of a shortest path with time-windows and a following readjustment of these time-windows. Both is done in polynomial-time. Using goal-oriented search we get computation times which are appropriate for real-time routing. Additionally, in comparison to a static routing approach, used in Container Terminal Altenwerder (CTA) at Hamburg Harbour, our algorithm had an explicit advantage.

1 Introduction

Nowadays automation in logistic systems is very popular. In such an automated logistic system Automated Guided Vehicles (AGVs) are used for transportation tasks and the control of these AGVs is the key to an efficient transportation system. Usually, the aim is to maximize the throughput.

Here, control means computation of routes (routing) on the one hand and collision avoidance on the other hand. The assignment of transportation tasks to AGVs is not part of this work.

Various aspects of the considered problem are of importance. The routes must be computed in appropriate time (for a real-time computation) with respect to the physical properties of the AGVs. The collision avoidance particularly has to deal with the dimensions of the AGVs. In our approach collisions are avoided already at the time of route computaion. This is called conflict-free routing.

One of the first paper on routing (free-ranging) AGVs without causing collisions was done by Broadbent et al. [3] in 1987. Krishnamatury, Batta and Karwan [9] discussed the problem for the special case when all requests are known right from the beginning.

We consider the online problem where requests appear sequentially and one must answer each request without having any information about later arriving requests. We extend the approaches of Huang, Palekar and Kapoor [7] and Kim and Tanchoco [8], respectively. In particular, we take physical properties of the AGVs into consideration in a more exact and flexible way and present an efficient algorithm for the problem.

2 The Model

Let $G = (V, A)$ be a directed graph which represents the lanes of the automated transportation system and let $\tau(a)$ be the transit time on arc a. Then an online routing algorithm has to deal with a sequence $\sigma = r_1, \ldots, r_n$ of requests. Each request $r_j = (s_j, t_j, \theta_j)$ represents a task with start node s_j and end node t_j while θ_j denotes the desired starting time.

2.1 Problems with static routing approaches

In static approaches for this problem one computes a route without taking time dependencies into consideration. This can be done by standard shortest path algorithms, i.e., the Dijkstra algorithm (see [4]). In such an approach congestion can be considered by additionally using load dependent arc costs (see [1]). The arising routes are not conflict-free and therefore one needs an additional conflict management at real-time. To guarantee that there are no collisions the moving AGVs must allocate the area they want to use next.

The advantage of this approach is clear: it is easy to implement a fast routing algorithm. But the disadvantages of the described collision avoidance, namely the appearance of deadlocks and livelocks (see Fig. 1), have an enormous effect on the performance of the system.

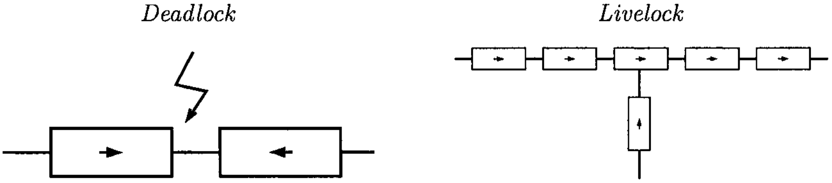


Fig. 1. Deadlocks and Livelocks

Deadlocks appear if two or more AGVs wish to allocate the same area. None is able to continue its route and the system is blocked. Livelock is the generic term for situations where an AGV is blocked repeatedly by other AGVs without having the possibility of allocating the area which is next on its route.

2.2 Conflict-free routing

In order to avoid the problems of the simple model given in Section 2.1, we compute conflict-free routes because in a conflict-free approach there is no need for an additional collision avoidance.

There are two key ingredients which must be considered in that approach. On the one hand, one has to deal with the physical dimensions of the AGVs because they usually have to claim several arcs in the directed graph at the same time. On the other hand, the approach has to be time-dependent (dynamic).

To avoid conflicts we use polygons $P(a)$ for each arc a which describe the blocked area when an AGV (the center of an AGV) is on arc a. Thus, it is prohibited to use two arcs at the same time if the corresponding polygons intersect each other. For each arc a, this leads to a set *confl*(a) of so called geographic-dependent arcs which should not be used at the same time.

The time-dependent behavior is modelled by time intervals. If an AGV travels over an arc a during the interval $[t_1, t_2]$, all geographic-dependent arcs are blocked from t_1 to t_2. This leads to time-windows for each arc of the directed graph G. These time-windows represent the times when arc a is free. After routing a request one has to readjust these time-windows according to the used arcs and their geographic-dependent arcs. If this is done consequently, one does not have to take care of the AGV dimension in route computation since it is already considered.

3 The Algorithm

The algorithm consists of a preprocessing step and (for each request) a route computation followed by a readjustment of the appropriate time-windows.

3.1 Preprocessing

In a preprocessing step all polygons $P(a)$ are compared pairwise. If two polygons $P(a)$ and $P(b)$, for arcs $a, b \in A(G)$, intersect, a is assigned to *confl*(b) and b is added to *confl*(a).

Additionally, in this preprocessing step the behavior of the AGVs is modelled by a list $OUT(a)$ of arcs, containing the set of arcs, which are allowed to be used after arc a respecting the physical properties of the AGV.

3.2 Route computation: shortest paths with time-windows

As pointed out in Section 2.2, the route computation can be done in an idealized model where the dimension of the AGV is not considered explicitly. One just has to compute a route for an infinitesimal point which represents the center of the AGV.

This problem is known as Shortest Path Problem with Time-Windows (SPPTW) [5,10]: Given are a graph G, a source node s, a destination node t, a start time θ, transit times τ_a, costs c_a and a set of time-windows $\mathcal{F}_a$ on each arc a. The aim is to compute a shortest path (concerning costs c) respecting the given time-windows. Here "respecting" means that AGVs wait and traverse the corresponding arc only during (open) time-windows.

The SPPTW is $\mathcal{NP}$-hard. The hardness can be shown by reduction of the of the Constrained Shortest Path Problem (CSPP, see [2]).[1]

Our algorithm for this problem is a generalized arc-based Dijkstra algorithm which deals with labels. A label $L = (a_L, d_L, I_L, pred_L)$ on an arc a_L consists of a distance value d_L, a predecessor $pred_L$ and a time interval I_L. Each label L represents a path from start node s to the tail of a_L, whereas d_L contains the overall transit time (travelling and waiting together) and the label interval $I_L = (A_L, B_L)$ represents the possible arrival times at arc a_L (at the tail of a_L). $pred_L$ is the predecessor of a_L on that path.

We define an order for these labels.

Definition 1 *A label L dominates a label L' if and only if*

$$d_L \leq d_{L'} \quad \textit{and} \quad I_{L'} \subseteq I_L.$$

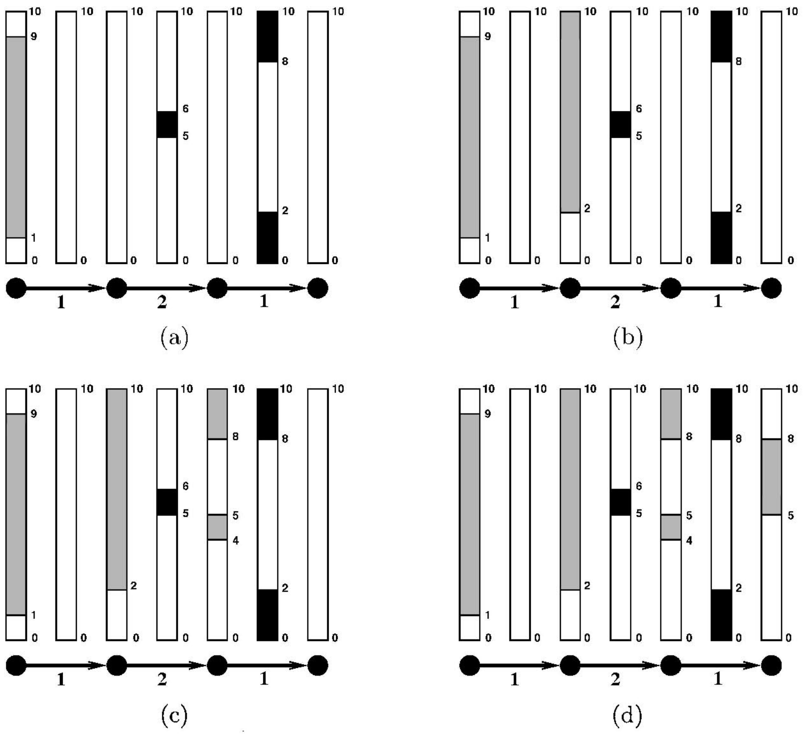
The labels are stored in a priority queue H (a binary heap for example). The generalized arc-based Dijkstra algorithm works as follows.

- **Initialization.**
  Create a label $L = (a, 0, (\theta, \infty), nil)$ for all outgoing arcs a of s and add them to the priority queue H.
- **Loop.**
  Take the label L with lowest distance value d_L from H. If there is no label left in the queue, notify that there is no path from s to t. If t is the tail of a_L, write the corresponding path to output.
  - **For** each time-window on a_L.
    - **Label Expansion.**
      Try to expand the label interval along a_L through the current time-window (new label interval should be as large as possible, see Fig. 2), add the costs $c(a_L)$ to the distance value and set the predecessor. If there is no possible expansion, consider the next time-window.
    - **Dominance Test.**
      Add the new label to each outgoing arc a in $OUT(a_L)$, if it is not dominated by any other label on a.

In general, the algorithm cannot be executed in polynomial-time, unless $\mathcal{P} \neq \mathcal{NP}$. But in our setting we have special costs on the arcs: the transit times of the paths including the corresponding waiting times. In that case we get a polynomial-time algorithm, because costs (distance values) correlate with the lower bounds of the label intervals.

Theorem 2 *In the case of transit times (including waiting times) as cost function, the described generalized arc-based Dijkstra algorithm solves the SPPTW in polynomial-time.*

[1] The instance of the SPPTW is constructed by placing time-windows $[0, R]$ at each arc while R denotes the resource constraint in the CSPP instance.



(a) (b) (c) (d)

Fig. 2. Label Expansion. The label intervals are represented by grey bars (nodes). The blockings are colored black (arcs). The white intervals between these blockings are the time-windows. The figures (a) to (d) show the successive expansion of the label intervals.

Proof. The algorithm computes all required paths since the expansion of the label intervals is maximal and no required path (label) will be dominated.

Consider the correlation between costs and transit times: they differ just by an constant additive, namely the starting time. Thus, for two labels that are expanded via the same time-window the distance value controls the dominance relation. That means, that in this case a label dominates another if and only if it has a lower distance value.

Therefore, the number of labels on an arc a is bounded by the number of time-windows on all ingoing arcs. Thus, the number of iterations in each loop is either bounded by the number of arcs or bounded by the number of time-windows. Hence, the algorithm terminates in polynomial-time with a shortest path (or the notification that there is no path). □

We can directly conclude for the SPPTW:

Corollar 3 *The SPPTW with transit time (including waiting times) as cost function can be solved in polynomial-time.*

For additional acceleration of the algorithm we use goal-oriented search [6].

3.3 Readjustment of the time-windows

For each arc a of a computed route we consider all geographic dependent arcs in *confl*(a) during the time interval $[t_1, t_2]$ representing the transit over a. Then we verify for each time-window on an arc of *confl(a)*, whether the time-window and $[t_1, t_2]$ overlap. Depending on that, the time-window is shortened, erased or left unchanged.

4 Computational Results

Two questions certainly arise concerning the presented conflict-free approach. Is the approach really better than the static one? Is the algorithm suitable for real-time computation? Concerning these questions we present some results for test instances (scenarios) on a graph with about 30.000 arcs.

4.1 Comparison with a static approach

In order to measure the performance of the computed routes we consider the sum of all transit times (all requests). We compare these overall transit times with a static approach used in Container Terminal Altenwerder (CTA) at Hamburg Harbour.

The comparison shows that the conflict-free approach is superior to the static one. The strongest dominance of our algorithm is achieved for scenarios with heavy traffic (many AGVs) which clearly indicates the potential of the conflict-free approach.

4.2 Computation times

Besides obtaining a better solution in comparison with other approaches, a real-time computation requires fast answers. The computation times[2] in Table 1 show that our algorithm is able to provide this. The table is divided into computation times for determining a shortest path with time-windows (search) and computation times for readjusting the time-windows (blocking).

On average, the computation (in both cases) take not more than some hundredth of a second. The entire real-time computation (search and blocking together) in all tested cases takes less than half a second which is suitable for this real-time computation.

[2] Hardware: AMD-Athlon 2100+ (1,7 Mhz) with 512 MB RAM.

Table 1. Computational times (in sec.)

Scenarios	Search		Blocking	
	maximal	⌀	maximal	⌀
1A (15 AGVs)	0.21	0.018	0.19	0.026
2A (15 AGVs)	0.14	0.020	0.17	0.038
3A (20 AGVs)	0.22	0.021	0.20	0.030
4A (44 AGVs)	0.21	0.019	0.19	0.022
1B (32 AGVs)	0.18	0.022	0.18	0.024
2B (38 AGVs)	0.28	0.021	0.17	0.025
1C (22 AGVs)	0.09	0.021	0.13	0.042
2C (44 AGVs)	0.17	0.023	0.09	0.029
3C (48 AGVs)	0.17	0.022	0.12	0.040

References

1. Aspnes, J. et al. (1997) On-line routing of virtual circuits with applications to load balancing and machine scheduling. Journal of the ACM 44, 486–504
2. Beasley, J. E., Christofides, N. (1989) An algorithm for the resource constrained shortest path problem. Networks 19, 379–394
3. Broadbent, A. J. et al. (1987) Free-ranging agv and scheduling system. In Automated Guided Vehicle Systems, 301–309
4. Cormen, T. H., Leiserson, Ch. E., Rivest, R. L. (1990) Introduction to Algorithms. MIT Press, Cambridge, Massachusetts
5. Desrosiers et al. (1986) Methods for routing with time windows. European Journal of Operations Research 23, 236–245
6. Hart, P., Nilsson, N., Raphael, B. (1968) A formal basis for the heuristic determination of minimum cost paths. In IEEE Transactions on Systems, Science and Cybernetics SCC-4, 100–107
7. Huang, J., Palekar, U. S., Kapoor, S. (1993) A labeling algorithm for the navigation of automated guided vehicles. Journal of engineering for industry 115, 315–321
8. Kim, Ch. W., Tanchoco, J. M. A. (1991) Conflict-free shortest-time bidirectional agv routeing. International Journal of Production Research 29(12), 2377–2391
9. Krishnamurthy, N., Batta, R., Karwan, M. (1993) Developing conflict-free routes for automated guided vehicles. Operations Research 41, 1077–1090
10. Sancho, N. G. F. (1994) Shortest path problems with time windows on nodes and arcs. Journal of mathematical analysis and applications 186, 643–648

Dynamical Configuration of Transparent Optical Telecommunication Networks*

Andreas Tuchscherer[1]

[1] Konrad-Zuse-Zentrum für Informationstechnik Berlin, Department Optimization, Takustr. 7, D-14195 Berlin-Dahlem, Germany. Email: tuchscherer@zib.de. Supported by the DFG research center "Mathematics for key technologies" in Berlin

Abstract. *All-optical* telecommunication networks allow for switching connections by *lightpaths* which can pass several network links without any opto-electronic conversion. Upon arrival of a connection request, it must be decided online, i.e., without knowledge of future requests, if it is accepted and in that case on which lightpaths the connection is routed. This *online problem* with the goal of maximizing the total profit gained by accepted requests is called *Dynamic Singleclass Call Admission Problem* (DSCA). We present existing and new algorithms for DSCA as well as their theoretical and practical evaluation.

1 Introduction

The links of an *all-optical* telecommunication network consist of *optical fibers* by which data is transmitted using optical signals. With *optical switches* in the network nodes, an optical signal may be transmitted along several fibers without leaving the optical layer, i.e., no opto-electronic conversion is performed at intermediate nodes. Such an optical channel is called a *lightpath.* In the absence of optical conversion devices, each lightpath uses just one wavelength since the optical switches maintain the wavelength of a passing signal. Applying the technique of *wavelength division multiplexing* (WDM), it is possible to transmit several optical signals of different wavelengths along one fiber at the same time. Hence, multiple lightpaths may traverse one fiber simultaneously if they have different wavelengths. However, any two lightpaths which share one fiber must use different wavelengths. This condition is called *wavelength conflict constraint.*

One crucial aspect in today's telecommunication service is the following. In many real-world applications like telephony and video conferencing demands change in a highly dynamic way. This requirement can be realized in all-optical networks using novel optical switches that are software-controlled and reconfigurable within milliseconds, thus, data connections can to be set up and taken down on demand. Moreover, the requested demands usually become known only shortly before they are actually required, which results

* This paper gives a brief summary of [9]. The accomplished research is based on a joint project with T-Systems Nova GmbH, financed by the DFN.

in substantial uncertainty for planning. We refer to them as *connection requests* or *calls*. Depending on the provided capacities of the optical network, the customers may suffer service blocking, especially in situations of demand peaks. Hence, for each newly arriving connection request, the network operator has to decide whether he accepts or rejects the call. In doing so it is also admissible to reject a call even if a corresponding connection could still be established. For accepted calls fixed lightpaths must be provided for the total holding time of the connection. Each accepted call yields a specific profit. We call this problem with the goal of maximizing the total profit gained by accepted connection requests *Dynamic Singleclass Call Admission Problem*, or DSCA for short. Note that the decision on each request must be made irrevocably and without knowledge of future calls. Therefore, DSCA is counted among the so-called *online optimization problems* (see [3] for details on online optimization).

In this paper we are concerned with all-optical networks without wavelength conversion capabilities and assume that there is no crucial limit in the switching capability of the optical switches in the network. As a consequence, the only bottleneck for realizing lightpaths results from the fiber capacities, i.e., the numbers of provided wavelengths. From now on, we will refer to all-optical networks simply as *optical networks*.

1.1 Related Work

In previous work for DSCA, the following simplifying assumptions are usually made on the problem parameters: Each network link consists only of one optical fiber, each fiber is equipped with the same set of wavelengths, and each call requires only one lightpath and yields the same profit, i.e., the goal of maximizing the total profit is equivalent to minimize the *blocking probability* for a connection request.

Also motivated by the need for distributed implementations, particularly simple algorithms for DSCA have previously been proposed. These schemes restrict for each pair of end nodes the set of allowed routes for connecting lightpaths to a few predefined paths. The actually used path is chosen according to availability and some priority order. For the selection of the wavelength, first approaches used fixed and random wavelength search orders [4,2], i.e., the first possible wavelength in the given order is chosen. In [1], Bala, Stern, Simchi, and Bala propose adaptive search orders that incorporate network state information, namely the current utilizations of different wavelengths.

As a counterpart, Mokhtar and Azizoglu present five algorithms, also called *greedy algorithms*, where any lightpath can potentially be chosen. In particular, these algorithms always accept a connection request if possible. Greedy algorithms are based on shortest path computation and different strategies for selecting the wavelength. Experimental results exposed the superiority of this approach over the algorithms above using restricted path selection w.r.t. the blocking probability in realistic settings where only a little fraction of calls must be rejected. Moreover, it turned out that taking into account network state information like the wavelength utilization can improve the behavior of algorithms.

In joint work with Hülsermann, Jäger, Krumke, Poensgen, and Rambau, we proposed adapted versions of the greedy algorithms, including the specification of tie-breaking rules [6]. In our model, still one fiber is installed per link but the wavelengths provided by the fibers may differ. Using extensive experimental studies that are based on a well-founded traffic model, which is also applied in this paper, we showed that the choice of breaking ties can affect the blocking probability of a greedy algorithm significantly.

To the best of our knowledge, no rejection criteria for the call admission part have yet been proposed, except for the restriction to few admissible lightpaths due to constraint routing.

1.2 Outline

The following section is devoted to the mathematical problem definition of DSCA. In Section 3, we present the algorithms considered in this paper. Then the main results of their evaluation, concerning theory and simulation, are given in Section 4. Finally, we make some concluding remarks in Section 5.

2 Problem Definition

The foundation in the problem definition is the optical network.

Definition 1 *An* optical network *is a triple (G, Λ, W), where*

- *$G = (V, E)$ is a simple and undirected graph,*
- *$\Lambda = \{\lambda_1, \dots, \lambda_k\}$ is a set of wavelengths, and*
- *$W : E \to 2^\Lambda$ is a map from E to the power set of Λ, where $W(e)$ is the set of wavelengths generally available on edge e.*

A lightpath *in the optical network (G, Λ, W) is a pair (p, λ) which consists of a path p in G together with a wavelength $\lambda \in \Lambda$ such that $\lambda \in W(e)$ for each edge $e \in E(p)$.*

The wavelength conflict constraint may now be formulated as follows.

Definition 2 (Wavelength conflict constraint) *For each two simultaneously routed lightpaths (p_1, λ_1) and (p_2, λ_2) in an optical network (G, Λ, W), we have:*

$$E(p_1) \cap E(p_2) = \emptyset \text{ or } \lambda_1 \neq \lambda_2.$$

A lightpath (p, λ) is called free *if it can be realized without violating the wavelength conflict constraint.*

Having defined an optical network and the wavelength conflict constraint, we can now state the considered problem.

Definition 3 (Dynamic Singleclass Call Admission Problem) *An instance of the* Dynamic Singleclass Call Admission Problem *(*DSCA*) is given by an optical network* (G, Λ, W)*, a time horizon* T*, and a sequence of connection requests* $(\sigma_1, \sigma_2, \ldots)$ *with* $\sigma_j = (u_j, v_j, b_j, t_j, d_j, p_j)$*, where*

$u_j, v_j \in V$	*are the end nodes,*
$b_j \in \mathbb{N}$	*is the number of required lightpaths,*
$t_j \in [0, T]$	*is the start time,*
$d_j \in \mathbb{R}_+$	*is the duration,*
$p_j \in \mathbb{R}_+$	*is the profit.*

The task is to maximize the total profit gained such that valid answers are given to all connection requests. The answer for each σ_j *must be given without knowledge of calls with later start times and specifies whether the request is accepted or rejected. If* σ_j *is accepted, it contributes* p_j *to the total profit but requires* b_j *lightpaths connecting* u_j *and* v_j *in* (G, Λ, W) *from* t_j *until* $t_j + d_j$*. In doing so, the wavelength conflict constraint must be satisfied all the time.*

3 Algorithms

In the following, we present two different classes of online algorithms for DSCA. The *greedy algorithms* have originally been proposed by Mokhtar and Azizoglu in [7]. We distinguish between two variants: partial wavelength search (PWS) and total wavelength search (TWS). Let u and v denote the end nodes of the requested connection.

PWS: Let $\lambda_{i_1}, \ldots, \lambda_{i_k}$ be some order on the set of wavelengths. If there is a free $[u, v]$-lightpath, route a shortest one in wavelength λ, where λ is the first wavelength in the order providing any free $[u, v]$-lightpath.

TWS: Let $\lambda_{i_1}, \ldots, \lambda_{i_k}$ be some order on the set of wavelengths. If there is a free $[u, v]$-lightpath, route a shortest one in wavelength λ, where λ is the first wavelength in the order providing a *globally shortest* free $[u, v]$-lightpath.

Sorting the wavelengths in order of decreasing current availability (number of edges where the wavelength can currently be used) turned out to yield the best versions in partial and total wavelength search [8,9]. We denote the corresponding algorithms by PACK(P) and PACK(T).

On the other hand, so-called *network fitness algorithms* have been developed at Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB) in a joint project with T-Systems Nova GmbH (cf. [8,9]). The idea is to route a given connection request, if accepted, in such a way that the resulting network state allows for a maximum potential profit of future calls. Since we do not know the calls to come, we can only relate potential future profit at any point in time to the current network state (a network state corresponds to a configuration of routed lightpaths). Hence, we would like to assign a *fitness* level to each network state that reflects the network's capability to accept further calls. Obviously, it is desirable to maintain the state of the network as *fit* as possible. To this end, we route an accepted call in such a way that the remaining network fitness after realizing the lightpath is maximized. After expiration of a routed connection, the fitness value increases again. The crucial task is to define a proper measure for the network fitness.

FIT: Let $fit : \mathcal{S} \to \mathbb{R}_+$ be some network fitness function, where $\mathcal{S}$ denotes the set of all possible network states of (G, Λ, W). If there is a free $[u, v]$-lightpath, route such a lightpath (p, λ) that the resulting state $S + (p, \lambda)$ yields a maximum fitness value.

Note that a network fitness algorithm must consider all currently free $[u, v]$-lightpaths for each routing decision. Since their number is in general exponential in the size of the optical network, we cannot hope for an efficient implementation. However, as simulations show, a reasonable fitness algorithm always select short lightpaths. Hence, it suffices to consider only the l shortest $[u, v]$-lightpaths for some $l \in \mathbb{N}$, which can be computed in polynomial time.

We consider two network fitness algorithms called *available-lightpaths-reduction* (ALR) and *single-flow-reduction* (SFR). While ALR defines the fitness as the total number of currently free lightpaths in the optical network, the algorithm SFR defines the fitness as the sum over all pairs of distinct nodes s and t and each wavelength λ of the maximum number of free *edge-disjoint* $[s, t]$-lightpaths in wavelength λ.

4 Results

Concerning the theoretical evaluation of online algorithms the standard tool is *competitive analysis* (cf. [3]). A deterministic online algorithm ALG is called *c-competitive* if for each sequence of requests σ its profit ALG(σ) is at least $1/c$ times the profit of an optimal offline algorithm OPT that knows the sequence in advance, i.e., $\text{ALG}(\sigma) \geq 1/c \cdot \text{OPT}(\sigma)$. The *competitive ratio* of ALG is defined to be the infimum of all $c \geq 1$ such that ALG is c-competitive.

It is easily shown that for a special version of DSCA all online algorithms have the same competitive ratio.

Theorem 1 ([9]) *For the problem* DSCA *with $d_j = \infty$ and $p_j = b_j$ for each request σ_j, the competitive ratio of each deterministic competitive algorithm is km, where k denotes the number of wavelengths and m denotes the number of edges in the optical network.*

Proof. Let ALG be an arbitrary deterministic algorithm for DSCA. If ALG rejects the first given call, it is not competitive: For a sequence $\sigma = (\sigma_1)$ consisting of one request, ALG makes zero profit, while $\text{OPT}(\sigma) = b_1 > 0$.

If ALG accepts the first call of a request sequence σ, we have $\text{ALG}(\sigma) \geq 1$. Moreover, OPT can at most gain a profit of one for each edge and wavelength, yielding km in total. This implies that ALG is km-competitive.

It is shown in [5] that no deterministic algorithm can be better. Denote each call by (u_j, v_j, b_j) specifying the end nodes u_j, v_j for the connection and the number b_j of required lightpaths. Consider the line graph with nodes denoted by $v_1, \ldots, v_n$ from left to right where all k wavelengths are available on all m edges and the following request sequence:

$$\sigma_1 = (v_1, v_n, 1), \sigma_2 = (v_1, v_2, k), \sigma_3 = (v_2, v_3, k), \ldots, \sigma_n = (v_{n-1}, v_n, k).$$

Since ALG accepts σ_1, it must reject all subsequent calls, achieving a total profit of $\text{ALG}(\sigma) = 1$. In contrast, OPT accepts the calls $\sigma_2, \ldots, \sigma_n$ at a total profit of km. □

Motivated by the incapability of competitive analysis to evaluate algorithms for DSCA, the remaining part of this section deals with the experimental evaluation of the presented algorithms.

The simulation studies have been carried out for a 14-nodes network with the dimensioning depicted in Fig. 1. The dimensioning is based on the static traffic demands shown in Table 1 which were generated w.r.t. US-American population data (see [6]). It is worth to mention that this dimensioning is capable of satisfying all static demands simultaneously.

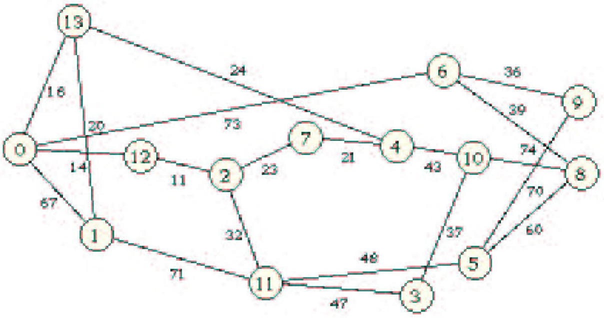


Fig. 1. The 14-nodes network with dimensioning (on an edge with label n the wavelengths $\lambda_1, \lambda_2, \ldots, \lambda_n$ are generally available)

In the experiments we make the assumptions that each accepted call yields a profit of 1 and the holding time of each connection is randomly chosen according to an exponential distribution with mean 1. Hence, each connection

Table 1. Demand matrix for the 14-nodes network

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	13	-	-	-	-	-	-	-	-	-	-	-	-	-
3	2	3	-	-	-	-	-	-	-	-	-	-	-	-
4	4	6	1	-	-	-	-	-	-	-	-	-	-	-
5	6	9	2	4	-	-	-	-	-	-	-	-	-	-
6	3	4	1	2	3	-	-	-	-	-	-	-	-	-
7	5	7	2	3	6	2	-	-	-	-	-	-	-	-
8	1	1	0	1	1	0	1	-	-	-	-	-	-	-
9	4	6	1	3	4	3	4	1	-	-	-	-	-	-
10	9	13	2	6	10	5	8	1	14	-	-	-	-	-
11	6	9	2	4	6	4	5	1	13	16	-	-	-	-
12	11	16	4	7	11	4	8	2	7	15	10	-	-	-
13	1	2	0	1	1	0	1	0	1	2	1	2	-	-
14	3	5	1	2	3	1	2	0	2	4	3	5	1	-

request σ_j is of the form:

$$\sigma_j = (u_j, v_j, t_j, d_j).$$

Denote by $d(s,t)$ the amount of static demand between nodes s and t, and let $m \in \mathbb{N}$. Calls for connections between s and t are generated by $m \cdot d(s,t)$ sources according to a *Poisson arrival process* with intensity 1/12, i.e., the inter arrival times between two requests are exponentially distributed with mean 12. The parameter m is called *multiplex factor*, and it controls the offered traffic load. The relation between call durations and the call inter arrival times reflect the observation that a customer who has a permanent connection uses it only for about 1/12 of the time. We compare the blocking probabilities of the considered algorithms on request sequences which are generated for multiplex factors from 1 to 12. Since blocking probabilities greater than 5% are considered unacceptable the graphics are plotted within a logarithmic scale. The results are depicted in Fig. 2. It turns out that the total wavelength search version PACK(T) is superior to the partial wavelength search version PACK(P) and produces solutions with about the same quality as ALR. But the network fitness algorithm SFR performs best. Similar results are obtained in the case when the duration of each call equals 1.

5 Conclusions

We have shown that from the competitive analysis perspective all online algorithms are equally bad for a special version of DSCA. Furthermore, experimental results showed that the algorithm SFR performs best, corresponding to intuition. Concerning the incorporation of rejection criteria, we believe that rejecting calls can only be advantageous if either the call profits vary significantly or network load is high, which is not the case in real-world applications.

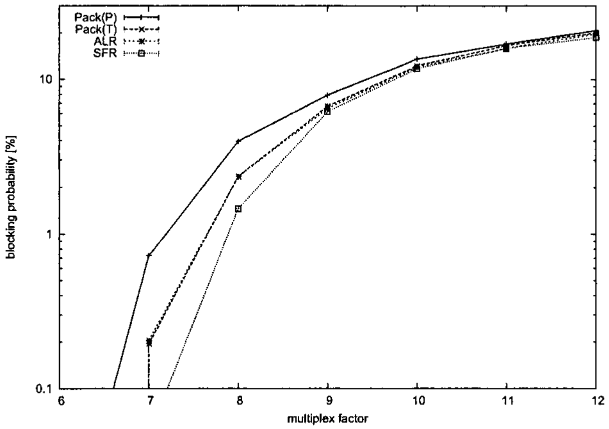


Fig. 2. Results of selected algorithms in a 14-nodes network

References

1. Bala K., Stern, T., Simchi K., Bala K. (1995) Routing in a Linear Lightwave Networks. IEEE/ACM Transactions on Networking **3**, 459–469
2. Birman A., Kershenbaum, A. (1995) Routing and Wavelength Assignment Methods in Single-hop All-optical Networks with Blocking. Proceedings of the INFOCOM '95, 431–438
3. Borodin A., El-Yaniv R. (1998) Online Computation and Competitive Analysis. Cambridge University Press
4. Chlamtac I., Ganz A., Karmi G. (1992) Lightpath Communications: An Approach to High Bandwidth Optical WAN's. IEEE Transactions on Communications **40**, 1171–1182
5. Krumke S. O., Poensgen D. (2002) Online Call Admission in Optical Networks with Larger Demands. Proceedings of the 28th International Workshop on Graph-Theoretic Concepts in Computer Science **2573**, Springer, 333–344
6. Hülsermann R., Jäger M., Krumke S. O., Poensgen D., Rambau J., Tuchscherer A. (2003) Dynamic Routing Algorithms in Transparent Optical Networks. Proceedings of the 7th IFIP Working Conference Optical Network Design & Modelling, Kluwer Academic Press, 293–312
7. Mokhtar A., Azizoglu M. (1998) Adaptive Wavelength Routing in All-optical Networks. IEEE/ACM Transactions on Networking **6**, no. 2, 197–206
8. Poensgen D. (2003) Facets of Online Optimization. Dissertation, Technische Universität Berlin, Cuvillier Verlag Göttingen
9. Tuchscherer A. (2003) Dynamical Configuration of Transparent Optical Telecommunication Networks. Diploma Thesis, Technische Universität Berlin
10. Zhang Z., Acampora M. (1995) A Heuristic Wavelength Assignment Algorithm for Multi-hop WDM Networks with Wavelength Routing and Wavelength Reuse. IEEE/ACM Transactions on Networking **3**, 281–288

The value of information in a container collection system for end-of-life vehicles

Ieke le Blanc [1], Rene Schreurs [2], Hein Fleuren [1], Harold Krikke [1]

[1] CentER Applied Research, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands
[2] Auto Recycling Nederland, P.O. Box 12252, 1100 AG Amsterdam Zuidoost, The Netherlands

Abstract In this paper we discuss the value of accurate inventory information for distribution planning purposes in a reverse logistics system. Three levels of information are introduced and the value is assessed in a simulation loop. For every planning period the operational vehicle routing problem is solved to optimality by route enumeration and set partitioning. Extensive sensitivity analysis is performed and discussed to analyze the factors that influence the value of accurate inventory information for reverse distribution planning. Network density appears to be one of the main determinants.

1. Introduction

1.1 Reverse logistics and uncertainty

Reverse logistics is the management of good flows in the opposite direction of the traditional supply chain, with the purpose of value recovery or proper disposal (Rogers and Tibben-Lembke 1998). Products can be returned at several stages of their life cycle as so-called commercial, repairable, end-of-use or end-of-life returns. The complexity of reverse logistics management is largely caused by uncertainty (Fleischmann et al. 2000). Uncertainty in the behavior of the system is caused by the lack of information and control mechanisms regarding quantity, timing, product compositions and quality of returns. Uncertainty and the type of return are among the major determinants of the right reverse supply chain (Krikke et al. 2004). Within the right logistic network design, the application of modern sensor technology can help in reducing and managing the uncertainty. Examples in reverse logistics are the application of so-called "data-logs" in power-tools (Klausner and Hendrickson 2000) or sensors embedded in cars and car tires (Van Nunen and Zuidwijk 2004). These applications reduce the complexity in foreseeing when the return occurs, the handling of the returns or the selection of the appropriate recovery option. Here we focus on returns of end-of-life products, where material recycling is the appropriate disposition option. Recovery processes in these networks are typically centralized, resulting in relative high transportation cost for these low valued recyclables.

In forward logistics, inventory management and control is common practice. Real-time stock information has become the standard. In reverse logistics, the stocks of cores or recyclables are often disregarded and no registration exists at all. If a certain amount of recyclables is available at a collection point, transfer to a centralized facility takes place. The lack of information about the stock size at the collection sites makes it impossible to plan collection trips ahead. Ultimately, inventory and routing decisions should be taken simultaneously (Herer and Levy 1997). A research on the collection of liquids coming from end-of-life vehicles showed opportunities for significant cost savings in a network with low density (Le Blanc et al. 2004). In this research, based on a real-life case study in the Netherlands, we focus on the value of inventory information for the collection planning of containers for various network densities.

1.2 Background case study

The study presented in this paper is performed for Auto Recycling Nederland (ARN). ARN is an organization coordinating the recycling of end-of-life vehicles (ELVs) in the Netherlands on behalf of the automotive industry. Existing ELV-dismantlers, logistic service providers (LSPs) and recyclers carry out the work for ARN on a commercial basis, recycling over 275,000 wrecks yearly. High volume materials dismantled from ELVs such as PU-foam, rubber, glass, tires and bumpers are stored and collected in containers. When a container is full, the ELV dismantler places a request at the LSP to collect the container. Within 5 working days, the LSP visits the dismantlers and exchanges the full container for an empty one. The containers are brought to a depot where some processing takes place. Afterwards, they are sent to the recyclers in consolidated batches. Figure 1 gives an overview of the system. Yearly, approximately 5,000 containers are collected in this manner in the ARN system. The collection of containers concerns about 70% of the total collection cost incurred by ARN. Research has shown that, in the earlier mentioned collection system for oils and fuels, the cost of collection can be reduced significantly by knowledge of accurate stock levels at the ELV-dismantlers. The goal of this research is to examine the possibilities to extend these results to the collection system of containers.

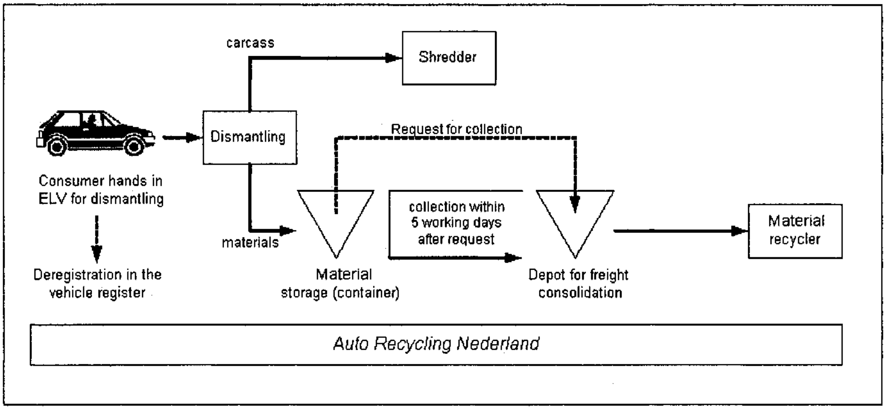

carcass
Shredder
Dismantling
Request for collection
Consumer hands in ELV for dismantling
collection within 5 working days after request
Material recycler
materials
Deregistration in the vehicle register
Material storage (container)
Depot for freight consolidation
Auto Recycling Nederland

Fig. 1. Overview of the logistic chain of ARN

2. Three degrees of information

The better and more accurate the data on the inventory levels, the higher the costs of obtaining these data. We are interested in the tradeoff between the level of information and the costs in the system or in other words: what is the value of information? To this end we compare three degrees of information:

- Degree 1: there is no information on the inventory levels. A container is collected only if the ELV-dismantler indicates the container is full. This scenario corresponds to the current situation.
- Degree 2: inventory levels are forecasted and the forecasts, together with the collection requests of the ELV-dismantlers are used to construct the collection planning.
- Degree 3: inventory levels are known accurately at any time. The logistic service provider can plan the collection of containers proactively. To obtain this level of information, expensive sensors are needed to determine the filling of the containers at the ELV-dismantlers.

Schreurs (2004) investigates and tests several types of forecasting models for the container collection system of ARN. Schreurs suggests an intuitive model based on the ratio between the time passed since the last collection and average time between two collections and ratio between the number of deregistrations and the average number of deregistrations. ARN has real-time insight in the number of deregistrations in the national vehicle register for each ELV-dismantler. The model of Schreurs performs well and even outperforms more sophisticated forecasting models. We consider this easy to implement model of the estimates on the inventory positions in the degree 2 scenarios.

3. Simulation of the distribution planning

3.1 Simulation

In order to estimate the value of information, a simulation model is used. The number of deregistrations is simulated using the empirical distribution for each ELV-dismantler. Furthermore, the amount of materials in the containers is estimated, based on the number of deregistrations. Since the ELV-dismantlers won't put the materials in the containers one-by-one, it is assumed that materials of the ELVs end up in a container after 20 deregistrations. When the amount of materials in a container exceeds a certain limit, the container must be collected and an order that must be collected ("MUST-order") is generated. If a container is not full yet, but almost full according to the forecasting model or accurate information, a "CAN-order" is generated. These CAN-orders can be collected but don't necessarily need to. Since orders are created a vehicle routing problem needs to be solved for each period, we have to express the future benefits for collecting the CAN-order in the current period. The future benefits that we use are:

$$FB_{co} = \alpha * LC_{co} * L_{co} \qquad (1)$$

where:

FB_{co}	Future benefits of collecting CAN-order co in this period
α	Correctionfactor between ½ and 1
LC_{co}	Linehaulcost
L_{co}	Loadfactor of the container

The correction factor a expresses the combination possibilities for must-orders. If $\alpha = 1$ no combinations are made and the full linehaul cost are charged to collect a single container. If $\alpha = ½$ perfect combination possibilities exist, because half of the vehicle capacity is used, and therefore half of the linehaul cost are charged.

The simulation is performed with 10 replications. Each replication runs over 11 years, 10 years to calculate the costs of the scenario's plus an additional first year to exclude start-up effects.

3.2 Vehicle routing

For pure comparison we want to exclude the randomness of a vehicle routing heuristic and compare "optimal" solutions with each other. These solutions are obtained by enumeration of all possibilities and selecting the optimal combinations by solving a set partitioning problem (Fleuren 1988). ARN hires four LSPs in the Netherlands for four separate areas and for each LSP all possible routes including at least one MUST-order are generated. The orders included in the routes are only the orders in their respective collection-areas. The small number of orders that fit in a truck makes it easy to enumerate all possibilities (Laporte et al. 2000). After that, the set-partitioning problem is solved for each LSP. To solve the set partitioning instances, we use a solver developed by Van Krieken et al. (2004). The main building blocks of the solver are Lagrangean relaxation for determining the lower bounds and branch and bound for finding the optimal solution. Furthermore, several problem reduction techniques are used to reduce the number of variables and constraints in the problem. Calculation times for a simulation run 10 replications of 10 years are within two hours on a normal desktop PC.

Indices

$mo \in MO$	Must-orders
$co \in CO$	Can-orders
$r \in R$	Routes

Parameters

C_r	Cost of route r
FB_r	Future benefits of route r (future benefits of the can-order in the route)
$Y_{mo,r}$	1 if MUST-order mo is in route r, 0 else
$Y_{co,r}$	1 if CAN-order co is in route r, 0 else

Variables

X_r	1 if route r is executed, 0 else
S_{co}	1 if CAN-order co is not executed, 0 else

Routing problem

$$\text{Min} \quad \sum_r (C_r - FB_r) * X_r \tag{2}$$

$$\text{S.t.} \quad \sum_r Y_{mo,r} * X_r = 1 \qquad \forall mo \tag{3}$$

$$\sum_r Y_{co,r} * X_r + S_{co} = 1 \qquad \forall co \tag{4}$$

$$X_r \in \{0,1\} \qquad \forall r \tag{5}$$

$$S_{co} \in \{0,1\} \qquad \forall co \qquad (6)$$

Eq. (2) minimizes the costs of the routes that are executed. The comparison takes place on real cost corrected for future benefits. Eqs. (3) and (4) make sure that each MUST-order is executed and each CAN-order is executed at most once.

The data used is gathered together with logistic experts of ARN. Extensive tests on the model and datasets are performed, validating and verifying its correctness.

4. Results

4.1 Base scenarios

The results of the ARN case study for the base scenarios are given in Table 1. We see that some savings can be made. If we accurately know the inventory levels we can save 3.1% on the total costs. More than 12% of the containers are collected before the ELV-dismantler requested the collection. When we have to use the forecasts these values drop.

	Degree 1: No information	Degree 2: Forecasting model	Degree 3: Accurate information
Savings as percentage of total costs	0.0 %	2.0 %	3.1 %
Savings in distance (km per year)	0	13,600	19,000
Savings in distance as percentage of total distance	0.0 %	3.2 %	4.5 %
Percentage of CAN-orders	0.0 %	9.0 %	12.3 %

Table 1. Results of the base scenarios

4.2 The influence of changing the network density

The savings in the ARN case are limited to a few percent, however Le Blanc et al. (2004) reports considerable savings in a comparable system with less denser network density. In order to analyze the influence of the network density to the results, we now change the density. This is done by assuming that a lower volume of materials is collected; e.g. a network density of 60% means that the amount of materials collected from each vehicle drops with 40%. In Figure 2 we can see what this means for the savings that can be made.

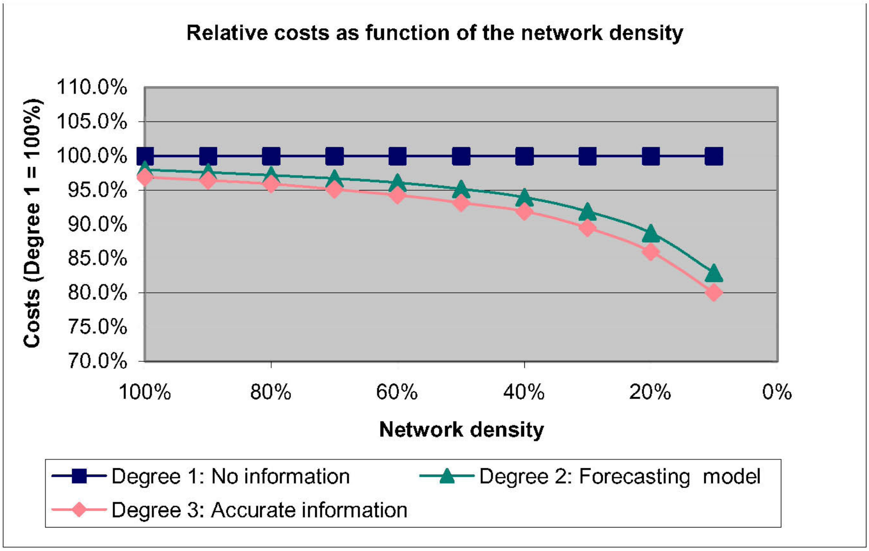


Fig. 2. Costs of each information level compared for different network densities (the costs of degree 1 always equals 100%)

We can see clearly that the lower the network density, the higher the relative value of information, due to the fewer combination possibilities with MUST-orders. The number of vehicle registrations in the forecasting model is a good predictor for the inventory levels. Therefore, with the forecasting model instead of accurate information the performance is marginally worse, such that the investment in expensive sensor technology to obtain precise information is not justified.

5. Discussion and conclusion

In a real-life case study we have analyzed the effects of the value of inventory data in collection planning for various network densities. In dense networks, the transportation costs are hardly influenced by the irregularity in the occurrence of transportation orders, since the number of possible combinations to build a route is large. Note that accurate inventory information can still be valuable in the matching of supply (returns) with demand. When the network density decreases, fewer combinations can be made and the value of the information increases.

Network density is related to the average capacity occupation of orders in comparison to the truck. When orders are smaller compared to the vehicle capacity, more transportation orders are required to consider a network dense. In the ARN case discussed here, a container occupies half of a trucks capacity, such that the network density is high.

In further research we will elaborate this topic further with analysis on the influence of consolidation decisions, the influence of other order sizes and vehicle capacities and the efficiency gain by planning on a central level.

Acknowledgement

The authors would like to thank Roelof Reinsma and Annemieke van Burik of Auto Recycling Nederland for their assistance and support during the project.

References

Fleischmann M., Krikke HR, Dekker R, Flapper SDP (2000) A characterisation of logistics networks for product recovery. Omega 28: 653-666

Fleuren HA (1988) A computational study of the set partitioning approach for vehicle routing and scheduling problems. Ph.D. thesis, University of Twente

Herer Y, Levy R (1997) The metered inventory routing problem, an integrative heuristic algorithm. International Journal of Production Economics 51: 69-81

Klausner M, Hendrickson CT (2000) Reverse-logistics strategy for product take-back.. Interfaces 30(3): 156-165

Krikke H.R., Le Blanc HM, Van de Velde SL (2004) Product modularity and the design of closed-loop supply chains. California Management Review 46(2): 23-39

Laporte G, Gendreau M, Potvin JY, Semet F (2000) Classical and modern heuristics for vehicle routing problem. International Transactions in Operational Research 7: 285-300

Le Blanc HM, Van Krieken MGC, Fleuren HA, Krikke HR (2004) Collector Managed Inventory, a proactive planning approach to the collection of liquids coming from end-of-life vehicles. CentER Discussion Paper 22-2004

Rogers DS, Tibben-Lembke RS (1998) Going backwards: reverse logistics trends and practices. Reverse Logistics Executive Council

Schreurs RP (2004) Voorspelbaarheid van de inzameling van materialen afkomstig van autowrakken en de impact op de distributieplanning (in Dutch). Master thesis, Tilburg University

Van Krieken MGC, Fleuren HA, Peeters MJP (2003) A lagrangean relaxation based algorithm for solving set partitioning problems. CentER Discussion Paper 44-2004

Van Nunen JAEE, Zuidwijk RA (2004) E-enabled closed-loop supply chains. California Management Review 46(2): 40-54

Approximate Policies for Hybrid Production and Rework Systems with Stochastic Demand and Yield

Christian Gotzel, Karl Inderfurth

Otto von Guericke University of Magdeburg, Faculty of Economics and Management, P.O. Box 4120, 39106 Magdeburg, Germany

Abstract. We consider a production inventory problem under periodic review with stochastic demand and stochastically proportional yield. Defective products (i.e. the yield loss) arising from the unreliable production process can be temporarily stored in an inventory and passed through a rework process. Rework is completely reliable and brings defective products up to the same quality level as new ones, so that they can be used for demand fulfilment. We focus on the application of linear heuristic policies similar to those emerging from MRP application, and develop simple expressions for the computation of policy parameters. Results of a numerical study indicate that the heuristic approach performs quite well.

1 Introduction

The production-and-rework problem is common to several industrial production processes where a production reject arises as a result of deficiencies in the process or due to inconsistent raw material quality. Whereas a fraction of the production output can be directly used to fulfil the demand, the remaining fraction is passed through a rework process to receive products offering the desired quality level.

We consider a production inventory problem with stochastic yield and demand. Production yield is a random fraction of the production order. This yield model is usually referred to as stochastically proportional yield. Defective products, i.e., the yield loss, are stored in an inventory to be available for future rework. There is no disposal option. Yield and demand distributions are assumed to be stationary and independent. Procurement, rework, holding and shortage costs are strictly proportional. Any unsatisfied demands are backlogged. Production and rework leadtimes are deterministic but may differ.

Production-inventory problems with stochastic yield have been addressed by several authors. In [2] and [3] it is shown that the optimal control policy is rather complex even without a rework option. Applying linear heuristics for this problem as proposed in [1] and [3] has proven to give a good approximation. However, there is only few research directed to production systems with rework, mostly considering lot-sizing problems without regarding uncertainty, see [6] for an overview. An adjusted MRP approach for a hybrid production and remanufacturing planning is presented in [4].

In the next section of our paper we formulate a simple control policy for the problem under general leadtime conditions. In a subsequent section, we derive expressions for the determination of policy parameters and present results of a numerical study. We end with some final conclusions.

2 Control Policy

In the sequel we will consider a production and rework facility as described above. Events and decisions are timed in the following manner. At the beginning of period t, inventory levels are observed and the decision on the production quantity p_t is made. In case of a zero production leadtime λ_P, the yield fraction $z_t \cdot p_t$ becomes immediately available at the serviceables inventory, otherwise it becomes available at the end of period $t+\lambda_P$-1. At the same time, the yield loss $(1-z_t)\cdot p_t$ enters the recoverables inventory. Then the decision on the rework quantity is made based on all information available at this time. Zero leadtime rework orders increase the serviceables inventory immediately whereas for rework leadtimes λ_R>0 the reworked items become available at the end of period $t+\lambda_R$-1. After this decision, the demand of period t is realised and fulfilled from the inventory.

We use the following notation:

h_S : serviceables holding cost per unit
h_R : recoverables holding cost per unit
v : shortage cost per unit
d : stochastic demand with mean μ_d and variance σ_d^2
z : stochastic yield fraction with mean μ_z and variance σ_z^2
x_S : initial serviceables net inventory (on hand minus backorder)
x_R : initial recoverables inventory
p : production quantity
r : rework quantity

Production and rework quantities are determined using a two-parameter (S,M)-policy with produce-up-to level S and rework-up-to level M. This policy structure has proven a good performance in a hybrid production-remanufacturing system with external returns, see [5]. The production decision is based on a comparison of an appropriately defined inventory position $x_{S,t}^{P}$ with a critical inventory level S.

$$p_t = \max\left\{S_t - x_{S,t}^{P}, 0\right\} \tag{1}$$

Based on the rework-specific inventory position $x_{S,t}^{R}$, the rework quantity is chosen to raise the inventory up to a critical level M as far as there is a sufficient quantity of recoverables available.

$$r_t = \min\left\{x_{R,t}, \max\left\{M_t - x_{S,t}^{R}, 0\right\}\right\} \tag{2}$$

In order to define the inventory positions $x_{S,t}^{P}$ and $x_{S,t}^{R}$ adequately we will look at the different leadtime relations separately. First we consider the cases of identical leadtimes and shorter production leadtime ($\lambda_R \geq \lambda_P$).

Production affects the serviceables inventory in period $t+\lambda_P$, but in addition has an effect on the inventory level of period $t+\lambda_P+\lambda_R$ for the following reason: Since only a fraction $z_t \cdot p_t$ of the production order will enter the serviceables inventory after production leadtime λ_P and the remaining quantity will be available for rework in $t+\lambda_P$, this part of the original production quantity will arrive λ_R periods later at the serviceables inventory. It is important to account for the 'delayed' items, particularly if the yield fraction is very low. An inflating production rule (as proposed for the case without rework, see [7]) that focuses only on the time point $t+\lambda_P$ neglects this effect and would generate excess recoverables inventory in the case of lower yield fractions. Therefore, the inventory position for production $x_{S,t}^{P}$ aggregates net stock (including currently available recoverables) minus backorders and all outstanding production and rework quantities.

$$x_{S,t}^{P} = x_{S,t} + x_{R,t} + \sum_{i=1}^{\lambda_P - 1} p_{t-i} + \sum_{i=1}^{\lambda_R - 1} r_{t-i} \tag{3}$$

A rework order at time t will affect the serviceables inventory of period $t+\lambda_R$. In addition, outstanding production orders placed in period $t-\lambda_P+1$ to $t-1$ and rework orders placed in period $t-\lambda_R+1$ to $t-1$ will affect the inventory up to this point of time. Including on-hand inventory minus backorders and estimating the production yield by its expectation gives for $x_{S,t}^{R}$

$$x_{S,t}^{R} = x_{S,t} + \mu_z \cdot \sum_{i=0}^{\lambda_P - 1} p_{t-i} + \sum_{i=1}^{\lambda_R - 1} r_{t-i} \tag{4}$$

If the rework process provides items faster than production, i.e. $\lambda_R<\lambda_P$, then the inventory position for production is given as in (3). Yield losses that arise from all outstanding production orders as well as the yield loss to be expected for the current production order being available (after rework) until period $t+\lambda_P+\lambda_R$ are enclosed in this inventory variable.

The inventory position for rework includes the serviceables on hand minus backorders plus the yield expectation of all production orders as well as all rework orders that will increase the serviceables inventory until $t+\lambda_R$.

$$x_{S,t}^{R} = x_{S,t} + \mu_z \cdot \sum_{i=1}^{\lambda_R} p_{t-\lambda_P+i} + \sum_{i=1}^{\lambda_R-1} r_{t-i} \tag{5}$$

3 A Heuristic for Determining Policy Parameters

As a next step, we will derive parameters for the policy in (1) and (2), based on a modified newsboy approach. The optimal order-up-to level from the basic newsboy model is known to be $S=F^{-1}(\alpha)$ with $\alpha=c^{-}/(c^{-}+c^{+})$ where c^+ denotes the overage (holding) cost, c^- denotes the underage (backlog) cost and $F^{-1}(...)$ is the inverse of the demand cdf. For the problem under consideration, a production underage is associated with marginal costs of v. If, on the other hand, an overage is caused by excess units, then the marginal costs can be estimated by $c^{+}=h_S+\mu_z\cdot\left[(\mu_z-1)\cdot h_S+(1-\mu_z)\cdot h_R\right]$ to reflect the different holding costs associated with both fractions of the production output. Applying these marginal costs to the newsboy optimality condition, we can express the produce-up-to level by

$$S=F_{\lambda_P+\lambda_R+1}^{-1}(\alpha_S) \tag{6}$$

with $\alpha_S=v/\left[v+h_S+\mu_z\cdot\left[(\mu_z-1)\cdot h_S+(1-\mu_z)\cdot h_R\right]\right]$. $F_{\lambda_P+\lambda_R+1}^{-1}$ stands for the inverse cdf of the complete demand net of yield deviations from the respective expectations during the time span of $\lambda_P+\lambda_R+1$ periods.

A surplus unit reworked causes marginal costs of $c^+=h_S-h_R$ since the item is transferred from the recoverables to the serviceables inventory. A single unit short that causes a backlog has marginal costs of v. However, if one excess unit is stocked in the recoverables inventory, additional costs of h_R are incurred so that we can express the underage costs by $c^-=v+h_R$. Therefore, using $\alpha_M=(v+h_R)/(v+h_S)$ we can determine the rework-up-to level by

$$M=F_{\lambda_R+1}^{-1}(\alpha_M) \tag{7}$$

In order to derive expressions for *S* and *M* we determine the parameters of the particular probability function *F*, starting with the cases of equal leadtimes and shorter rework leadtime which can be treated jointly ($\lambda_P \geq \lambda_R$).

First we consider the production decision, and thereby separate the two limiting cases. In case of a very high yield fraction, almost all of the material flow goes directly through production into the servicables inventory. Then, the serviceables inventory on hand $y_{t+\lambda_P}$ at the end of period $t+\lambda_P$ after the demand is met can be written as

$$y_{t+\lambda_P} = x_{S,t} + \sum_{i=1}^{\lambda_P-1} \tilde{z}_{t-i} \cdot p_{t-i} + \sum_{i=1}^{\lambda_R-1} r_{t-i} + \tilde{z}_t \cdot p_t + r_t + \sum_{i=1}^{\lambda_P-\lambda_R} r_{t+i} - \sum_{i=0}^{\lambda_P} \tilde{d}_{t+i} \tag{8}$$

After inserting (8) with $r_t = x_{R,t}$ (describing the steady state) and $\sum_{i=1}^{\lambda_P-\lambda_R} r_{t+i} = \sum_{i=1}^{\lambda_P-\lambda_R} \left(1 - z_{t-\lambda_P+i}\right) \cdot p_{t-\lambda_P+i}$ (reflecting returns becoming available for rework up to period $t+\lambda_P-\lambda_R$ in case $\lambda_P>\lambda_R$) into the respective newsboy condition $Prob\left\{y_{t+\lambda_P} \geq 0\right\} = \alpha_S$ and separating the stochastic variables we obtain

$$Prob\left\{\xi_t^1 := \sum_{i=0}^{\lambda_P} \tilde{d}_{t+i} + \sum_{i=1}^{\lambda_R-1} \left(1-\tilde{z}_{t-i}\right) \cdot p_{t-i} + \left(1-\tilde{z}_t\right) \cdot p_t \leq x_{S,t}^P + p_t\right\} = \alpha_S \tag{9}$$

or $F_{\xi^1}\left(x_{S,t}^P + p_t\right) = \alpha_S$. Using the order-up-to policy condition that production equals recent period's demand, we find that the stochastic variable ξ^1 has mean $\mu_{\xi^1} = \left(\lambda_P + 1\right) \cdot \mu_d + \left(1-\mu_z\right) \cdot \sum_{i=1}^{\lambda_R-1} p_{t-i} + \left(1-\mu_z\right) \cdot d_{t-1}$, and its variance is given by $\sigma_{\xi^1}^2 = \left(\lambda_P + 1\right) \cdot \sigma_d^2 + \sigma_z^2 \cdot \sum_{i=1}^{\lambda_R-1} p_{t-i}^2 + \sigma_z^2 \cdot d_{t-1}^2$.

In the second case, if the yield fraction is very low then most of the material flows through both production and rework processes so that we face a two-stage serial system. The serviceables inventory after demand is met in period $t+\lambda_P+\lambda_R$ is given by

$$y_{t+\lambda_P+\lambda_R} = x_{S,t} + \sum_{i=1}^{\lambda_P-1} p_{t-i} + \sum_{i=1}^{\lambda_R-1} r_{t-i} + p_t + \sum_{i=1}^{\lambda_R} \tilde{z}_{t+i} \cdot \tilde{p}_{t+i} + r_t - \sum_{i=0}^{\lambda_P+\lambda_R} \tilde{d}_{t+i} \,. \tag{10}$$

In this case the yield uncertainty relating to any outstanding production orders (including the current period *t*) disappears, because returns arising from these orders can be reworked until period $t+\lambda_P$ and therefore are available at the serviceables inventory up to period $t+\lambda_P+\lambda_R$. However, future production orders that will be placed in periods t+1 to $t+\lambda_R$ are afflicted with yield risk. Inserting (10) with with r_t=$x_{R,t}$ into $Prob\left\{y_{t+\lambda_P+\lambda_R} \geq 0\right\} = \alpha_S$ yields

$$Prob\left\{\xi_t^2 := \sum_{i=0}^{\lambda_P+\lambda_R} \tilde{d}_{t+i} - \sum_{i=1}^{\lambda_R} \tilde{z}_{t+i} \cdot \tilde{p}_{t+i} \leq x_{S,t}^P + p_t\right\} = \alpha_S \tag{11}$$

or $F_{\xi^2}\left(x_{S,t}^P + p_t\right) = \alpha_S$. After some transformations, ξ^2 can be shown to have mean $\mu_{\xi^2} = \left[\lambda_P + (1-\mu_z)\cdot\lambda_R + 1\right]\cdot\mu_d$ and variance $\sigma_{\xi^2}^2 = (\lambda_P+1)\cdot\sigma_d^2 + \lambda_R\cdot\left[\sigma_z^2\cdot\mu_d^2 + \sigma_d^2\cdot(1-\mu_z)^2 + \sigma_z^2\cdot\sigma_d^2\right]$.

To find a single expression for the produce-up-to parameter S and the corresponding safety stock value SST_P we propose a linear combination of ξ^1 and ξ^2 with weighting factor μ_z, i.e. $\xi := \mu_z\cdot\xi^1 + (1-\mu_z)\cdot\xi^2$ guaranteeing that both limiting cases are properly incorporated. Assuming ξ to be normally distributed and solving $F_\xi\left(x_{S,t}^P + p_t\right) = \alpha_S$ for p_t yields $p_t = \mu_\xi + k_P\cdot\sigma_\xi - x_{S,t}^P$ where k_P denotes the safety factor $k_P = \Phi^{-1}(\alpha_S)$ with Φ^{-1} as the inverse standard normal cdf. $SST_P = k_P\cdot\sigma_\xi$ represents a safety stock. As a result for λ_R>0, we obtain the following expression for the safety stock

$$SST_P = k_P\cdot\sqrt{\begin{aligned}&(\lambda_P+1)\cdot\sigma_d^2 + \mu_z\cdot\sigma_z^2\cdot\left(\sum_{i=1}^{\lambda_R-1} p_{t-i}^2 + d_{t-1}^2\right) + \\ &+(1-\mu_z)\cdot\lambda_R\cdot\left[\sigma_z^2\cdot\mu_d^2 + \sigma_d^2\cdot(1-\mu_z)^2 + \sigma_z^2\cdot\sigma_d^2\right]\end{aligned}} \tag{12}$$

which is added to μ_ξ to calculate the produce-up-to level $S = \left[\lambda_P + (1-\mu_z)^2\cdot\lambda_R + 1\right]\cdot\mu_d + \left(\mu_z - \mu_z^2\right)\cdot\left(\sum_{i=1}^{\lambda_R-1} p_{t-i} + d_{t-1}\right) + SST_P$. In the special case λ_R=0 all returns can immediately be reworked so that the yield risk completely disappears and we obtain $S = (\lambda_P+1)\mu_d + k_P\cdot\sqrt{(\lambda_P+1)\cdot\sigma_d^2}$.

Now we turn to the rework-up-to parameter M. The serviceables inventory after the current rework and production orders are received and demand is met, is given by

$$y_{t+\lambda_R} = x_{S,t} + \sum_{i=1}^{\lambda_R} \tilde{z}_{t-\lambda_P+i}\cdot p_{t-\lambda_P+i} + \sum_{i=1}^{\lambda_R-1} r_{t-i} + r_t - \sum_{i=0}^{\lambda_R} \tilde{d}_{t+i} \tag{13}$$

After inserting (13) into the condition $Prob\left\{y_{t+\lambda_R} \geq 0\right\} = \alpha_M$ with $\alpha_M = (v + h_R)/(v + h_S)$ we obtain

$$Prob\left\{\zeta_t := \sum_{i=0}^{\lambda_R} \tilde{d}_{t+i} - \sum_{i=1}^{\lambda_R}\left(\tilde{z}_{t-\lambda_P+i} - \mu_z\right) \cdot p_{t-\lambda_P+i} \leq x_{S,t}^R + r_t \right\} = \alpha_M \tag{14}$$

or $F_\zeta\left(x_{S,t}^R + r_t\right) = \alpha_M$ where the random variable ζ has mean $\mu_\zeta = \left(\lambda_R + 1\right) \cdot \mu_d$ and variance $\sigma_\zeta^2 = \left(\lambda_R + 1\right) \cdot \sigma_d^2 + \sigma_z^2 \cdot \sum_{i=1}^{\lambda_R} p_{t-\lambda_P+i}^2$. Defining the safety factor k_R as $k_R = \Phi^{-1}\left(\alpha_M\right)$ we can find the following expression for the safety stock value SST_R:

$$SST_R = k_R \cdot \sqrt{\left(\lambda_R + 1\right) \cdot \sigma_d^2 + \sigma_z^2 \cdot \sum_{i=1}^{\lambda_R} p_{t-\lambda_P+i}^2} \tag{15}$$

which is used to compute the corresponding rework-up-to parameter $M = \left(\lambda_R + 1\right) \cdot \mu_d + SST_R$.

For the case $\lambda_R > \lambda_P$, proceeding in the same way we obtain the expression

$$SST_P = k_P \cdot \sqrt{\begin{aligned}&\left(\lambda_P + 1\right) \cdot \sigma_d^2 + \mu_z \cdot \sigma_z^2 \cdot \left(\sum_{i=1}^{\lambda_P - 1} p_{t-i}^2 + d_{t-1}^2\right) + \\ &+ \left(1 - \mu_z\right) \cdot \lambda_R \cdot \left[\sigma_z^2 \cdot \mu_d^2 + \sigma_d^2 \cdot \left(1 - \mu_z\right)^2 + \sigma_z^2 \cdot \sigma_d^2\right]\end{aligned}} \tag{16}$$

for the safety stock and

$$S = \left[\lambda_P + \left(1 - \mu_z\right)^2 \cdot \lambda_R + 1\right] \cdot \mu_d + \mu_z \cdot \left[\left(1 - \mu_z\right) \cdot \left(\sum_{i=1}^{\lambda_{P-1}} p_{t-i} + d_{t-1}\right) + x_{R,t} + \sum_{i=1}^{\lambda_R - \lambda_P - 1} r_{t-i}\right] + SST_P$$

for the produce-up-to level. For the rework decision, the safety stock turns out to be

$$SST_R = k_R \cdot \sqrt{\begin{aligned}&\left(\lambda_P + 1\right) \cdot \sigma_d^2 + \sigma_z^2 \cdot \sum_{i=0}^{\lambda_P - 1} p_{t-i}^2 + \\ &+ \left(\lambda_R - \lambda_P\right) \cdot \left[\sigma_z^2 \cdot \mu_d^2 + \sigma_d^2 \cdot \left(1 - \mu_z\right)^2 + \sigma_z^2 \cdot \sigma_d^2\right]\end{aligned}} \tag{17}$$

and the respective rework decision parameter is given by $M = \left(\lambda_R + 1\right) \cdot \mu_d - \mu_z \cdot \left(\lambda_R - \lambda_P\right) \cdot \mu_d + SST_R$.

Finally, we present some results of a computational study to show how the above heuristic performs in an infinite horizon setting. In a base-case scenario, demand is normally distributed with μ_d=100 and σ_d=15, the yield rate is beta-distributed with μ_z=0.5 and σ_z=0.1, holding and shortage costs are h_S=3, h_R=2 and ν=27, leadtimes are λ_R=λ_P=12. Numerical results are given in Tab. 1 where the yield rate expectation and leadtimes are varied under ceteris paribus conditions. To evaluate the performance of our heuristic approach we used the ordering policy as described in section 2 as well as a production policy based on a linear inflation rule and optimised over their parameters. Costs are expressed relatively to the respective numerically obtained lower bound which has been standardised to 100. As it can be seen from the data, in the case of equal leadtimes (λ_R=λ_P) the heuristic performs quite well for general yield rate situations. This can also be observed if λ_R<λ_P. However, if the rework leadtime is very long compared to the production leadtime then the performance considerably deteriorates. In this particular case, an inflating production rule seems to be preferable.

Table 1. Cost performance for different yield and leadtime conditions

μ_z	C	λ_R	C	λ_P	C
0,05	100	0	101	0	131
0,20	100	2	101	2	109
0,35	100	4	100	4	101
0,50	100	6	100	6	100
0,65	101	8	100	8	100
0,80	100	10	100	10	100
0,95	100	12	100	12	100

4 Conclusions

In this paper we discussed a stochastic yield problem with rework and focused on the question of formulating an adequate ordering policy and determine their parameters. In comparison to inflating production rules as known from heuristics for the problem without rework, the proposed (*S*,*M*)-policy is still suitable to cope with very low yield situations where the model shows characteristics of a serial two-stage inventory problem. Our results show that, except for cases where the rework leadtime is very long compared to the regular production leadtime, the heuristic approach guarantees a high performance. For the situations with bad performance further research is needed to develop a better heuristic approach.

References

1. Bollapragada S, Morton T (1999) Myopic heuristics for the random yield problem. Operations Research 47(5):713-722
2. Gerchak Y, Vickson R, Parlar M (1988) Periodic review production models with variable yield and uncertain demand. IIE Transactions 20:144-150
3. Henig M, Gerchak Y (1990) The structure of periodic review policies in the presence of random yield. Operations Research 38(4):634-643
4. Inderfurth K, Jensen T (1999) Analysis of MRP with recovery options. In: Leopold-Wildburger U, Feichtinger G, Kistner K (eds) Modelling and decisions in economics, Springer, Heidelberg, pp 189-228
5. van der Laan E, Kiesmüller G, Kuik R, Vlachos D, Dekker R (2004) Stochastic inventory control for product recovery management. In: Dekker R, Fleischmann M, Inderfurth K, van Wassenhove, L (eds) Reverse Logistics: Quantitative models for closed-loop supply chains, Springer, Heidelberg, pp 181-220
6. Minner S, Lindner, G (2004) Lot sizing decisions in product recovery management. In: Dekker R, Fleischmann M, Inderfurth K, van Wassenhove, L (eds) Reverse Logistics: Quantitative models for closed-loop supply chains, Springer, Heidelberg, pp 157-179
7. Zipkin PH (2000) Foundations of inventory management, McGraw-Hill, Boston

Life cycle considerations in remanufacturing strategies – a framework for decision support

Wiebke Stölting, Thomas Spengler

Technical University of Braunschweig, Institute for Economics and Business Administration, Department of Production Management, Katharinenstr. 3, 38106 Braunschweig, Germany, Email: w.stoelting@tu-bs.de; t.spengler@tu-bs.de.

Abstract. Remanufacturing is the most valuable product recovery option since here the value added to the product can be obtained. The goal of this contribution is to develop an instrument which supports decision makers in their considerations about implementing a remanufacturing strategy. The framework is built on a two-stage proceeding, where in the first step an approach for modeling this strategic decision process is presented. In the second step, the contribution aims to show how to evaluate alternative decisions economically by developing a life cycle model for remanufacturing and calculating the life cycle costs.

1 Introduction

Typically, legal reasons are the main driver for the take-back of products at the end of the use phase [11]. Regulations, e.g. on waste electrical and electronic equipment [4], define producers' responsibility for their products throughout the entire product life including the after-use phase. In addition to this, [7] identify business motives for product take-back like market forces and recapturing hidden economic value. Actually, these drivers contribute to the increasing significance of recovery strategies.

Two main types of recovery options can be identified in dependence of the object they refer to: product recovery and material recovery [13]. Remanufacturing is a high-grade product recovery option where the product is disassembled to part level, all modules and parts are inspected and repaired or replaced if necessary and the product is upgraded to an as new quality level. Another product recovery option is cannibalization, where a limited set of reusable parts are recovered and used as spare parts or for the production of new products.

Current literature on remanufacturing mainly covers specific operational [e.g. 10, 6] and process oriented [e.g. 8, 9] topics. Thus, the focus of this contribution is placed on the strategic decisions related to remanufacturing and aims at the development of a concept for decision support for choosing a recovery option.

2 Model of the decision situation

In order to realize legal demands and to pursue recovery options in an economically efficient way a number of strategic decisions have to be made by the producer in an insecure environment.

In order to be able to balance decision alternatives, external influencing factors which will develop in an unknown way have to be taken into consideration. Depending on the particular specification of these factors one decision alternative might be more promising under a business perspective than another. Relevant external factors that influence the internal decisions about the choice of the recovery option are:

- Market development
- Development of competitors
- Development of technology
- Development of legal regulations
- Return flow of discarded products
- State of returned products

For the given decision situation we assume that subjective probabilities for the development of these factors can be achieved by questioning experts about their expectation and to derive the needed data from their statements.

The characteristics of the decision process considered in our field of research can be subsumed as follows: The producer faces a multi-stage dynamic decision process under risk. An approach to model this type of decision situation is the decision tree, which shows the logical structure of the decision problem and contains all relevant elements of the decision situation as outlined in Fig. 1 [3].

t : Period, $t = 1,...,T$

j : Decision alternatives

i : States of nature

x_{tj} : Binary decision variable $x_{tj} = \begin{cases} 1, \textit{if alternative j is chosen} \\ 0, \textit{else} \end{cases}$

s_{ti} : State of nature in t which leads to decision node / result D_{ti}/R_{ti} $(i=1,...,I)$

D_{ti} : Decision node i in period t; $D_{ti} = \{d_{tj}$ *which are possible*$\}$

S_{tj} : Chance node j in period t; $S_{tj} = \{s_{ti}$ *which are possible*$\}$

p_{ti} : Probability that state of nature s_{ti} will occur

R_{ti} : Result node if the state of nature s_{ti} occurs in period t: $R_{ti} = R(s_{ti})$

Rtj : Result node of a decision alternative j in period t; $Rtj = Rti$

if $i \in \{1,...,I\} \wedge s_{ti} \in S_{tj}$

$EV\{R_{tj}\}$: Expected value of result R_{tj}

μ^*t : Maximum expected value of all decision alternatives j in period t

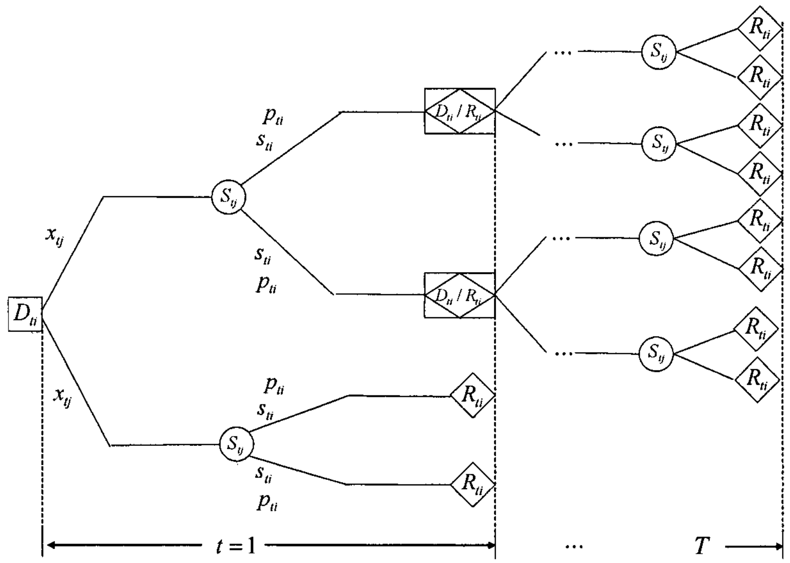


Fig. 1. Basic structure of a decision tree

Squares indicate decision nodes D_{ti} where a number of different decision alternatives j can be made by the decision maker in period t. Circles denote chance nodes S_{tj} representing states of nature characterized by the external influencing factors above. At each chance node a number of different states of nature s_{ti} can occur. To each of the states of nature a subjective probability $p_{ti} = p(s_{ti})$ for its occurrence can be assigned. The final leaves of the tree show the ultimate consequences and therefore the results $R_{ti} = R(s_{ti})$ associated with each decision alternative j and state of nature s_{ti} combination. In cases where the result of a decision alternative and state of nature combination leads to a new decision that has to be made, the nodes are denoted by D_{ti}/R_{ti}.

Since for the stochastic situation considered in our field of research the results R_{ti} will only occur with a certain probability p_{ti} we have to use the expected value approach to identify the best decision alternative j [1]. The expected value of the result R_{tj} of a decision alternative j is the sum of the weighted results R_{ti} for the decision alternative j:

$$EV\{R_{tj}\} = \sum_{i\in\{1,\dots,I\}\wedge s_{ti}\in S_{tj}} p_{ti}\cdot R_{ti} \qquad (1)$$

R_{tj} becomes R_{ti} if state of nature s_{ti} occurs subsequent to the decision alternative j that was chosen. In each period t the maximum expected value of all decision alternatives j has to be detected:

$$\mu_t^* = \max\left(EV\left(R_{tj}\right)\right) \qquad (2)$$

Then the binary decision variable x_{tj} denotes that the decision alternative j with the maximum expected value is chosen:

$$x_{tj} = \begin{cases} 1, \textit{if } EV\left(R_{tj}\right) = \mu_t^* \\ 0, \textit{else} \end{cases} \tag{3}$$

The probabilities of the states of nature following one chance node must add to one:

$$\sum_{i \in \{1,\ldots,I\} \wedge s_{ti} \in S_{tj}} p_{ti} = 1 \tag{4}$$

At each decision node one decision has to be made:

$$\sum_{j \in \{1,\ldots,J\} \wedge d_{tj} \in D_{ti}} x_{tj} = 1 \qquad \forall t \tag{5}$$

The problem formulation stated above is valid for single-stage decision problems. In order to determine the optimal decision policy for a multi-stage decision problem the evaluation of the different decision alternatives has to take place at each stage of the decision tree. The optimal policy can then be determined using the dynamic optimization approach [2] of backward recursion (Fig. 2).

The starting point is the last stage of the problem for which an optimal solution must be found. Then, one has to move step by step backwards finding the optimal solution for each stage under consideration of the preceding result until every stage including the first is covered and the optimal policy for the entire decision problem has been found.

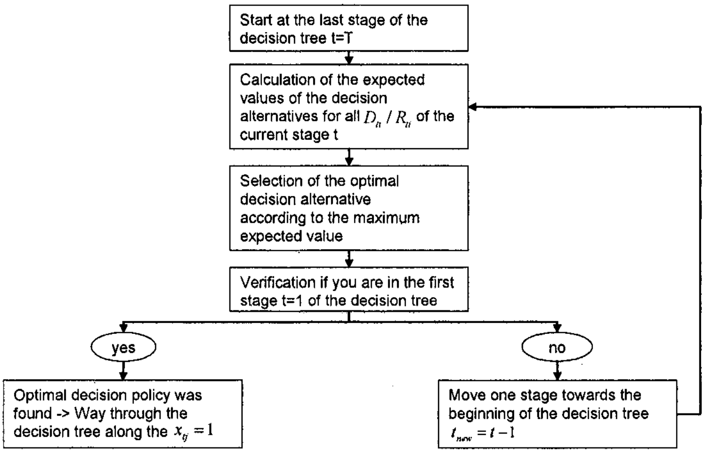

Start at the last stage of the decision tree t=T
Calculation of the expected values of the decision alternatives for all D_{ti} / R_{ti} of the current stage t
Selection of the optimal decision alternative according to the maximum expected value
Verification if you are in the first stage t=1 of the decision tree
yes
no
Optimal decision policy was found -> Way through the decision tree along the $x_{tj} = 1$
Move one stage towards the beginning of the decision tree $t_{new} = t - 1$

Fig. 2. Flowchart for the backward recursion of the decision tree

A simplified decision tree representing the decision about the adequate recovery option is illustrated by Fig. 3.

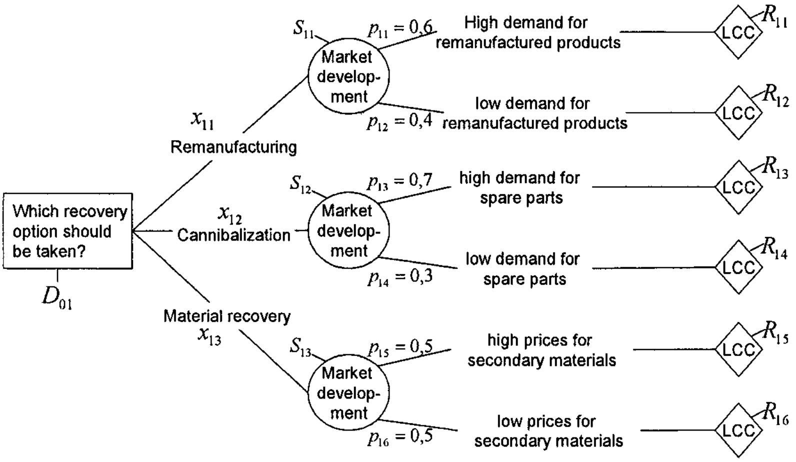


Fig. 3. Decision tree for choosing a recovery option

The identification of the optimal decision can take place by assigning a value to each of the ultimate consequences represented by the final leaves of the tree. Since the choice of a recovery option is a product oriented decision, we can use the product life cycle costing (LCC) approach to ascertain this value [5]. For that purpose, we have to model the product life cycle and determine the net present value related to the cash flows of each decision alternative and state of nature combination. This will take place in the following section.

3 Evaluation of the life cycle costs

In this section a model will be developed that represents the entire life cycle of a product under consideration of the remanufacturing option[1]. Based upon this model, the life cycle costing calculation can be realized in the subsequent step of the concept. In this contribution the life time of a product is considered as the time the company has to deal with the idea of the product starting with research and development and then covering the actual physical existence of the product through production and use until all the individual members of the product project end their life.

[1] To keep this contribution short we will not depict the life cycle models for the other recovery options, they have to be set up likewise.

When developing a life cycle model for remanufacturing a connection between product project oriented modeling approaches and models concerning product individuals has to be established. A product project can be understood as the integrated consideration of all the single pieces belonging to one type of product in contrast to the isolated consideration of every single piece of product as a product individual [12]. The combination is reasonable since the strategic decisions are made for the product project while the cash flows surveyed for the life cycle costing calculation are referred to the individual products.

The model of the product project life cycle comprises the three main periods of preparation, production and after-care. Research and development of the product are carried out during the preparation period. The production period covers the time span of the market distribution of the product and ranges from start until end of production. During the after-care period finally all the activities related to the products that are still out in the field after the end of production are conducted like e.g. repair service or disposal.

We integrate a pattern of the individual product's life into this model. The consideration of product individuals is especially reasonable during the production period since here the relation of cash flows to each product individual is established. A relation of the product individual to the preparation period does not seem reasonable since development does not form part of the life of each individual product. The product individual runs through the phases of production, distribution, use, take-back, remanufacturing, remarketing, 2nd use, take-back and disposal.

The integrated depiction of these two perspectives takes place in one life cycle model for remanufactured products (Fig. 4). Here, the life cycle of a product individual starts at a certain point during the product project's production period and ends some time later during that period.

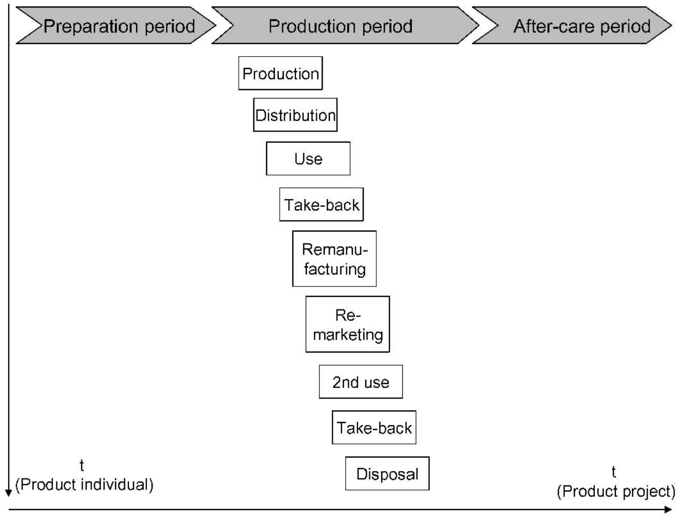


Fig. 4. Integrated closed-loop product life cycle model for remanufacturing

The decision about the recovery option can be considered as an investment decision. Thus, methods of dynamic investment appraisal can be applied. The cash flow for the preparation and for the after-care period is estimated for the entire product project. For the market period the cash inflows and outflows for each product individual can be determined for each phase of its life cycle. In dependence of the assumptive development of the sales figures an aggregation of these cash flows for the entire product project can be conducted. Thus, the cash flow of the production period can be determined. Based upon this data a calculation of the net present value (NPV) of the investment (the remanufacturing decision) can take place:

$$NPV = \sum_{t=1}^{T} (Ci_t - Co_t) \cdot (1+r)^{-t} \tag{6}$$

t : Period of the product project life cycle, $t = 1,...,T$
Ci_t : Cash inflow in period t
Co_t : Cash outflow in period t
R : rate of interest

The NPV for the other recovery options can be calculated likewise. For the determination of the cash flows it is necessary to consider the possible states of nature concerning the market development identified in the decision tree. The NPVs are finally assigned to the corresponding branches of the tree. The evaluation of the decision tree using the life cycle costing calculation of the NPVs and the probabilities given by experts will then lead to the identification of the optimal decision.

4 Perspective: Further strategic planning problems

The decision situation described in this contribution represents only one sub problem of the strategic planning problems that relate to the implementation of a remanufacturing system. There are a number of additional strategic questions that have to be considered by the decision maker. Relevant among these are:

- Should the remanufacturing process be outsourced or carried out internally?
- Which capacities should be set up for the recovery process?
- Should certain agreements concerning the return of the products be concluded between producer and costumer?

For each of these sub problems an economic evaluation of the different decision alternatives has to be carried out. A combination of the economically efficient alternatives will then lead to the optimal overall strategy. To give an example: A producer implements a remanufacturing strategy for the products taken back, carries the process out himself, sets up a certain capacity for the remanufacturing process and distributes the products via leasing contracts.

When evaluating the different alternatives it seems to be reasonable to take into account, that certain strategies offer the possibility to postpone decisions to a later point in time. Thus, the decision maker can watch the development of the external influencing factors and gather additional information. This flexibility should be considered in the process of evaluating the alternatives. This is the field of research we will address in the future.

References

1. Anderson D, Sweeney D, Williams T (2000) Introduction to management science: quantitative approaches to decision making. South-Western College Publishing, Cincinnati
2. Bellmann R, Dreyfus S (1962) Applied Dynamic Programming. Princeton University Press, Princeton, New Jersey
3. Blohm H, Lüder K (1995) Investition. Franz Vahlen Publishing, Munich.
4. European Commission (2003) Directive 2002/96/EC of the European Parliament and of the Council on waste electrical and electronic equipment. Brussels
5. Fabrycky W, Blanchard B (1991) Life-Cycle Cost and Economic Analysis. Prentice-Hall, Englewood Cliffs, New Jersey
6. Guide V. (2000) Production planning and control for remanufacturing: industry practice and research needs. Journal of Operations Management 18: 467-483
7. Gungor A, Gupta S (1999) Issues in environmentally conscious manufacturing and product recovery. Computer & Industrial Engineering 36: 811-853
8. Inderfurth K (2002) Optimal Policies in Hybrid Manufacturing/ Remanufacturing Systems with Product Substitution. Working Paper Series, Faculty of Economics and Management, Otto von Guericke University, Magdeburg, Germany Working Paper No. 1/2002
9. Lambert A (2003) Disassembly sequencing: a survey. International Journal of Production Research 16: 3721-3759

10. Minner S (2001) Economic production and remanufacturing lot-sizing under constant demands and returns. In: Proceedings of the Symposium on Operations Research (OR 2000). Springer, Berlin Heidelberg New York, pp. 328-332
11. Ritchey J, Mahmoodi F, Frascatore M, Zander A (2001) Assessing the Technical and Economic Feasibility of Remanufacturing. In: Proceedings of the Twelfth Annual Conference of the Production and Operations Management Society (POM-2001). Orlando, FL. March 30 – April 2, 2001
12. Siestrup G (1999) Produktkreislaufsysteme. Erich Schmidt Publishing, Berlin
13. Thierry M, Salomon M, van Nunen J, van Wassenhove L (1995) Strategic Issues in Product Recovery Management. California Management Review 2: 114-135

Stochastic Models of Customer Portfolio Management in Call Centers

Oualid Jouini[1], Yves Dallery[1], and Rabie Nait-Abdallah[2]

[1] Ecole Centrale Paris, Laboratoire Génie Industriel, Grande Voie des Vignes,
92295 Chatenay-Malabry Cedex, France
jouini@lgi.ecp.fr, dallery@lgi.ecp.fr

[2] Bouygues Telecom, 20 Quai du Point du Jour, 92640 Boulogne Billancourt
Cedex, France
rnait@bouyguestelecom.fr

Abstract. We investigate the interest of migrating from a call center where all agents are pooled and customers are treated indifferently by any agent, towards a call center where customers are grouped into clusters with dedicated teams of agents. Each cluster will be called a portfolio. Customers of a same portfolio are always served by an agent of the corresponding team. There is no specialization involved in this organization in the sense that all customer portfolios as well as all agents teams have (statistically) identical behaviors. The purpose of this paper is to investigate how the benefits of moving to this new organization in terms of the management of the workforce can outweigh its drawback that comes from the increasing of variability.

1 Introduction

A call center is a complicated service system, in which managers must take into account the behavior of both customers and agents [1]. The purpose of this paper is to provide insights to help managers in the design and management of call centers. It is the result of a collaboration with *Bouygues Telecom*, a french mobile phone company. We investigate the adequacy of migrating from a call center where all agents are pooled and customers are treated indifferently, towards a call center where customers are grouped into clusters with dedicated agents. The aim of *Bouygues Telecom* through migrating into customer portfolio management is to better manage their employees and as a consequence to satisfy customers more accurately. This management approach makes agents more responsible towards their own customers. Moreover, partitioning agents into groups creates competition, which motivates agents to give better answers to customers. In this paper, we argue how these advantages may outweigh the variability that results from the loss in economy of scale originally associated with the pooled system. Also, in the proposed organization, all portfolios and corresponding set of dedicated agents are totally identical (statistically). Therefore, issues such as training and forecasting can be done in a homogenous manner.

Our study is linked to pooling phenomenon. Pooling in queueing systems has been described first by Kleinrock [2]. Results on pooling queues are obtained in [3], [4], and [5]. Whitt has also studied the issues of partitioning customers into service groups [6]. All above results do not take into account the human element. This takes us to a second area of literature, that is, human element. It is the main characteristic of call centers and contact centers. Both customers and agents are people [7]. Indeed, call center management requires a mix of disciplines that are not typically found in organizations [8]. The review of Boudreau [9] follows through this new area. He propose a framework which is a fertile source of research opportunities.

The remainder of this paper is structured as follows. In Section 2, we develop a simple queueing model that is then used to address the issue of benefits versus costs of migrating from the pooled organization to the dedicated organization. In Section 3, we extend this analysis to the situation where there is an out-portfolio flow. Finally, we conclude and propose some directions for future research.

2 Analysis of the Efficiency of the Team-Based Organization

In this section we present simple models of call centers. Complex aspects, such as abandonment, retrials, time-varying operations, etc, are not considered. Analyzing simple models are relevant by their intuitive interpretations, and insights to support managers' decisions.

2.1 The Models

The first model is a call center viewed as an $M/M/C$ (Erlang C) queue with s identical servers; arrival process of customers (or calls) is assumed to be Poisson, service times are assumed to be exponentially distributed and independent of each other, and the service discipline is assumed to be first-come, first-served (FCFS). The mean interarrival time is given by $1/\lambda$, and $1/\mu$ is the mean service time. Blocking, abandonment, and retrials are ignored. We refer to this model as the Pooled System (see Fig. 1).

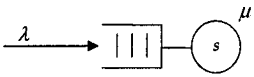


Fig. 1. Pooled System

The second model is the unpooling of the first model to n identical call centers, so that s is a multiple of n. Each call center is viewed as an $M/M/C$ queue with s_n identical servers. The arrival rate to each call center is λ_n, and the service rate is μ_n. We refer to this model as the Dedicated System (see Fig. 2).

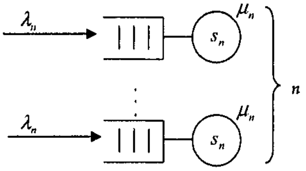


Fig. 2. Dedicated System

In the Dedicated System, customers are grouped into identical portfolios. We assume that all the portfolios have the same arrival and service rates, and we associate to each portfolio a set of service agents (pool of servers). On the one hand, 100% of customers in this system are served by their dedicated service agents which provides significant benefits in terms of quality of response. But on the other hand, this way of management requires that: either we increase the total number of servers to reach the same *QoS* in terms of waiting time, or we let customers wait longer than in the case of the Pooled System. We focus on the *QoS* measured by the average waiting time before beginning service W and the 80/20 rule $W(t < 20\ s)$, which is an industry standard for telephone service [1]. Under the 80/20 rule, at least 80% of the customers must wait no more than 20 s. In *Bouygues Telecom*'s call center, like in most call centers, the arrival rate of calls varies over time. We assume constant number of agents, and constant arrival and service rates, as well as a system that achieves a steady-state quickly within each half-hour interval of time. Under the stability condition $\lambda/s\mu < 1$, performance measures in terms of the waiting time distribution in queue, are obtained using the Erlang C formulas. Note that the Pooled System is a special case of the Dedicated System. They are identical when $n = 1$.

The comparison between the Pooled System and the Dedicated System will always show that, when keeping the service rate μ fixed, the average waiting time is smaller in the first system for any total arrival rate λ and any total number s of servers. Under these conditions, it is intuitively clear that the Dedicated System is less efficient because a call may wait for one server while another server is idle, such situation does not occur in the Pooled System. It was shown in [3] that efficiency increases when separate traffic systems are combined into a single system. However, it is assumed that the service time distributions are the same in the systems being combined. This common distribution condition is crucial since it may be in some cases disadvantageous to combine systems with different service time distributions [6].

2.2 Evaluation of Service Rate Percentage Increase

We start from a Pooled System with a given QoS ($W(t)$ or W), and our purpose is to evaluate the required service rate in a Dedicated System with n pools in order to ensure the same QoS ($W_n(t) = W(t)$ or $W_n = W$). Total staffing level and total arrival rate are held constant.

We consider a Pooled System with an arrival rate of $\lambda = 197.06$ calls per min, the service rate is $\mu = 0.2$ calls per min (the average service time is 5 min) which is the case in *Bouygues Telecom*), and there are $s = 1000$ agents in the call center. In this system, 80% of customers wait no more than 20 s and the corresponding average waiting time is $W = 0.18$ min. In the Dedicated System, each call center has a staffing level $s_n = s/n = 1000/n$, and an incoming of calls rate of $\lambda_n = \lambda/n$. We vary n from 1 to 50. For each number n of separated call centers, we calculate the service rate μ_n, so that, the average waiting time is $W_n = 0.18$ min. We repeat the same analysis for the QoS in terms of the 80/20 rule. Fig. 3 shows the curves of the required percentage of service rate increase, calculated as $100 \times (\mu_n - \mu)/\mu$, according to number of pools n in order to reach $W_n = 0.18$ min and $W_n(20s) = 80\%$, respectively. For both types of QoS, the required increase of the service rate do not grow in a dramatic fashion. We notice that for a Dedicated System with $n = 20$ separate systems, required mean service time is about 4 min and 25 s in order to reach $W_n = 0.18$ min, and it is about 4 min and 30 s in order to reach $W_n(20s) = 80\%$. The required value is not too far from the actual mean service time (5 min). In practice, an increase in service rate efficiency of the order of 10% seems very reasonable to achieve because of the team-based organization.

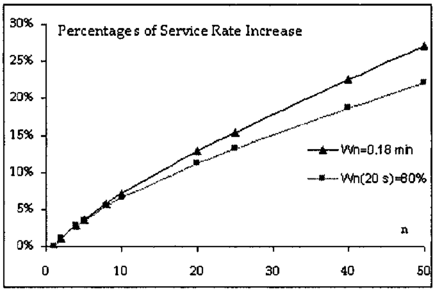


Fig. 3. Percentages of service rate increase according to number of pools n in a Dedicated System in order to achieve $W_n = 0.18$ min and $W_n(20s) = 80\%$

Another advantage of dedicated systems is their robustness with respect to errors in the estimation of the arrival rate of calls. For example, consider the Pooled System, $n = 1$, described above and the Dedicated System, $n = 10$, obtained by increasing the service rate in order to ensure the same QoS as in the pooled system. The QoS of the Pooled System is much more affected than the one of the Dedicated System by an underestimation of the arrival rate of calls. Let us give an explanation. Under the original expected arrival rate, the server utilization in the Pooled System, 98.53%, is closer to 1 than the one in the Dedicated System, 91.99%. When the arrival rate is underestimated, the deterioration of the quality of service is increasing faster when the server utilization is closer to 1 since the queue becomes less and less stable. For example, assume now that we underestimate the total arrival rate of calls (which is now $\lambda = 197.06$ calls per min for both systems) by only 1.41%. Then real server utilization of the Pooled System becomes 99.92% and the one of the Dedicated System becomes 93.28%. As a consequence, the average waiting time of the first system goes beyond 5 min and the one of the second system is only 0.27 min. This shows that the team-based organization is much more robust than the original pooled organization. This is a very attractive feature that gives another strong argument in favor of the team-based organization.

3 Call Center Models with Out-Portfolio Flow

In this section, we address the same issues as in Section 2 by considering two new models (a pooled model and a dedicated model) of call centers. They differ from the above models by taking an anonymous flow (out-portfolio flow) of calls into account. The latter consists of calls for which one cannot associate a portfolio to them when they enter the call center. An anonymous call can be a call of a customer of *Bouygues Telecom* who does not communicate his phone number to the Computer-Telephone Integration (CTI), a person who is not a customer of *Bouygues Telecom*, etc.

3.1 The Models

The first model is a call center viewed as a non-preemptive priority $M/M/C$ queue with the parameters λ, μ and s servers. The arrival rate λ is composed of the identified flow (PTF) and the anonymous flow (OPTF). An anonymous call is served only if at least one server is idle and no identified call is in queue. Let p be the proportion of the anonymous call flow (In Bouygues Telecom p is order of 10%). The arrival rate of the identified flow is $(1 - p)$ λ, and the arrival rate of the anonymous flow is p λ. We refer to this model as the Portfolio Pooled System (see Fig. 4).

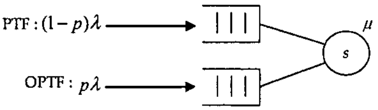


Fig. 4. Portfolio Pooled System

As in the subsection 2.1, we propose a second model given by unpooling the Portfolio Pooled System to n identical call centers, so that s is a multiple of n. Each call center is viewed as a non-preemptive priority $M/M/C$ queue with s_n identical servers and a service rate μ_n. The total arrival rate of the PTF flow is $\lambda^{PTF} = (1-p)\ \lambda$. The arrival rate to each call center is $\lambda_n^{PTF} = \lambda^{PTF}/n$, and there is an OPTF flow which must wait in a lower priority queue. This queue is served only when at least one of the s service agents is idle and no customer is waiting in the corresponding portfolio queue. The arrival rate of the OPTF flow is $\lambda^{OPTF} = p\ \lambda$. We refer to this model as the Portfolio Dedicated System (see Fig. 5).

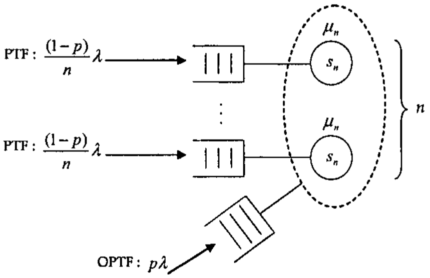

PTF : $\frac{(1-p)}{n}\lambda$
μ_n
s_n
PTF : $\frac{(1-p)}{n}\lambda$
μ_n
s_n
n
OPTF : $p\lambda$

Fig. 5. Portfolio Dedicated System

Performance measure of the Portfolio Pooled System (non-preemptive priority $M/M/C$ queue) in terms of the waiting time distribution was treated by (Davis 1965). Literature on priority in queueing systems is abundant [2], [10]. However, the particular configuration of the Portfolio Dedicated System makes its quantitative analysis complicated. For this, we developed a set of models that give us lower bounds of performance measures of the Portfolio Dedicated System. So the improvement in efficiency that will follow from our analysis can be viewed as an upper bound on the efficiency improvement that will actually be required.

3.2 Evaluation of Service Rate Percentage Increase

We define a global quality of service QoS^{global} for all types of customers as $QoS^{global} = (1-p)\, QoS^{PTF} + p\, QoS^{OPTF}$. In what follows, we focus only on the QoS measured in terms of the average waiting time. Similar results could be obtained for the 80/20 rule. We start from a Portfolio Pooled System with a given quality of service W^{global}, and our purpose is to evaluate the required service rate μ_n in a Portfolio Dedicated System with n identical teams in order to ensure the same global average waiting time $W_n^{global} = W^{global}$. Total staffing level, and total arrival rate of calls are held constant.

In the Portfolio Pooled System, the arrival rate is $\lambda = 197.06$ calls per min, the service rate is $\mu = 0.2$ calls per min, and there are $s = 1000$ agents in the call center. Server utilization is then $\rho = 98.53\%$. In addition, let the proportion of OPTF customers varies, $p = 0\%$, $p = 5\%$, $p = 10\%$, or $p = 20\%$. As expected, W^{global} does not depend on p. Indeed, in a Portfolio Pooled System, we have not the case that one server is idle while a customer (PTF or OPTF) is waiting for service. If we vary p, order of service of a given customer may change, but the overall average waiting time remains unchanged. Here, $W^{global} = 0.18$ min for all values of p. For each value of p, we consider now the corresponding Portfolio Dedicated System. We vary n from 1 to 50. For each n, we evaluate the required service rate μ_n (using pessimistic models), so that, the global average waiting time is $W_n^{global} = W^{global} = 0.18$ min. In Fig. 6, we plot for each value of p, the curve of the percentage of required service rate increase, calculated as $100 \times (\mu_n - \mu)/\mu$, in the Portfolio Dedicated System according to the number of pools n, in order to reach a global quality of service of $W_n^{global} = 0.18$ min. We notice that for a given p, we have the same qualitative results as in Section 2.2. Indeed, the required increase of the service rate is not very important and it is feasible to reach in practice. In particular, when $p = 0\%$ we find the same quantitative results, because the Portfolio Pooled System reduces to the Pooled System and the Portfolio Dedicated System reduces to the Dedicated System.

The new interesting insight here is that the more the proportion of OPTF customers is important, the more the impact in terms of the required increase of service rate is reduced. Particulary, migrating towards a Portfolio Dedicated System (with any $p > 0\%$) is always less costly than migrating towards a Dedicated System ($p = 0\%$). For example, consider a Portfolio Dedicated System with $n = 10$. If the OPTF proportion is $p = 5\%$, we need to increase the service rate by 4.91%. However, with a proportion $p = 20\%$ we need only to increase the service rate by 1.17%. We explain this advantage by the fact that the OPTF flow is used to reduce idle periods of servers while customers are waiting in the Portfolio Dedicated System. Idle times would not exist in the case of $p = 100\%$ (Pooled System).

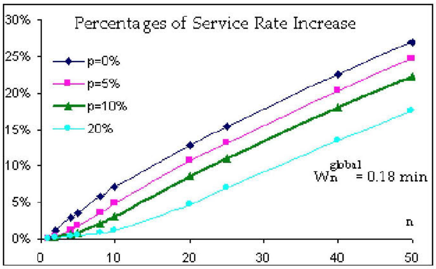


Fig. 6. Percentages of required service rate increase according to number of pools n in a Portfolio Dedicated System in order to achieve $W_n^{global} = 0.18$ min

4 Conclusion

We focused on a fundamental problem in the design and management of call centers, and in general, stochastic service systems, that is, pooling effects versus team management benefits. We argued how team management benefits, that comes from the portfolio/team one to one link, may outweigh the economy of scale associated with the pooled organization. First, we study partitioning of a large call center into identical and separated call centers, where agents of a same team are dedicated to one portfolio of customers. Queueing models involved in this part of the study are simple. They give us important insights and help us to understand the behavior of more complicated systems. Then, we extend our analysis to the more general situation with an additional out-portfolio flow. We show that the costs of migrating towards the Dedicated System are not as important as it may appears. We also present further insights, as robustness of dedicated systems regarding errors in the estimation of the arrival rate. In a future study, we will extend our models by considering abandonments and limited waiting lines and more general service times distributions. We will also try to improve the approximation models discussed here to get more accurate analysis. Finally, a more ambitious extension will be to investigate the introduction of team-based organization in a call center where agents have specific skills.

References

1. Gans N., Koole G., Mandelbaum A. (2003) Telephone Call Centers: Tutorial, Review, and Research Prospects. Manufacturing & Service Operations Management. **5**, 73–141
2. Kleinrock L. (1976) Queueing Systems, Computer Applications. A Wiley-Interscience Publication. Vol II
3. Smith B., Donald R., Whitt W. (1981) Resource Sharing for Efficiency in Traffic Systems. Bell System Technical Journal. **60**, 39–55
4. Rothkoph M. H., Rech P. (1987) Perspectives on Queues: Combining Queues is not Always Beneficial. Operations Research. **35**, 906–909
5. Whitt W. (1992) Understanding the efficiency of multi-server service systems. Management Science. **38**, 708-723
6. Whitt W. (1999) Partitioning Customers into Service Groups. Management Science. **45**, 1579-1592
7. Whitt, W. (2002) Stochastic models for the design and management of customer contact centers: Some research directions. Department of Industrial Engineering and Operations Research, Columbia University
8. Fischer M., Garbin D. et al. (1999) Traffic engineering of distributed call centers: Not as straight forward as it may seem. Mitretek Systems
9. Boudreau J., Hopp W. et al. (2003). On the interface between operations and human ressources management. Manufacturing & Service Operations Management. **5**, 179–202
10. Kella O., Yechiali U. (1985) Waiting times in the non-preemptive priority M/M/C queue. Stochastic models. **1**, 257–262

AN MILP MODELLING APPROACH FOR SHELF LIFE INTEGRATED PLANNING IN YOGHURT PRODUCTION

M. Lütke Entrup[1], M. Grunow[1], H.O. Günther[1], T. Seiler[1] and P. van Beek[2]

[1] Department of Production Management, Technical University Berlin,
Wilmersdorfer Str. 148, D-10585 Berlin, e-mail: HO.Guenther@pm-berlin.net

[2] Operations Research and Logistics Group, Wageningen University,
Hollandseweg 1, NL-6706 KN Wageningen, e-mail: Paul.vanBeek@wur.nl

Abstract. In the production of perishable products such as dairy, meat, or bakery goods, the consideration of shelf life in production planning is of particular importance. Retail customers with relatively low inventory turns can benefit significantly from longer product shelf life as wastage and out-of-stock rates decrease. However, in today's production planning and control systems shelf life issues with regard to specific products or customers are seldom taken into account. Therefore the objective of this paper is to pay attention to these issues. The way to do that is by means of optimization models in which shelf life aspects are integrated into operational production planning and scheduling functions. Specifically we make use of so-called Mixed Integer Linear Programming (MILP) models. Our research is based on an industrial case study of yogurt production. Relying on the principle of block planning, an MILP model for weekly production planning is presented that is based on a combination of a discrete and a continuous time representation. Batch sizing and scheduling of numerous recipes and products on several packaging lines are considered in the model. Overnight production and, hence, the necessity for identifying two different shelf life values for the same batch is also included in the model formulation. Numerical experiments show that near-optimal solutions can be obtained within a reasonable computational time. Finally, the proposed MILP model can be adapted to cover specific features arising in other fresh food industries.

1. Introduction

Production planning of yogurt is certainly one of the most challenging tasks in the dairy industry as the planner has to cope with e.g. an extraordinary high number of products and variants as well as with sequence-dependent set-up times and costs on capital-intensive processing equipment. One important distinctive factor to consider in fresh food production planning is shelf life. Shelf life restrictions directly influence wastage, inventory levels and out-of-stock rates in the retail outlets. Furthermore, consumers tend to buy the product with the longest possible shelf life. The possibility to offer a longer shelf life than its competitors constitutes a pivotal competitive advantage for fresh food producers. Hence, the consideration of shelf life is crucial for production planning systems in the dairy or other fresh food industries. The remainder of this paper is organized as follows. A short literature review of research of scheduling in make-and-pack production as well as on production planning for perishable products is given in Section 2. An introduction into the production of yogurt is provided in Section 3. In the main part of this paper (see Section 4), a MILP model is presented that integrates shelf life into production planning and scheduling. The model is numerically validated in Section 5. Finally, an outlook is given on its applicability in other fresh food industries.

2. Literature Review

Batch production in the chemical industry can be taken as a reference for the production of yogurt as both must consider numerous variants, which are based on few product types or recipes. In literature, this production environment is named "make and pack production" (e.g. Neuhaus et al., 2002; Méndez and Cerdá, 2002). Major issues of operational production planning are lot sizing and scheduling, which can be performed in one single or two separate planning steps. As set-ups in yogurt production are sequence-dependent, the exact set-up costs and times can only be determined after the sequencing of the orders. Moreover, as sequencing depends on lot sizing, both tasks must be performed simultaneously (c.f. Sikora et al., 1996). Neuhaus et al. (2002) present an approach that simultaneously considers lot sizing and scheduling based on the block-planning principle. By integrating several variants of a product type or recipe into a "block", the complexity of the model can be significantly reduced without being unrealistic. For the determination of the sequence of batches within a block, a "natural" sequence of batches often exists, for example from the lower taste to the stronger or from the brighter color to the darker.

With regard to the consideration of shelf life in production planning, two different avenues can be distinguished. A vast body of literature exists on *inventory management* for perishable products, which covers food products as well as the behavior of radioactive materials, photographic film, prescription drugs, or blood conserves. Nahmias (1982) and Raafat (1991) give comprehensive literature overviews and an analysis of proposed inventory models for perishables. The major drawback of all perishable inventory models is that production issues are almost completely neglected although the shelf life of products is actually determined by their time of production. In addition, production capacities, sequence-dependent set-up times, or production on multiple units or lines are not reflected. Considering the integration of shelf life into *production planning and scheduling*, most research is concerned with adding a shelf life constraint to the Economic Lot Scheduling Problem. Soman et al. (2004) provide a review of the major contributions. Shortcomings of this model type include the constant demand rate assumption, the single facility consideration and the neglect of sequence-dependent set-up times.

3. Production of Yogurt

The production of yogurt consists of five major steps. First, the raw milk is collected from the farms by tanker trucks and stored in silos in the dairy factory. Second, the raw milk is subject to a number of preparatory treatments, which include its standardization (in many cases with milk powder), its heating in a plate heat exchanger, its concentration in an evaporator and its homogenization. Third, the raw milk is transformed into yogurt during the fermentation process by adding starter cultures. For stirred yogurt, the fermentation is performed in tanks while set yogurt is fermented in the packaging after filling. This research, however, focuses on stirred yogurt. In fermentation, the recipe complexity is still relatively low compared to the packaging step. Fourth, the yogurt is flavored with fruits and other ingredients and filled into the retail containers on capital-intensive filling and packaging lines. This flavoring and packaging step has to cope with a high number of product variants, which is caused by the vast variety of packaging materials and tastes. In packaging, sequence dependent set-up times occur; nonetheless in many cases a natural sequence exists that can help to avoid extensive set-up times (e.g. the brighter color before the darker). Fifth, the packed products are stored and delivered to the retail outlets.

4. Model Formulation

The presented MILP-model aims at generating a weekly plan for the production of yogurt and focuses particularly on the filling and packaging stage. It is based on the block planning approach; all products that can be filled on the same packaging line and that are based on the same recipe are part of the same block. The model applies a combination of a continuous and a discrete time representation. Start and end times of batches on the packaging lines are reflected by continuous variables, while a discrete time grid is used for shelf life considerations. The planning horizon covers one week from Monday to Friday, including the possibility to use overtime capacity on Sunday before and on Saturday after the planning week. The demand to be covered also includes Monday and Tuesday of the following week.

Indices, sets

$j,k \in J$	final products
$l \in L$	lines
$s \in S$	days
$p \in P \subset S$	production days
$d \in D \subset S$	demand days
$r \in R$	recipes, blocks
$j \in J(r)$	products based on recipe r
$l \in LR(r)$	lines that can process recipe r
$l \in LJ(j)$	lines that can process product j
$d \in D(s)$	demand days (to meet the demand of these days, lots produced on day s can be considered)
$s \in S(d)$	days (the lots produced on these days can be considered to meet the demand of demand day d)

Data

rev_j	revenue for selling one unit of product j
$varc_j$	variable costs for the production of one unit of product j
sl_j	shelf life of product j in days
B	sufficiently large number
cap_l	capacity of line l, in units per day
$ster_l$	time for sterilizing line l
cl_l	time for cleaning line l
ben_j	maximum additional benefit for meeting the maximum shelf life of product j, in € per unit
$loss_r$	cleaning loss of fermented plain yogurt of recipe r, in kg
$closs_r$	cost for the cleaning loss of plain yogurt of recipe r, in € per kg
d_{jd}	demand of product j on demand day d
s_{js}	inventory of product j, produced on day s
c_l	costs of utilization of line l on a weekday, in € per day
mb	minimum batch size to be processed, in kg
ps_j	packaging size of product j, in kg per unit
ft_r	fermentation time for recipe r, in hours
fc	fermentation capacity, in kg-hours per day
cr_j	minimum remaining shelf life of product j required by the customer (in % of maximum shelf life)
q_j	quarantine time of product j
os_l	overtime supplement for weekend production on line l, in € per day
ad_j	percentage of plain yogurt contained in one unit of product j
fdp	start of the first possible production day within the week (Sunday)
ldp	start of the last possible production day within the week (Saturday)
bp_j	position of product j in the corresponding block

Decision variables

$S_{rpl} \in \{0,1\}$	=1, if recipe r is produced on production day p on line l (0, otherwise)
$C_{jpl} \in \{0,1\}$	=1, if product j is produced until the end of production day p-1 and at the beginning of day p (0, otherwise)
$X_{jpl} \in \{0;\infty]$	units of product j produced on line l on production day p
$Z_{jds} \in \{0;\infty]$	units of product j produced on production day s that are used to meet the demand of demand day d
$L_{rpl} \in \{0;\infty]$	duration of recipe/block r on production day p on line l
$END_{rpl} \in \{0;\infty]$	end time of recipe/block r on production day p on line l
$EST_l \in \{0;\infty]$	start time of line l in the production week
$LFT_l \in \{0;\infty]$	end time of line l in the production week
$SAO_l \in \{0;\infty]$	overtime at the end of the week on line l
$SUO_l \in \{0;\infty]$	overtime at the beginning of the week on line l

Objective function

$$\begin{aligned}\max &\sum_{j\in J}\sum_{d\in D}\sum_{s\in S(d)} Z_{jds}\cdot ben_j\cdot\frac{\left(\left(1-cr_j\right)\cdot sl_j\right)-\left(d-s\right)}{\left(1-cr_j\right)\cdot sl_j}+\sum_{j\in J}\sum_{d\in D} d_{jd}\cdot rev_j\\ &-\sum_{j\in J}\sum_{p\in P}\sum_{l\in LJ(j)} X_{jpl}\cdot varc_j-\sum_{l\in L}\left(LFT_l-EST_l\right)\cdot c_l-\sum_{l\in L}\left(SAO_l+SUO_l\right)\cdot os_l \qquad (1)\\ &-\sum_{p\in P}\sum_{r\in R}\sum_{l\in LR(r)}\left(S_{rpl}-\sum_{j\in J(r)} C_{jpl}\right)\cdot loss_r\cdot closs_r\end{aligned}$$

The objective function aims at maximizing the contribution margin taking into account a shelf life dependent pricing component. As the demand must be covered (Equation 5c), the regular revenues are fixed in the model. They are only included in the objective function in order to better interpret the objective value. Costs include the variable costs for input factors, the costs of the utilization of the packaging lines in regular and overtime mode and the set-up costs for cleaning losses.

Set-up of lines

$$\sum_{j\in J(r)} X_{jpl} \le S_{rpl}\cdot B \qquad p\in P;\ r\in R;\ l\in LR(r) \qquad (2)$$

Output Quantities

$$\left(S_{rpl}-\sum_{j\in J(r)} C_{jpl}\right)\cdot ster_l+\left(S_{rpl}-\sum_{j\in J(r)} C_{j,p+1,l}\right)\cdot cl_l+\sum_{j\in J(r)}\frac{X_{jpl}}{cap_l}\le L_{rpl}$$

$$p\in P\colon p<ldp;\ r\in R;\ l\in LR(r) \qquad (3a)$$

$$\left(S_{rpl} - \sum_{j \in J(r)} C_{jpl}\right) \cdot ster_l + S_{rpl} \cdot cl_l + \sum_{j \in J(r)} \frac{X_{jpl}}{cap_l} \leq L_{rpl}$$

$$p \in P: p = ldp;\ r \in R;\ l \in LR(r) \qquad (3b)$$

The utilization of line l is determined by the production volume per product divided by the capacity of the line and the necessary sterilization and cleaning times.

Sequencing and day bounds

$$END_{rpl} \geq L_{rpl} + END_{r-1,p,l} \qquad p \in P;\ r \in R: r>1;\ l \in LR(r) \qquad (4a)$$

$$END_{rpl} \leq p+1 \qquad p \in P;\ r \in R;\ l \in LR(r) \qquad (4b)$$

$$END_{rpl} - L_{rpl} \geq p \qquad p \in P;\ r \in R;\ l \in LR(r) \qquad (4c)$$

Stock balance and demand satisfaction

$$\sum_{l \in L(j)} X_{jpl} \geq \sum_{d \in D(s)} Z_{jdp} \qquad j \in J;\ s \in S: ldp \geq s \geq fdp;\ p \in P: p=s \qquad (5a)$$

$$s_{js} \geq \sum_{d \in D(s)} Z_{jds} \qquad j \in J;\ s \in S \qquad (5b)$$

$$d_{jd} = \sum_{s \in S(d): s+q_j < d \wedge (d-s) \leq (1-cr_j) \cdot sl_j} Z_{jds} = \sum_{s \in S(d)} Z_{jds} \qquad j \in J;\ d \in D \qquad (5c)$$

Utilization of packaging lines

$$LFT_l \geq END_{rpl} - \left(1 - S_{rpl}\right) \cdot B \qquad p \in P;\ r \in R;\ l \in LR(r) \qquad (6a)$$

$$EST_l \leq END_{rpl} - L_{rpl} + \left(1 - S_{rpl}\right) \cdot B \qquad p \in P;\ r \in R;\ l \in LR(r) \qquad (6b)$$

$$SUO_l \geq fdp + 1 - EST_l \qquad l \in L \qquad (6c)$$

$$SAO_l \geq LFT_l - ldp \qquad l \in L \qquad (6d)$$

Minimum batch size

$$\sum_{j \in J(r)} \left(X_{jpl} + X_{j,p+1,l}\right) \cdot ps_j \cdot ad_j + \left(1 - \sum_{j \in J(r)} C_{j,p+1,l}\right) \cdot B \geq mb$$

$$r \in R;\ p \in P: p < ldp;\ l \in LR(r) \qquad (7a)$$

$$\sum_{j \in J(r)} X_{jpl} \cdot ps_j \cdot ad_j + \left(\sum_{j \in J(r)} C_{jpl} \right) \cdot B + \left(\sum_{j \in J(r)} C_{j,p+1,l} \right) \cdot B \geq mb \cdot S_{rpl}$$

$$r \in R;\ p \in P\!: p < ldp;\ l \in LR(r) \qquad (7b)$$

$$\sum_{j \in J(r)} X_{jpl} \cdot ps_j \cdot ad_j + \left(\sum_{j \in J(r)} C_{jpl} \right) \cdot B \geq mb \cdot S_{rpl}$$

$$r \in R;\ p \in P\!: p = ldp;\ l \in LR(r) \qquad (7c)$$

Fermentation capacity

$$\sum_{r \in R} \left(\sum_{j \in J(r)} \sum_{l \in LR(r)} X_{jpl} \cdot ps_j \cdot ad_j \right) \cdot ft_r + \sum_{r \in R} \sum_{l \in LR(r)} \left(S_{rpl} - \sum_{j \in J(r)} C_{jpl} \right) \cdot loss_r \cdot ft_r \leq fc \qquad p \in P \ (8)$$

Conservation of the set-up state

$$\sum_{k \in J(r): bp_k > bp_j} X_{kpl} \leq \left(1 - C_{j,p+1,l} \right) \cdot B \qquad r \in R;\ j \in J(r);\ p \in P\!: p<ldp;\ l \in LR(r) \quad (9a)$$

$$END_{rpl} - p \geq C_{j,p+1,l} \qquad r \in R;\ j \in J(r);\ p \in P\!: p<ldp;\ l \in LR(r) \quad (9b)$$

$$X_{jpl} \geq C_{j,p+1,l} \qquad j \in J;\ p \in P\!: p < ldp;\ l \in LJ(j) \quad (9c)$$

$$END_{rpl} - L_{rpl} - p \leq 1 - \sum_{j \in J(r)} C_{jpl} \qquad r \in R;\ p \in P;\ l \in LR(r) \quad (9d)$$

$$X_{jpl} \leq \left(1 - \sum_{k \in J(r)} C_{kpl} + \sum_{k \in J(r): bp_k \leq bp_j} C_{kpl} \right) \cdot B \qquad r \in R;\ j \in J(r);\ p \in P;\ l \in LR(r) \ (9e)$$

$$\sum_{j \in J} C_{jpl} \leq 1 \qquad p \in P;\ l \in L \quad (9f)$$

$$S_{rpl} - \sum_{j \in J(r)} C_{jpl} \geq 0 \qquad r \in R;\ p \in P;\ l \in LR(r) \quad (9g)$$

$$C_{jpl} \leq 0 \qquad j \in J;\ l \in LJ(j);\ p = fdp \quad (9h)$$

In case of overnight production, the value of 1 is assigned to the binary variable C_{jpl}. This is only allowed if product j is produced until the end of the day and the production of the same product or a successor within the same block starts directly at the beginning of the following day.

5. Computational Results

The data set used to demonstrate the applicability of the proposed models consists of 30 products based on 11 recipes that are processed on 4 packaging lines (3 different line types). All products are assigned to a specific line type. The numerical investigation was performed on a PC with AMD XP 2600+ CPU and 1 GB RAM using ILOG's OPL Studio 3.6.1 as a modeling environment and the standard optimization software CPLEX 8.1. To solve the model, short computing times are desirable as the model is used every week for the weekly production planning. Furthermore, the model can be used for simulation purposes. When optimizing all four lines simultaneously, the model delivered an objective function value of 1,416,089 € after 300s with a remaining MIP gap of 2.37%. The shelf life dependent pricing component accounts for ca. 10% of the revenues. After 1,800s, the MIP gap was reduced to 2.34% with an unchanged objective value.[1] If each of the three line types is optimized separately, the objective value is increased to 1,431,929 € after 330.1s (sum of CPU times for all three optimization runs) with a remaining MIP gap of 0.57%. Therefore, decomposing the problem has shown to be effective in reducing computational times without impairing the overall quality of the production schedule. When analyzing the different line types in detail, it can be concluded that the presented model is particularly appropriate for large-scale bulk production with a limited variety of recipes, which requires the assignment of several shelf lives to a production batch.

6. Conclusion

The MILP model presented has shown to be suitable to generate near-optimal solutions for a planning and scheduling problem from the dairy industry. As the product shelf life has been considered explicitly, the proposed planning tool can be applied to generate production schedules, which result in an improved product freshness. As a next step, the transferability of the approach to fresh food products should be examined. Apart from other dairy products, shelf life integrated planning is especially advisory for meat products. The shelf life of fresh meat is even less than the shelf life of fresh yogurt products (e.g. minced meat products only last a few days), which increases the influence of the freshness on the purchasing decision. Moreover, fish, fruits, and vegetables as well as bakery goods may be subject to shelf life integrated planning.

[1] The MIP gaps always depend on the used data set.

References

Méndez, C. A., Cerdá, J., An MILP-based approach to the short-term scheduling of make-and-pack continuous production plants, in: OR Spectrum, 24 (2002), p. 403–429

Nahmias, S., Perishable Inventory Theory, A Review, in: Operations Research, 4 (1982), p. 680–708

Neuhaus, U., Günther, H.-O., Gür, B., Yang, G., An MILP model for flexible block planning in make-and-pack production, in: Proceedings of the 4th Asian Pacific Conference on Industrial Engineering and Management Systems, Taipei (2002) p. 375–378

Raafat, F., Survey of Literature on Continuously Deteriorating Inventory Models, in: Journal of the Operational Research Society, 42 (1991), p. 27–37

Sikora, R., Chhajed, D., Shaw, M. J., Integrating the lot-sizing and sequencing decisions for scheduling a capacitated flow line, in: Computers and industrial Engineering, 30 (1996), p. 659–679

Soman, C. A., van Donk, D. P., Gaalman, G. J. C., A basic period approach to the economic lot scheduling problem with shelf life considerations, in: International Journal of Production Research, 42, 8 (2004), p. 1677-1689

Dynamic optimization of routing in a Semiconductor Manufacturing Plant

Hermann Gold

Infineon Technologies AG, Wernerwerkstrasse 2, D-93049 Regensburg, Germany
e-mail: hermann.gold@infineon.com

Abstract. We consider a semiconductor manufacturing facility with multiple products and associated routes, single servers and batch servers, job class dependant service times and external arrivals. The system is modelled as an open queueing network. The aim is to optimize routing such that cycle time constrained capacity is maximized and can be checked in reasonable computation time. We use a decomposition approach based on the connected components of a properly defined fab graph. Taking into consideration arrival rate vectors and service time matrices the routing problem for the network is formulated as a Quadratic Programming Problem (QP) involving averages and variances. The strategy for use of the manifold routing options as they typically occur in semiconductor manufacturing is to distribute load in a way such that each connected component, also called closed machine set (CMS), approaches heavy traffic resource pooling behaviour. In the presence of batch servers, mainly in the furnace area of a fab, results for the batch service queue $M/D^{[r,K]}/1$ with threshold server starting policy are combined with a new result for a batch service system with infinitely many job classes and Round Robin service discipline and applied along with the QP solver.

1 Introduction and Problem Statement

In this paper we consider the problem of routing in a semiconductor front end manufacturing facility (called fab throughout this paper) as part of shop floor scheduling. Material is started into the fab in units of lots, also called jobs, where each lot includes a prespecified number of wafers. There is no defined upper bound on the number of jobs in the fab. To a small degree admission control is imposed by specific batch service operations occurring at some early process step for a given product, but as material flow is reentrant, and since scheduling can generally not be sufficiently tuned with respect to downstream batch operations because of product variety, this sort of admission control is neglected. For the interior of our network the reentrance of flow has the consequence that scheduling policies are taken to be non-anticipating. Therefore the mathematical model for the type of fabs considered is an open stochastic queueing network with external arrivals.

Each type of the multiple products is manufactured by accomplishing a product specific set of tasks in some specific order. The activity of processing a specific task requires a single machine resource, single server or batch server. Alternate machines as a means of accomplishing a task are considered as routing alternatives. Routing alternatives are memoryless, i. e. the set of machines of which the system manger can choose from to perform a given task does not depend on earlier choices for routing alternatives regarding the associated product. Routing alternatives can be categorized as to whether a routing decision has to be done immediately after a job finishes a previous step and enters the queue for the machine where the next processing step will be done, or each machine can choose from a global queue where all jobs belonging to a job class, which is qualified on that machine, reside. The former is called push routing, the latter is called pull routing (see e.g. [9]). In practice push routing does not exist in its pure sense, but the machines qualified for one and the same process can be located in different halls. Hence a set of machines can be partitioned into subgroups with push routing on the level of the subgroups and pull routing inside the subgroups.

In our mathematical model all the routing-relevant information of a job is included in the number of the closed machine set (CMS) currently visited and the number of the job class as which this CMS is visited by that job. A CMS is a connected component of the fab graph. The fab graph contains all machines of the fab as vertices and an edge between every pair of machines for which a process or task exists which is qualified on both machines. As we deal with Markovian routing, the problem of routing optimization can be decomposed into optimizing routing for each CMS independently.

For the network described we aim at minimizing some linear or convex function of cycle times (waiting plus service) for job classes as well as linear operating costs (e. g. costs for recycling wafers, chemicals, energy and labour). A widely used approach for controlling stochastic multiclass queueing networks, ideally in heavy traffic, is heavy traffic approximation using Brownian network models on the grounds of [5] - applications are typically shown under 90% load. This approach offers valuable insights into the behaviour and most critical aspects concerning a CMS. It uses discretization methods (called BIGSTEP, see [6]) to find the best candidate of discrete review policies for some given problem instance, however only the most basic parameters of such a policy can be found in a straightforward way. Furthermore the so called nominal plan which is input to the Brownian network is suggested to be created by a linear program. In our type of problems for a CMS with m job classes and n servers this nominal plan would select only at most $m + n - 1$ of the totally possible routing alternatives as routing options to be used by a scheduling policy (see e. g. [7]). This is optimal under heavy traffic in the strict sense (Utilization $U \rightarrow 1^{-}$) but far from optimal under the usual load conditions in a semiconductor wafer fab since for a fully flexible multi-server with k servers cycle time decreases with growing k. In addition there exists a more theoretical problem. The LP solution underlying the nominal plan in [7] might include routing alternatives in its optimal basis with respective value 0. Thus the graph for the corresponding CMS with edges between two machines, only when there exist routing alternatives in the nominal plan with probability > 0 for some job class on both machines, would be disconnected, hence the resources would not pool in the sense of [8] when applying the corresponding scheduling policy, which is a big disadvantage.

Incorporating the lessons learned from Harrison and other's work on Brownian network modelling, we approach the optimization problem as follows. i.) Create a nominal plan which balances utilization levels of the different machines of a CMS near the minimum possible value and distribute the load of each job class broadly upon the machines qualified to process it. ii.) In the presence of batch servers the goal of minimizing avoidable idleness has to be put back behind the goal of balancing the server efficiency η, defined as the average number of jobs per start on a batch machine divided by the maximum number of jobs per start on that batch machine, and cycle time targets for job classes with relatively low traffic intensity. i.) implies the aim to employ heavy traffic resource pooling in non-heavy traffic instances. Heavy traffic resource pooling is the art to make certain nodes of a stochastic network act as one single resource pool with respect to queue length (see [8] for a survey on resource pooling) under heavy traffic. For the types of network considered here this art comes down to making n servers with heterogeneous task qualifications inside a particular CMS behave as a G/G/n system. Since routing is Markovian, resource pooling of machines belonging to different CMS cannot occur.

The nominal plan is calculated by a quadratic program (QP) in our approach and is explained in the next section. The incorporation of estimates for server efficiencies in the presence of batch servers is described in Section 3. Experiences from application of the methods in the physical world of a semiconductor fab are summarized in Section 4.

2 Nominal Plan and Scheduling Policy

In the perspective of the decomposition of the fab graph, as defined in Section 1, into connected components the different routes of our open stochastic queueing network are given by $z_l \times 2$ matrices where z_l is the number of steps on route R_l for product type l as follows

$$R_l = \begin{pmatrix} cms_1 & jc_1 \\ cms_2 & jc_2 \\ & \vdots \\ cms_{z_l} & jc_{z_l} \end{pmatrix}, \qquad l \in L. \tag{1}$$

The first column of this matrix denotes closed machine sets as they are successively visited by route R_l. The second column gives the job classes as which a product of type l shows up at the CMS of each corresponding row as it travels through the network along route R_l. Lot release is given by a vector a of arrival rates a_l, $l \in L$, where L is the index set of product types with distinct routes. Product types with identical routes have the same index in this set. In the stationary case the vector a is used to calculate the arrival rates at the different closed machine sets. In the instationary case, e. g. during ramp-up or major changes in product mix, the vector of arrival rates is a multi-dimensional stochastic process $a(t)$ with time dependent arrival rates $a_l(t)$ where the index set L also depends on time. Note that there is no freedom of choice regarding routing at the level of routing described by the matrices of eqn. (1).

In the following an optimization technique for the routing of jobs inside closed machine sets is developed. Job classes with the same machine qualifications and identical service times on any given machine are identified and represented by one single job class index, the corresponding arrival rates are added. Beyond that, it is sufficient to distinct only job classes with linearly independent rows in the service time matrix as presented below. This latter reduction is particularly important, when there exist several of thousands of job classes as for instance in the photolithography area.

The following notation is used for a particular closed machine set.

m number of job classes
n number of machines
I Index set for job classes $I = \{1, \ldots, m\}$
J Index set for machines $J = \{1, \ldots, n\}$
F Indicator function matrix, with $f_{ij} = 1$ if job class i is qualified on machine j and 0 otherwise
B service time matrix with b_{ij} the service time of job class i on machine j
P branching probability matrix with p_{ij} the probability that a job of class i is processed on machine j
U utilization vector with u_j as the average utilization of machine j
λ arrival vector with λ_i the arrival rate of jobs from class i

The utilization vector associated with the closed machine set considered is given by

$$u_j = \sum_{i=1}^{m} \lambda_i p_{ij} f_{ij} b_{ij} \qquad j \in J \tag{2}$$

We consider the following optimization problem:

Minimize
$$\begin{aligned} f &= c_1 E[U] \;+\; c_2 Var[U] \qquad (3) \\ &= c_1 \frac{1}{n} \sum_{j=1}^{n} u_j \;+\; c_2 \sum_{j=1}^{n} (u_j - \frac{1}{n} \sum_{j=1}^{n} u_j)^2 \end{aligned}$$

under the constraints
$$\begin{array}{ll} 0 \leq p_{ij} \leq 1, & i \in I, j \in J \\ \sum\limits_{j=1}^{n} p_{ij} = 1, & i \in I \\ f_{ij} = 0 \Rightarrow p_{ij} = 0, & i \in I, j \in J \\ 0 \leq u_j \leq 1, & j \in J \end{array}$$

Typically we choose $c_2 \gg c_1$ in eqn. (3) The objective function f of this minimization problem is quadratic and the constraint functions are linear. Hence we deal with a quadratic programming (QP) problem. Writing the problem in quadratic form standard solution techniques can be applied. As a result we get the optimal branching probability matrix P, also called nominal plan. As a by-product of the solution of the quadratic program we get the groups of machines which can be loaded homogeneously on a time average bases. If in addition the CMS graph with edges associated only to the activities corresponding to some $p_{ij} > 0$ in the nominal plan is still connected, these groups would fulfil the heavy traffic resource pooling condition under heavy traffic, i. e. if the arrival vector were scaled up by some specific constant.

We finally come to the problem of identifying scheduling policies for each CMS in a straightforward way from nominal plans. As we do not operate the different CMS under ideal heavy traffic conditions we work with continuous review policies, basically using the concept of polling. A polling table is assigned to each server of each CMS following [1]. Beyond that we also attach a polling table to each queue, i. e. job class, of the system. The former polling tables are applied, when a server finishes a job and at least one of the job classes being polled by the server is non-empty at this time epoch. This mode will be dominant under heavy traffic. It is sometimes also referred to as sequencing in the literature, but we keep to the nomenclature of [9] or [7] and consider sequencing as the problem of ordering jobs within job classes. The polling table for queues comes into play when a job arrives to an empty queue and at least one of the servers which are qualified for and poll this job is free at the moment of the arrival. This situation is prevalent under low traffic. The polling table for servers and job classes are calculated on the basis of the branching probability matrices for servers and job classes, respectively. A cycle stealing mechanism is implemented, when an inflexible machine is in danger to starve because a flexible machine drains too many jobs from a flexible job class, or when particular constellations of states of the bottleneck subnetwork appear. More details will be given in a separate paper.

3 Batch Service Systems

Batch servers are servers which, upon becoming free, can collect multiple lots up to a certain maximum, all of whom receive service simultaneously. Hence there is a trade-off as to when a service period should be started with a partially full batch. Therefore, in the presence of batch servers inside a particular CMS the nominal plan has also to tell when to build a batch for processing. We assume that maximal batch sizes inside a particular CMS are identical on one and the same machine for all qualified job classes and that the service times are nearly independent of machines in batch server environments, as it is common in semiconductor furnace areas. For single server batch service systems with Markovian arrivals it is known from [2] that the optimal policy is of threshold type. We apply this type of scheduling also in the multi-server multiple job class case with a mixture of single and batch arrivals in the arrival stream and non-anticipating scheduling, i. e. the principal design of our batch collection rule is 'Allow for building a batch for job class i on machine j only when there are at least r_{ij} jobs of class i waiting to be processed, $1 \leq r_{ij} \leq K_{ij}$, where K_{ij} is the maximum number of jobs per start for job class i on server j. First order data now includes server efficiencies (as defined in Section 1) and the resulting effective machine utilizations u_{eff}. These characteristics as well as normalized waiting time can be calculated for individual batch machines with one job class using the analysis presented in [4] adapted for arrival processes as mentioned above. We also consider the system $M/D^{[r,K]}/1$ with threshold r and maximum batch size K, infinite waiting room, infinitely many job classes, each job class appearing with equal probability, Round Robin service discipline, and equal mix of single arrivals and batch arrivals of size K in the arrival stream with the following exception: The first arrival of each service cycle with respect to any job class is a single arrival. This reflects the fact, that the first lot of a batch will be seperated from the group after processing for sample testing thus imparting a single arrival through the reentrance of flow. For this special queueing system we have developed the following limit for the server efficiency via simulation analysis.

$$\lim_{m \to \infty} \eta = \max \left(U_{FB}, \frac{\binom{K+1}{2} - (K-r)(1+K/2)}{\binom{K+1}{2} - (K-r)K/2} \right), \tag{4}$$

denoted by η_∞ in the sequel, where U_{FB} is the nominal load under full batch policy (start server only with maximally sized batches). For a batch service system with finitely many job classes and job classes not appearing with equal probability we use the entropy of the job classes on a given machine to interpolate between the server efficiency η_1 of the system with only one job class and η_∞. Let the vector of positive probabilities of job classes allocated to a given machine j be q. Then entropy of job classes $H(q)$ on that machine is given by $\sum -q_i \, ld(q_i)$, ld the logarithmus dualis. We consider the virtual number w_j of jobs experienced on that machine as $w_j = 2^{H(q)}$. Finally we calculate $\eta_{i,j}$ for job class i on machine j according to

$$\eta_{i,j} = e^{-(w_j - 1)} \eta_{1,i,j} + (1 - e^{-(w_j-1)}) \eta_{\infty,i,j} \tag{5}$$

Hereby $\eta_{\infty,i,j}$ is calculated according to (4) with U_{FB} substituted by $U_{FB,j}$, K substituted by K_j and r substituted by $r_{i,j}$. Note that for the calculation of both $\eta_{1,i,j}$ and $\eta_{\infty,i,j}$ the total load seen on machine j has to be taken into account. The vectors U_{FB} and q for each $j \in J$ are won from QP solution with each term on the right hand sides of eqns. (2) divided by the corresponding $K_{i,j}$. The individual load contributions of job class i on machine j are denoted by $u_{FB,i,j}$. e is the Euler number.

Now we estimate effective machine utilizations and cycle times for job classes for some given threshold matrix of a CMS as follows:

1. Calculate effective load contributions $u_{eff,i,j} = u_{FB,i,j}/\eta_{i,j}$.
2. Calculate normalized waiting time to build a batch according to [3].
3. Calculate the normalized waiting times for batches using $M/D^{[r,K]}/1$ analysis, where $r = K := \lfloor \eta_{i,j} K_{i,j} \rfloor$.
4. Calculate total normalized waiting time by adding the results from 2. and 3. (as in [3]). The resource pooling effect is captured by division of the summands of 3. by the respective number of machines in the resource pool, thus yielding a (pessimistic) upper bound for the contribution on average waiting times for batches. Operating costs are calculated in the obvious way using $u_{eff,i,j}$ and $\eta_{i,j}$.

These estimates correlate well with simulated results for problem instances with given threshold matrix. For smaller CMS with a number of possible threshold matrices < 500 a full search is possible to find the optimal threshold matrix with respect to cycle time targets and operational costs. Above this value we use gradually increased upper limits for the waiting times to build a batch, this way restricting the region of possible threshold matrices.

4 Practical Application and Outlook

We have presented a fab modelling approach which combines methods from quadratic optimization, queueing theory, graph theory and simulation. The benefits of the application of the methods presented in a fab can be divided into three categories. Firstly, there is a gain in transparency of fundamental network data by reporting it in table form on the basis of CMS and heavy traffic resource pools. Secondly, for short term planning on a weekly or monthly basis the ability to decide whether a fab can handle a potential lot release scenario has been improved, mainly in two aspects. Computational times have been reduced due to proper decomposition, and we have now a clear cut between instances where eventually occurring imbalance of load is due to too bad scheduling and where imbalance of load is unavoidable due to dedication. Using simple load balancing heuristics can lead to utilization vectors with the maximum utilization up to 20% higher than under the optimal scheduling policy. Thirdly, scheduling parameters are given to system managers which helped for instance to improve overall server efficiency in the furnace area significantly by about 1.4%.

Amongst the manifold details which should be included in the queueing network model in the future, the following are considered most important: Modelling of job classes which require more than one resource for processing, e. g. reticles in the photolithography area, and parameterization of set up strategies with parameters gained from mathematical optimization. As far as the application of analytic queueing model results is concerned we will step towards more general arrival processes, e. g. Markov-Modulated Poisson Processes (MMPP). Last but not least a queueing theoretical proof of formula (4) for a specific batch service system with infinitely many job classes is considered as a challenging problem.

References

1. Boxma O. J., Levy H., Westrate J. A. (1991) Efficient visit frequencies for polling tables: minimization of waiting cost. Queueing Systems **9**, 133–162
2. Deb R. K. and Serfozo R. F. (1973) Optimal Control of Batch Service Queues. Adv. Appl. Prob. **5**, 340–361
3. Fowler, J. W. and Phojanamongkolkij, N. , Cochran, J. ,K., Montgomery, D. C. (2002) Optimal batching in a wafer fabrication facility using a multiproduct $G/G/c$ model with batch processing. Int. J. Prod. Res. **40**, No. 2, 275–292
4. Gold, H. and Tran-Gia, P. (1993) Performance analysis of a batch service queue arising out of manufacturing systems modelling. Queueing Systems **14**, 413–426
5. Harrison, J. M. (1988) Brownian models of queueing networks with heterogeneous customer populations. In: Fleming, W., Lions, P. L. (Eds.): Stochastic Differential Systems, Stochastic Control Theory and Applications. Springer, New York, 147–186
6. Harrison, J. M., The BIGSTEP approach to flow management in stochastic processing networks. In: F. Kelly, F., S. Zachary, S.,Ziendins, I. (Eds.): Stochastic Networks: Theory and Applications Oxford University Press, Oxford, 57–90
7. Harrison, J. M., Lopez, M. J. (1999) Heavy traffic resource pooling in parallel-server systems. Queueing Systems **33**, 339-368
8. Kelly, F. P., Laws, C. N. (1993) Dynamic routing in open queueing networks: Brownian models, cut constraints and resource pooling. Queueing Systems **13**, 47-86
9. Wein, L. M. (1991) Brownian networks with discretionary routing, Operations Research **39**, No. 2, 322-340

Blood Platelet Production: a multi-type perishable inventory problem

René Haijema[1], Jan van der Wal[1,2], and Nico M. van Dijk[1]

[1] Department of Quantitative Economics, University of Amsterdam, The Netherlands
[2] Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands

Abstract. Blood banks produce and store blood products in order to fulfil the uncertain demand at hospitals. Platelet pools are the most expensive and most perishable blood product having a shelf life of only four to six days. Production volumes need to be chosen carefully in order to reduce outdating while keeping the occurrence of shortages low.

We investigate the structure of the optimal production policy by solving a down sized periodic Markov Decision Problem. The optimal production volumes appear to depend on the number of pools on stock and their ages. Simulation results for the optimal MDP-policy suggest two rules: the $1D$ and $2D$ rule. Both rules perform quite well. The $2D$ rule performs nearly optimal even if one acknowledges the distinction of multiple and limited compatible blood groups and the uncertainty in the supply by donors.

1 Problem setting

Blood banks collect blood and produce blood products from it. From one whole blood donation, of 500 *ml* by a single donor, most of the red blood cells and plasma are filtered. The residual called, the *'buffy coat'*, contains a high concentration of blood platelets. The platelets of five buffy coats are distilled and pooled into a so-called platelet pool. These pools are delivered to hospitals on demand at least once a day (in the morning).

Blood banks are interested in setting (near-)optimal production volumes of platelet pools since it is the most expensive and most perishable blood product. The price of a platelet pool sold to a hospital is in the Netherlands 458 euro. The maximal lifetime of a platelet pool is set at 5 days at most blood banks, but current developments allow raising this maximum to 7 days. Since collection, production, and laboratory tests together take one day the effective shelf life of a platelet pool is 4 to 6 days. All other blood products can be kept on stock for weeks or even months.

Platelets are of live saving importance. As small particles inside the bloodstream platelets prevent bleeding by recognizing and 'repairing' damaged blood vessels. Platelets deteriorate rapidly in quality even inside the blood stream, but most people's platelets production at the bone marrow is sufficient to retain a safe level. Nevertheless after a major bleeding caused by a trauma or a surgery, patients may temporarily have a lack of platelets. These patients need to be transfused with platelet pools of *any* age up to the maximal shelf life. This demand category is called *'any'*, since there is no strong preference with respect to the age of the pools. The *'any'* demand comprises 30% of the total demand of about 176 pools per week per Dutch blood bank.

The remaining 70% of the demand is for patients suffering from a platelet function disorder. They need to be transfused at least once a week with *'fresh'* pools that are stocked for at most 3 days. Transfusing older platelet pools is less desirable. Although part of the transfusions at the hospitals are scheduled, a considerable amount of demand is unknown to blood banks.

At the supply side enough buffy coats are available. In the Netherlands only a third to a half of the buffy coats is used for processing platelet pools depending on the occurrence of donors. So it seems reasonable to ignore the uncertainty in the supply of buffy coats.

Another complication is the distinction of eight different blood groups based on four $AB0$-categories and the Rhesus-D factor ($RhD+$ or $RhD-$). Figure 1 shows the compatibility of the different blood groups and the frequencies by which they occur in the Western population. In section 5 we deliberate on the compatibility of the blood groups and also on the stochastic supply of buffy coats.

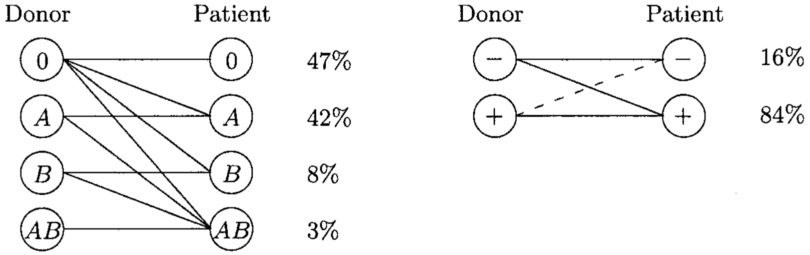


Fig. 1. Compatibility of blood groups of donor and patient following the $AB0$−system and Rhesus-D system (+/−)

We distinguish four types of linear costs:

- *shortage* costs: for demand that cannot be fulfilled from stock
- *mismatch* costs: for demand for a 'fresh' pool that is met by an older pool
- *outdating* costs: for a pool that has to be disposed because of its age
- *holding* costs: for keeping a pool on stock during a day.

Under this linear cost structure we are interested in (near-)optimal production volumes that minimize the *long-run average weekly cost*, taking into account:

- *two types* of 'periodic' demand
- the limited shelf life of the pools

- the production stop during weekends
- the uncertainty in the supply of buffy coats

and

- the distinction of *multiple* and *limited compatible blood groups.*

2 Literature and Main results

The work on inventory management of perishables started by Van Zyl, Nahmias, Cohen and Pierskalla, amongst many others. We refer to [4], [5] and [2] for an overview of the literature in this field. Most of the blood banking applications report on inventory models for Red Blood Cell Concentrates ($RBC's$). From these studies we learn that both outdating and shortages can be kept low by setting the ordering or production quantities according to an order-up-to rule. Most of the perishable inventory models are simulation models or based on the Markov Decision Problem (MDP) under very limiting assumptions to circumvent the computational complexity of the problem.

Studies on platelet inventory management show that hospitals can reduce significantly outdating by ordering more carefully. In the Netherlands outdating at the hospitals is reduced by stocking the pools at the blood banks. Still about 15 to 20% of the platelet pools remain unused due to outdating. This figure is reported in many other studies on platelet inventory management and seems to apply for many blood banks or hospitals in the USA and Europe.

The recent trend of raising the shelf life of platelet pools makes it desirable to distinguish demand for *'fresh'* pools and demand with less preference with respect to the age of the pools. This together with the periodicity in demand, the production stop during weekends, the production lead time, the distinction of multiple limited compatible blood groups and the uncertainty at the supply side, was not studied before.

Main results. Although the modelling of the production problem as an MDP is not new, our study seems to be the first that reports on the optimal policy obtained by solving a realistic MDP. The structure of the optimal solution suggests, what we call, a $2D$ replenishment rule. The $2D$ rule is in most of the cases less than 1% away from the optimal cost level and reduces the outdating to less than 1% of the production. Shortages and mismatches appear to occur almost never, even with all of the above mentioned complications.

3 Markov Decision Problem

The platelet production problem can be modelled as a Markov Decision Problem (MDP) minimizing the long-run average weekly cost. Relevant cost components are the holding, shortage, mismatching and outdating cost. In order to formulate the *MDP* let us define the decision epochs, the decisions, the states, the transitions and associated probabilities. For the moment we ignore the distinction of blood groups and we assume an unlimited supply of buffy coats. In section 5 we will put these assumptions to the test.

***MDP*-formulation.** At the start of each working day a decision is taken on the number (k) of platelet pools to produce. This decision is based on the day of the week (d) and the number of pools on stock ($\mathbf{x}$). If $d = 1, 2, ..., 7$ represent successively Monday till Sunday, then k must be 0 at $d = 6$ and 7.

The stock state $\mathbf{x}$ is a vector with the number of pools of each age category as its elements: $\mathbf{x} = (x_1, x_2, ..., x_m)$, with m is the maximal shelf life in days and x_r is the number of pools with r days residual shelf life.

The state transition from one day to the next, so from $(d, \mathbf{x})$ to $(d+1, \mathbf{y})$, depends on the demands for 'fresh' (j) and for 'any' (i) pools during that day, the production k and the issuing policy R. We write $\mathbf{y} = \mathbf{y}^R(\mathbf{x}, i, j, k)$.

The demand distributions for 'fresh' and 'any' pools are day dependent and denoted by $p_d^f(j)$ and $p_d^a(i)$. So the *MDP* is periodic. For simplicity we model both types of demand as Poisson processes.

In most perishable inventory models issuing the oldest items first minimizes the long run average cost, since this reduces outdating. One often refers to this policy as a First-In-First-Out (*FIFO*) policy. Since we consider two types of demand, R is a combination of two issuing rules. The demand for 'any' pools is met by issuing the oldest pools first (*FIFO*). The demand for 'fresh' pools is fulfilled by issuing the youngest pools on stock: a so-called Last-In-First-Out policy (*LIFO*). Within our discrete time model we treat the demand for fresh pools prior to the demand for pools of any age.

Let $c^R(d, \mathbf{x}, k)$ denote the expected cost on day d when having $\mathbf{x}$ on stock at the beginning of the day and producing k pools. $c^R(d, \mathbf{x}, k)$ is the sum of the holding, shortage, mismatch and outdating cost. Let $\mathbf{V}_n^R(d, \mathbf{x})$ be the minimum expected cost over an horizon of n days, starting at day d with initial stock $\mathbf{x}$. Setting $\mathbf{V}_0^R = 0$, one can compute iteratively $\mathbf{V}_n^R(d, \mathbf{x})$ by:

$$\mathbf{V}_n^R(d, \mathbf{x}) = \min_k \left\{ c^R(d, \mathbf{x}, k) + \sum_{i,j} p_d^f(j) p_d^a(i) \mathbf{V}_{n-1}^R(d+1, \mathbf{y}^R(\mathbf{x}, k, i, j)) \right\}$$

When n tends to infinity $\mathbf{V}_{n+7}^R(d, \mathbf{x}) - \mathbf{V}_n^R(d, \mathbf{x})$ converges to the minimal average weekly cost and the minimizing actions construct the optimal *MDP*-policy (cf. [3] and [6]).

In the next section we show some numerical results and we investigate the structure of the optimal policy. Let us first deliberate a bit more on the complexity of solving the Markov Decision Problem.

Complexity. Within each iteration all possible state transitions are considered for each pair (state, decision)= $(d, \mathbf{x}, k)$. Due to the number of states this iterative process is very time consuming: in our case with a maximal shelf life of 6 days each iteration would require about 10^{13} computations.

Scaling the demand distribution with a factor 4 reduces the computational work to a reasonable level. In the original setting of the problem we have assumed Poisson demands, now we fit new demand distributions, such that the means and standard deviations are the original ones divided by 4 (cf. [1]). The (variable) costs are multiplied by 4.

4 Optimal policy and Simple rules

Since the *MDP* cannot be solved for real-sized instances, we like to derive simple rules that closely resemble the optimal policy. The optimal policy obtained by solving the *MDP* of the down sized problem with a shelf life of 6 days is a benchmark for simple decision rules. All results we report here are based on realistic estimates of the parameters for a Dutch blood bank. We have set the following cost figures per pool, which are related to the variable production cost of about 150 euro per pool:

- shortage costs: 750
- mismatch costs: 200
- outdating costs: 150
- holding costs: 1 per day.

In [3] is shown that the structure of the optimal policy is not very sensitive to the actual numbers other than major changes in their ratios.

Simulation table. By simulating the optimal *MDP*-policy we constructed frequency tables that visualize the structure of the optimal production policy. For example, consider table 1 that shows the aggregated stock states and implied replenishment levels on Tuesday based on a simulation run of the optimal policy for 100,000 weeks.

The first row shows the total number of pools on stock minus the expected demand and the expected outdating: the so-called expected final stock level. The last row shows how many times each expected final stock level occurs during the simulation. The first column shows the replenishment levels, obtained by adding the optimal production volume to the expected final stock level. The last column shows that a replenishment level of 15 occurs in 74.8% of the Tuesdays within the simulation. The other numbers in the table denote the frequencies by which each combination of replenishment level and stock level occurred during the simulation, e.g. the combination (15, 5) occurs on 23,724 out of the 100,000 simulated Tuesdays. The associated production is 15-5=10.

Apparently a simple rule with a fixed order-up-to level of 15 closely approximates the optimal production policy. Often the optimal production volume is higher than this rule would imply. By looking at the stock state in more detail, it appears that higher replenishment levels are related to states in which relatively many 'old' pools are on stock. This suggests to consider

Table 1. Frequency table of stock and replenishment level combinations occurring on Tuesday for the optimal *MDP*-policy. Frequencies are obtained from a simulation of 100,000 weeks.

S ↓ stock →	2	3	4	5	6	7	8	9	10	11	12	13	14	Cum. freq.
23													4	4
22												22		22
21											86			86
20									1	248				249
19								6	703	3				712
18							34	1818	31	1				1884
17						6257	3703	177	25	1				10163
16					8292	2116	1504	175	1					12088
15	3152	14168	21030	23724	10082	2590	39	7						74792
Cum. freq.	3152	14168	21030	23724	18374	10963	5280	2183	761	253	86	22	4	100000

a replenishment rule that takes into account both the total number of pools on stock and their age.

1*D* and 2*D* rule. From the structure of the optimal *MDP*-solution we derive two simple near-optimal decision rules, which we refer to as the 1*D* rule and 2*D* rule.

- The 1*D* rule consists of a replenishment level S_d^t for each working day d. The production volume k is derived from S_d^t by subtracting the expected number of pools left the next morning prior to replenishment, which is in fact the total expected final stock prior to replenishment but after disposing off the outdated pools (y^t).
- The 2*D* rule consists of two replenishment levels S_d^t and S_d^f for each working day. Production volumes are set to $\max\{S_d^t - y^t, S_d^f - y^f\}$, where y^f is the expected number of fresh pools left the next morning prior to replenishment.

S_d^t and S_d^f can be seen as target-inventory levels for the next morning.

A local search procedure is used for optimizing the replenishment levels for fresh and total stock. The local search is based on simulation and steepest decent search. Table 2 shows that the replenishment rules for the down sized problem obtained this way perform nearly optimal for varying values for the maximal shelf life. We conclude that raising the shelf life from 4 to 6 days results in considerable cost savings, since outdating becomes then less a problem.

Table 3 presents the results for the real sized problem with maximal life time 6 days. The 95%-Confidence Interval (*CI*) is based on 10 simulation runs of 100,000 weeks each. Compared to the down sized problem the cost levels are a bit higher, due to the scaling of the problem. The fitting technique resulted in demand distributions that are somewhat skewed to the left. In [3] one finds more results that illustrate the superior quality of the 2*D* rule.

Table 2. Comparison of the optimal *MDP*-policy and the (local) optimal $1D$ and $2D$ rules for the down sized problem. The exact average weekly costs are obtained by solving the associated Markov Chains.

Shelf life	Opt. policy	$1D$-rule	$2D$-rule
4	1680	+3.2%	+1.8%
5	1034	+2.3%	+0.6%
6	692	+3.3%	+0.5%

Table 3. Comparison of the $1D$ rule and $2D$ rule by ten simulation runs of the real sized problem for 100,000 weeks (maximum shelf life = 6 days).

Local search results	$1D$-rule	$2D$-rule
95% *CI* weekly cost	$[788.7 \pm 2.1]$	$[774.9 \pm 2.3]$
S^t	(48,64,47,53,72)	(44,63,47,52,72)
S^f	-	(27,49,31,38,36)

Outdating and shortages. Based on the simulations we conclude that both outdating and the occurrence of shortages can be kept at a low level. Applying the rules reduces outdating to 1%, the shortage rate to 0.1% and mismatches occur almost never.

Conclusion. By a combined *MDP* and simulation approach it is shown that near-optimal simple replenishment rules can be found: the so-called $1D$ and $2D$ rules. As opposed to substantial outdating in practice (in the order of 20%) these rules are promising: outdating can be reduced significantly.

Remark. One could simplify the rules somewhat by aggregating the present stock instead of the expected final stock. Although outdating occurs not frequently, one still has to account for the ageing process, e.g. some pools that are today 'fresh' might be too old to meet tomorrow's 'fresh' demand.

5 Multiple blood groups and limited supply

Thus far we ignored the eight different blood groups and the stochastic supply of buffy coats. Incorporating the blood groups in the *MDP* would dramatically blow up the state space. (With a shelf life of 6 days and 8 different blood groups the state space would have $1 + 6 \cdot 8 = 49$ dimensions.) In this section we show that one can closely approximate the minimal cost level by adopting the production volumes from the aggregated model.

In practice blood banks produce pools only from donor groups 0 and A, since compatible matching is the standard. Therefore we limit our attention to the supply of buffy coats of types 0 and A. In figure 1, we read that about

90% of the Western population has either blood group 0 or A. With respect to the Rhesus-D factor we note that not even 50% of the $RhD-$ patients requires strict matching. This is only 8% of all demand.

Based on blood bank data we estimate that of all available buffy coats on average 10% is of type 0−, 50% is of type 0+, 10% of type $A-$ and 30% of type $A+$. The supply of buffy coats is stochastic as is the appearance of donors. We model the supply of buffy coats by Poisson processes. Further we assume that the supply on Friday is 10% higher than on the other working days. During the weekends the supply is zero.

We have developed a multiple-group simulation program that evaluates the $2D$ rule in a setting where one distinguishes blood groups and acknowledges the uncertainty in the supply of buffy coats. The required aggregated production volume suggested by the $2D$ rule is reached by producing as many pools of the *'most compatible buffycoats'* in the order 0−, 0+, $A-$, $A+$. Fulfilling the demand for 'fresh' prior to the demand for 'any' pools one first matches the demand of the *'least compatible patients'* in the order 0−, $B-$, $A-$, $AB-$, 0+, $B+$, $A+$, $AB+$. The demand set by each patient group is fulfilled by issuing the *'least compatible pools'* first ($A+$, $A-$, 0+, 0−). This way one keeps the occurrence of shortages and outdating low.

Table 4 reports on the (relative) production volumes of each blood group as well as the average weekly cost for the multiple-group model. When only a third of the buffy coats is used for platelet production it suffices to produce only from 0− and 0+ buffy coats. When buffy coats are more scarce some pools of blood group A are produced. The average weekly cost should be compared with 774.9 for the single-group model reported in table 3.

Table 4. Average weekly costs and the distribution of the production of the different blood groups for varying fractions of used buffy coats. The results are obtained from 10 runs of 100,000 weeks of the multi-group simulation model.

% buffy coats	Costs	0−	0+	$A-$	$A+$
33%	[775.8 ± 3.1]	28.8%	71.2%	0%	0%
40%	[777.4 ± 2.9]	23.7%	75.7%	0.6%	0%
50%	[779.8 ± 3.1]	18.7%	75.3%	4.3%	1.7%

Conclusion. The results show that the replenishment rules found by the single-group model perform almost as well as those found by the multiple-group model. Hence, when searching for good replenishment levels, one can ignore the existence of blood groups and the uncertainty at the supply side. This observation is of practical importance: not only to simplify the search for good replenishment levels but also to simplify the determination of production volumes based on the replenishment rules.

6 Evaluation

Finding optimal production volumes for blood banks is complicated due to the *'curse of dimensionality'* of the MDP. It has been shown that two major complicating factors of the platelet production problem

- the distinction of blood groups and
- the uncertainty in the supply of buffy coats

can be relaxed. As a consequence we can investigate the structure of the optimal policy by solving a simplified and down sized but still realistic MDP.

From a simulation of the optimal MDP-policy frequency tables of stock and optimal replenishment levels are obtained, which illustrate that a simple $1D$ rule approximates well the optimal policy. It also suggests a $2D$ rule that performs even better.

It appears that the optimal production volumes depend on both the number of pools on stock and their ages. As both rules are easy to use in practice, implementation and extensions seem of practical interest.

Acknowledgement

We are most grateful to professor dr. Cees Th. Smit Sibinga, Director of Sanquin Consulting Services and advisor of the WHO Collaborating Center for Blood Transfusion and the WFH International Hemophilia Training Centre, for fruitful discussions and insights into the problem. We also wish to acknowledge Ing. Boudewijn Hinloopen, head Processing, and Minze Broersma, manager Logistics, (both of Sanquin, region North-East, The Netherlands), for providing the necessary data and for giving us a better understanding of the production and distribution process of platelet pools.

References

1. Adan, I.J.B.F., M.J.A. van Eenige and J.A.C. Resing (1995), Fitting discrete distributions on the first two moments, *Probability in the Engineering and Informational Sciences*, **9**, pp. 23-632.
2. Goyal, S.K. and B.C. Giri (2001), Recent trends in modeling of deteriorating inventory, *European Journal of Operational Research* **134** **(1)**, pp. 1-16.
3. Haijema, R., J. van der Wal and N.M. van Dijk, Blood Platelet Production: Optimization by Dynamic Programming and Simulation, *to appear in Computers and Operations Research, special issue on OR and Health Care.*
4. Nahmias, S. (1982), Perishable Inventory Theory: A Review, *Operations Research*, **30**, pp. 680-708.
5. Prastacos, G.P. (1984), Blood Inventory Management: an Overview of Theory and Practice, *Management Science*, **30** **(7)**, pp. 777-800
6. Su, S. and R. Deininger (1972), Generalization of White's method of successive approximations to periodic Markovian decision processes, *Operations Research*, **20**, pp. 318-326.

Zeitdiskrete Modellierung der Wechselwirkungen der Plan-Vorgaben bei Verwendung der Liefertreue als Leistungsgröße für die interne Supply Chain in der Halbleiterindustrie

Kirsten Hilsenbeck[1], Alexander Schömig[2], Walter Hansch[1]

[1] Fachgebiet Halbleiterproduktionstechnik, Lehrstuhl für Technische Elektronik, Technische Universität München, Arcisstr. 21, D-80333 München, Germany, Kirsten.Hilsenbeck@tum.de, Hansch@tum.de
[2] Corporate Logic, Infineon Technologies AG, D-81730 München, Germany, Alexander.Schoemig@infineon.com

Abstract. Die Steuerung der Fertigung von Chips in der Halbleiterindustrie wird aus Vorgabewerten von so genannten Performance-Indikatoren abgeleitet. Kennzahlen der Liefertreue werden dabei als Leistungsgrößen und Plan-Vorgaben verwendet. Um die Wechselwirkungen der Plan-Vorgaben in der internen Supply Chan einer Halbleiterfertigung zu untersuchen, wurde ein zeitdiskretes Modell entwickelt. Die Ergebnisse zeigen, dass sich die einzelnen Leistungs-Vorgaben gegenseitig beeinflussen und somit nicht unabhängig festgelegt werden können.

1 Einleitung

Die Mikroelektronik spielt im täglichen Leben eine zunehmend wichtigere Rolle. Neben Anwendungen in Computern und Telekommunikationsgeräten sind Mikrochips mittlerweile in Dingen des allgemeinen täglichen Gebrauchs wie Haushaltsgeräten oder Kraftfahrzeugen in großer Anzahl zu finden und praktisch unverzichtbar geworden. Für das Jahr 2004 erwartet man laut *World Semiconductor Trade Statistics (WSTS)* ein weltweites Marktvolumen in der gesamten Halbleiterbranche von ca. 213 Milliarden US-Dollar bei einem Wachstum von 28,4 Prozent gegenüber 2003. In der gegenwärtigen Aufschwungsphase sind Verbesserungen hinsichtlich der Steuerung und Kontrolle der firmeninternen und übergreifenden Wertschöpfungskette von hohem Interesse (Levine 2004). Abbildung 1 zeigt im Überblick die Struktur der internen Wertschöpfungskette eines typischen Halbleiterchipherstellers. Die Gesamtausbeute an guten Chips liegt auch bei reifen Technologien teilweise unter 100%. Die gesamte Durchlaufzeit eines Auftrages kann bei sehr komplexen Produkten bis zu vier Monate betragen.

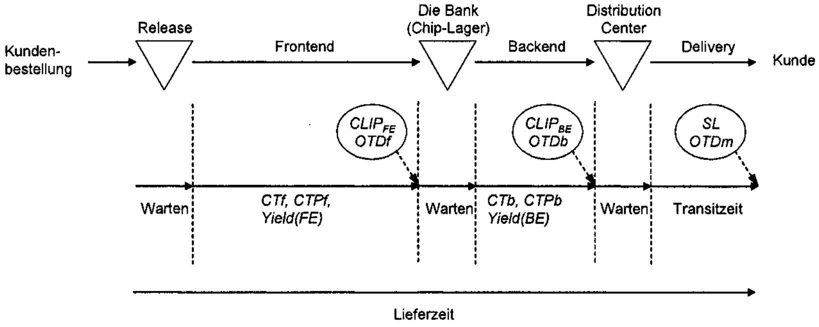


Abb. 1. Die interne Supply Chain besteht aus drei Stufen (Frontend, Backend, Delivery) sowie zwei Zwischenlager (Die Bank, Distribution Center)

Die größte Herausforderung bei der Steuerung und Leistungsbewertung der internen Wertschöpfungskette liegt in deren Struktur und mit weltweiten Fertigungsstätten, Beziehungen mit Zulieferer und Verflechtungen mit ausgelagerten, externen Fertigungsschritten (Frederix 1996, Schömig u. Fowler 2000) begründet, sowie in der Heterogenität dieser Knotenpunkte im Materialfluss. Bei den meisten Halbleiterchipherstellern erhalten das Frontend wie das Backend wöchentlich basierte Mengenvorgaben je Chiptyp, basierend auf einen MRP Planungslauf (Schneeweiss u. Wetterauer 2000) unter Berücksichtigung von festen Kundenaufträgen, Bedarfsprognosen und Lagerwiederbefüllungen. Die Feinterminierung der Einlastung der Fertigungsaufträge (*„Einschleusung“*) obliegt der Fertigung, die in der Regel eine gleichmäßige Verteilung anstrebt. Die Durchsteuerung durch das Frontend bzw. das Backend erfolgt dann relativ losgelöst vom übergeordneten Planungssystem, mit täglichen, automatischen Rückmeldungen hinsichtlich des Fertigungsfortschritts. Die der Planung zugrunde liegenden Fertigungsauftragsdurchlaufzeiten sind den tatsächlich realisierten angepasst.

Diese Werte dienen dann auch zur Leistungsbewertung des Gesamtsystems. Zu den wichtigsten Leistungsgrößen zählen dabei die Produktionsausbeute, die Liefertreue gegenüber den Kunden, sowie die interne Durchlaufzeit. Eine weitere, in der Halbleiterindustrie wichtige und gerne benutzte Leistungsgröße, ist die sog. *Confirmed Line Item Performance (CLIP)*. Gemäß ihrer Definition (s. Gl. 1) ist diese Größe gleichzeitig Mengen und Termin bezogen (im Beispiel auf die Woche t, Fehlmengen werden über die Zeit akkumuliert).

$$CLIP_{BNR}(t) = \min\left[\frac{Liefermenge_{BNR}(t)}{Bestellmenge_{BNR}(t) + Fehlmenge_{BNR}(t-1)}, 1\right] \cdot 100\% \tag{1}$$

Sie wird für das Frontend, das Backend und mittelbar für den Kundenendtermin in Gestalt eines Zielwertes vorgegeben und ermittelt. Das Erreichen der Vorgaben hängt dabei offensichtlich von Faktoren wie der Durchlaufzeit, der Streuung der Durchlaufzeit, der Plandurchlaufzeit, dem

Einschleusmodus und der Zwischenlagerstrategie ab. Diese einzelnen Leistungsvorgaben beeinflussen sich gegenseitig und können im Grunde nicht unabhängig festgelegt werden. Mangels einfacher Berechnungsmöglichkeiten werden in der Praxis diese Abhängigkeiten vernachlässigt, so dass sich inkonsistente Zielvorgaben ergeben können.

Das im Folgenden vorgestellte Rechenverfahren zeigt, wie über mehrere Fertigungsstufen hinweg konsistente Zielvorgaben für den CLIP abhängig von Soll und Ist-Durchlaufzeiten, sowie deren Streuung berechnet werden können. Wir verwenden ein stochastisches, zeitdiskretes Berechnungsmodell, wie es in (Tran-Gia 1996) beschrieben wird und erfolgreich zur Modellierung von Lagerhaltungspolitiken (Tempelmeier 2003) oder komplexen Warteschlangensystemen (Dümmler u. Schömig 1999) verwendet wurde. Die an sich kontinuierlich Zeitgröße wird mit der Konstante Δt diskretisiert. Für Berechnungen hat sich eine Diskretisierungskonstante von $\Delta t = 0{,}001$ Tage als sinnvoll erwiesen.

2 Modell

2.1 Frontend

Die Einschleusung ins Frontend wird durch die Zufallsvariable Rf (Release Frontend) mit gleicher Wahrscheinlichkeit der Einschleusung zu jedem diskreten Zeitpunkt der Einschleuswoche beschrieben. Ihre Wahrscheinlichkeitsfunktion ergibt sich zu:

$$f_{Rf}(k) = \begin{cases} \dfrac{1}{7\,\mathrm{d}/\Delta t} & , \quad 1 \le k < 7\,\mathrm{d}/\Delta t \\ 0 & , \quad k \ge 7\,\mathrm{d}/\Delta t \end{cases} \qquad \text{mit } k = 1, 2, 3, \ldots \tag{2}$$

Die Durchlaufzeit der Lose eines Produktes im Frontend CTf (Cycle Time Frontend) wird als normalverteilt angenommen mit einem Mittelwert μf und einer Standardabweichung σf. Ihre Wahrscheinlichkeitsfunktion ergibt sich somit zu (3). Dabei steht $g(t)$ für die Dichtefunktion der Gaußschen Normalverteilung und $G(t)$ für ihre Verteilungsfunktion. Die Zufallsvariable Fertigstellung Frontend Cf (Completion Frontend) ergibt sich aus der Addition der Zufallsvariablen für den Zeitpunkt der Einschleusung Rf und der Durchlaufzeit CTf. Ihre Wahrscheinlichkeitsfunktion berechnet sich somit aus der Faltung der beiden Wahrscheinlichkeitsfunktionen.

$$\begin{aligned} f_{CTf}(k) &= \int_{(k-1)\cdot\Delta t}^{k\cdot\Delta t} g(t) \\ &= G(k\cdot\Delta t) - G((k-1)\cdot\Delta t) \end{aligned} \tag{3}$$

$$Cf = Rf + CTf \tag{4}$$

$$f_{Cf}(k) = f_{Rf}(k) \otimes f_{CTf}(k) \tag{5}$$

Der Stichtag für den Liefertermin *DDf* (Due Date Frontend) ergibt sich aus der Plandurchlaufzeit *CTPf* (Cycle Time Plan Frontend) plus sieben Tage aufgrund der siebentägigen Einschleuswoche. Die Plandurchlaufzeit wird in Wochen angegeben, d.h. ihr Wert ist immer ein Vielfaches von sieben Tagen. Die Wahrscheinlichkeit für das rechtzeitige Erreichen der Die Bank, also die On-Time-Delivery des Frontends *OTDf* kann in der Verteilungsfunktion der Fertigstellung Frontend am Zeitpunkt des Stichtages abgelesen werden.

$$DDf = CTPf + 7\,\mathrm{d} \tag{6}$$

$$OTDf = P(Cf \leq DDf) = \frac{F_{Cf}(k = DDf \,/\, \Delta t)}{F_{Cf}(k_{\max})} \tag{7}$$

2.2 Backend

Der $CLIP_{BE}$ bewertet die Performance des Backends unabhängig von der Vorgeschichte der Produktion. D.h. alle Chips, die nicht rechtzeitig in der Die Bank waren, werden aus der bestellten Menge zur Berechnung des $CLIP_{BE}$ herausgenommen. Zur Ermittlung der Performance des Backends (Backend Only) werden deshalb gleichmäßig verteilte Wahrscheinlichkeiten der Einschleusung innerhalb der Einschleuswoche verwendet.

$$f_{Rbo}(k) = \begin{cases} \dfrac{1}{7\,\mathrm{d}/\Delta t} & , \quad 1 \leq k < 7\,\mathrm{d}/\Delta t \\ 0 & , \quad k \geq 7\,\mathrm{d}/\Delta t \end{cases} \qquad \text{mit } k = 1, 2, 3, \ldots \tag{8}$$

Der Stichtag *DDb* zur Ermitlung der On-Time-Delivery Backend ergibt sich aus der Plandurchlaufzeit des Backends plus sieben Tage. Die Plandurchlaufzeit des Backends beträgt wie die Plandurchlaufzeit des Frontends immer ein Vielfaches von sieben Tagen. Die Durchlaufzeit der Lose eines Produktes im Backend *CTb* wird wiederum als normalverteilt angenommen mit einem Mittelwert μb und einer Standardabweichung σb.

$$DDb = CTPb + 7\,\mathrm{d} \tag{9}$$

$$\begin{aligned} f_{CTb}(k) &= \int\limits_{(k-1)\cdot\Delta t}^{k\cdot\Delta t} g(t) \\ &= G(k \cdot \Delta t) - G((k-1) \cdot \Delta t) \end{aligned} \tag{10}$$

Die Wahrscheinlichkeitsfunktion der Fertigstellung für die alleinige Betrachtung des Backends (Backend Only) ergibt sich aus der Faltung der optimalen Einschleusung und der Durchlaufzeit. Die On-Time-Delivery Backend ist der Funktionswert der Verteilungsfunktion der Fertigstelung Backend Only für den Zeitpunkt der Fälligkeit Backend *DDb*.

$$f_{Cbo}(k) = f_{Rbo}(k) \otimes f_{CTb}(k) \tag{11}$$

$$OTDb = P(Cbo \leq DDb) = \frac{F_{Cbo}(k = DDb / \Delta t)}{F_{Cbo}(k_{\max})} \tag{12}$$

Zur Ermittlung der Liefertreue des Backends bezogen auf das Gesamtsystem muss zunächst die Wahrscheinlichkeitsfunktion der Einschleusung mit Berücksichtigung des Vorgeschehens ermittelt werden. Die Einschleusung in das Backend erfolgt nach der planmäßigen Lieferwoche des Frontends. Verspätete Bestellungen werden im Backend bevorzugt weiterverarbeitet. Der Zeitpunkt der Einschleusung in das Backend liegt also immer nach dem Fälligkeitstermin des Frontends.

$$Rb > DDf \tag{13}$$

Für die Wahrscheinlichkeitsverteilung der Einschleusung ins Backend gilt folgendes: Ist ein Los im Frontend rechtzeitig fertig gestellt worden, so hat es innerhalb der planmäßigen Einschleuswoche des Backends eine für jeden Zeitpunkt gleich große Wahrscheinlichkeit eingeschleust zu werden. Ist ein Los im Frontend nicht rechtzeitig fertig gestellt worden, seine Lieferung fällt aber noch in die Einschleuswoche, so hat es für die restlichen Tage der Einschleuswoche eine gleichverteilte Einschleuswahrscheinlickeit. Lose, die erst nach Ablauf der Einschleuswoche des Backends vom Frontend fertig gestellt werden, werden sofort in das Backend eingeschleust und weiter verarbeitet.

Um die Wahrscheinlichkeitsfunktion der Einschleusung ins Backend ermitteln zu können, wird zunächst die gestutzte Wahrscheinlichkeitsfunktion der Fertigstellung Frontend benötigt. Dazu wird der zeitdiskrete Operator der Stutzung (Tran-Gia u. Schömig 1996) verwendet (14). Der Stutzungs-Operator π_m addiert alle Werte der Wahrscheinlichkeitsfunktion für Variablenwerte kleiner oder gleich *m* und weist sie als Funktionswert genau diesen Variablenwertes zu. Die Wahrscheinlichkeitswerte aller Variablenwerte kleiner als *m* werden zu null gesetzt, die Wahrscheinlichkeitswerte aller Variablenwerte größer als *m* behalten ihren ursprünglichen Wert.

$$\pi_m(x(k)) = \begin{cases} 0 & , \quad k < m \\ \sum_{l=-\infty}^{m} a(l) & , \quad k = m \\ x(k) & , \quad k > m \end{cases} \tag{14}$$

Die auf das erste diskrete Zeitintervall der Einschleuswoche des Backends gestutzte Wahrscheinlichkeitsfunktion der Fertigstellung Frontend wird somit durch folgenden Ausdruck beschrieben:

$$\pi_{\frac{DDf}{\Delta t}+1}\left(f_{Cf}(k)\right) \tag{15}$$

Ausgegangen wird von der gestutzten Wahrscheinlichkeitsfunktion der Fertigstellung Frontend. Sie gibt die Wahrscheinlichkeit an, an welchem Tag ein Los frühestens zur Einschleusung ins Backend bereit steht.

Die Wahrscheinlichkeitsfunktion für die Einschleusung in das Backend ergibt sich aus der Summe der Einzelfunktionen, die mit dem jeweiligen Wert der gestutzten Wahrscheinlichkeitsfunktion der Fertigung Frontend multipliziert werden. Für die Zeit nach der Einschleuswoche des Backends sind die Wahrscheinlichkeitswerte der Fertigstellung Frontend und der Einschleusung Backend identisch. Mathematisch wird dieser Sachverhalt in folgender Gleichung beschrieben.

$$f_{Rb}(k)=\begin{cases} 0 & , \quad k \le \frac{DDf}{\Delta t} \\ \sum\limits_{i=\frac{DDf}{\Delta t}+1}^{k} \pi_{\frac{DDf}{\Delta t}+1}\left(f_{Cf}(i)\right)\cdot\frac{1}{\frac{7\,\mathrm{d}}{\Delta t}-i+\frac{DDf}{\Delta \mathrm{t}}+1} & , \quad \frac{DDf}{\Delta t} < k \le \frac{DDf+7\,\mathrm{d}}{\Delta t} \\ f_{Cf}(k) & , \quad k > \frac{DDf+7\,\mathrm{d}}{\Delta t} \end{cases} \tag{16}$$

Die Durchlaufzeit der Lose eines Produktes im Backend *CTb* wird wiederum als normalverteilt angenommen mit einem Mittelwert μb und einer Standardabweichung σb.

$$\begin{aligned} f_{CTb}(k) &= \int_{(k-1)\cdot\Delta t}^{k\cdot\Delta t} g(t) \\ &= G(k\cdot\Delta t)-G((k-1)\cdot\Delta t) \end{aligned} \tag{17}$$

Die Zufallsvariable Fertigstellung Backend *Cb* ergibt sich wiederum aus der Addition der Zufallsvariablen Einschleusung Backend *Rb* und Durchlaufzeit Backend *CTb*, womit sich die Wahrscheinlichkeitsfunktion aus der Faltung der beiden Wahrscheinlichkeitsfunktionen ergibt.

$$f_{Cb}(k)=f_{Rb}(k)\otimes f_{CTb}(k) \tag{18}$$

Die On-Time-Delivery Manufacturing wird am Ende der planmäßigen Lieferwoche des Backends ermittelt. Daraus ergibt sich der Stichtag der Lieferung. Er ergibt sich aus den Fälligkeitsterminen des Frontends und des Backends.

$$DDm = DDf + DDb \tag{19}$$

Die Wahrscheinlichkeit für das rechtzeitige Erreichen des Distribution Centers, die On-Time-Delivery Manufcturing kann in der Verteilungsfunktion der Fertigstellung Backend am Zeitpunkt des Stichtages *DDm* abgelesen werden.

$$OTDm = P(Cb \leq DDm) = \frac{F_{Cb}(k = DDm / \Delta t)}{F_{Cb}(k_{\max})} \tag{20}$$

3 Ergebnisse

Abbildung 2 zeigt die Wahrscheinlichkeitsfunktionen einer Bespielrechnung. Allgemein gilt: Die *OTDm* ist immer kleiner als die *OTDb*. Für eine *OTDf* = 1 ist die *OTDm* gleich der *OTDb*. Die Abhängigkeit der *OTDf* und der *OTDm* von μf ist gekoppelt mit der *CTPf*. Eine Änderung von μf wird durch die entsprechende gleiche Änderung von *CTPf* bei gleich bleibenden übrigen Werten kompensiert. Es erfolgt keine Veränderung der *OTD*-Werte. Das gleiche gilt für eine gleichmäßige Änderung von μb und *CTPb*. Der Einfluss der Standardabweichung der Durchlaufzeit im Frontend auf die *OTDm* ist je nach μf und *CTPf* unterschiedlich. Für einen gleichen Abstand zwischen *CTPf* und μf ergeben gleiche Standardabweichungen die gleichen *OTD*-Werte. Siehe dazu auch Abbildung 3.

Zu einem Wertepaar aus *OTDf* und *OTDb* können mehrere *OTDm* existieren. Eine eindeutige Zuordnung einer *OTDm* kann gegeben werden, wenn die Angaben je um eine zusätzliche Information über die Wahrscheinlichkeitsverteilung der Durchlaufzeit des Frontends und des Backends ergänzt werden. Dabei kann es sich wahlweise um die Standardabweichung der Durchlaufzeit oder die Differenz von Plandurchlaufzeit und Mittelwert der Durchlaufzeit handeln.

4 Zusammenfassung und Ausblick

Wir haben ein zeitdiskretes Berechnungsverfahren vorgestellt, mit dem die Zusammenhänge der Liefertreue innerhalb der Supply Chain über mehrere Stufen dargestellt werden können.

In weiteren Modellen kann gezeigt werden, dass alleine durch Veränderungen in den Steuergrößen und Modifikationen in der Einschleustechnik die Performance-Werte verbessert werden können. So können durch tagesfeine Liefertermine bei sonst gleichem Produktionsverhalten sowohl die Durchlaufzeit als auch die *OTDm* verbessert werden. Ein Wegfall der Einschleuswoche in das Backend erzielt weitere Verbesserungen für Durchlaufzeit und *OTDm*.

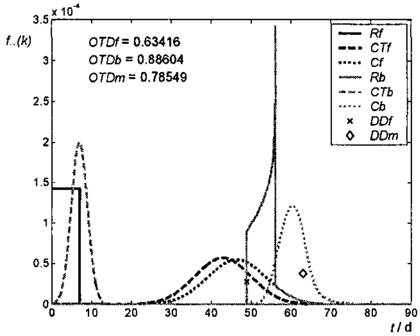


Abb. 2. On-Time-Deliveries für *CTf~N*(*μf* = 43 d, *σf* = 7 d); *CTPf* = 56 d; *CTb~N*(*μb* = 7 d, *σb* = 2 d); *CTPb* = 7 d

diff / σ	1	2	3	4	5	6	7	8	9	10	11	12
-5	0,287	0,30901	0,34203	0,36988	0,39084	0,40654	0,41855	0,42794	0,43546	0,44159	0,44668	0,45097
-4	0,4287	0,43458	0,44616	0,45596	0,46323	0,46861	0,4727	0,47588	0,47842	0,48048	0,48219	0,48363
-3	0,57145	0,56555	0,55394	0,54413	0,53685	0,53145	0,52736	0,52417	0,52163	0,51956	0,51784	0,5164
-2	0,71314	0,69111	0,65807	0,63021	0,60923	0,59352	0,58151	0,57211	0,56459	0,55845	0,55336	0,54906
-1	0,8453	0,80079	0,75187	0,7103	0,53685	0,65332	0,63416	0,61901	0,6068	0,59678	0,58844	0,5814
0	0,94304	0,88607	0,83048	0,78131	0,74127	0,70955	0,6844	0,66481	0,64957	0,63421	0,62292	0,61345
1	0,98811	0,94351	0,89157	0,84125	0,79742	0,7611	0,73145	0,70718	0,68713	0,6704	0,65625	0,64412
2	0,99879	0,97621	0,93542	0,88941	0,84562	0,80723	0,77469	0,74806	0,72644	0,70507	0,68864	0,67464
3	0,99995	0,99163	0,96434	0,92621	0,8856	0,84747	0,8137	0,78519	0,76153	0,73795	0,71953	0,70368
4	1	0,99758	0,98185	0,95292	0,91764	0,88173	0,84822	0,81896	0,79404	0,76884	0,74885	0,73147
5	1	0,99943	0,99151	0,97132	0,94244	0,91017	0,87822	0,84846	0,82156	0,79759	0,77633	0,75746
6	1	0,99989	0,99636	0,98335	0,96098	0,9332	0,90379	0,87592	0,85076	0,82409	0,80227	0,78283
7	1	0,99998	0,99858	0,99079	0,97436	0,95138	0,92518	0,89916	0,87487	0,84828	0,82619	0,80623
8	1	1	0,99949	0,99516	0,98368	0,96539	0,94275	0,91909	0,89618	0,87015	0,84819	0,82802
9	1	1	0,99984	0,99759	0,98994	0,9759	0,9569	0,93509	0,91229	0,88974	0,86811	0,84776
10	1	1	0,99995	0,99886	0,994	0,98359	0,96808	0,94993	0,93088	0,90711	0,88641	0,86673
11	1	1	0,99999	0,99949	0,99654	0,98908	0,97675	0,96143	0,94202	0,92237	0,90271	0,88366
12	1	1	1	0,99978	0,99807	0,9929	0,98335	0,97073	0,95618	0,93565	0,91723	0,899
13	1	1	1	0,99991	0,99896	0,99549	0,98828	0,97813	0,96583	0,94709	0,93006	0,91231
14	1	1	1	0,99997	0,99946	0,99721	0,9918	0,98394	0,97378	0,95686	0,94131	0,92519
15	1	1	1	0,99999	0,99973	0,99831	0,99449	0,98843	0,98025	0,96512	0,95108	0,93618

Abb. 3. Werte der einstufigen On-Time-Delivery in Abhängigkeit von der Differenz aus Plandurchlaufzeit und Mittelwert der Durchlaufzeit sowie der Standardabweichung der Durchlaufzeit

Literatur

Dümmler M, Schömig A (1999) Using Discrete-Time Analysis in the Performance Evaluation of Manufacturing Systems. In: Proceedings of the International Conference on Semiconductor Manufacturing Operational Modeling and Simulation 1999 (SMOMS '99). San Francisco, CA, U.S.A., pp 1-8

Frederix F (1996) Planning and Scheduling Multi-Site Semiconductor Production Chains: A Survey of Needs, Current Practices and Integration Issues. In: Manufacturing Partnerships: Delivering the Promise, pp 107-116

Levine B (2004) Today's Supply Chain Management Challenge: Keeping the Recovery On Track. Supplying the Stress during the Upturn. Semiconductor Manufacturing, Vol. 5, Issue 2, February 2004, pp. 39-52

Schneeweiss L, Wetterauer U (2000) Semiconductor Manufacturing. In: Stadtler H, Kilger C (eds) Supply Chain Management and Advanced Planning. Concepts, Models, Software and Case Studies. Springer, Berlin Heidelberg, pp 267-280

Schömig A, Fowler JW (2000) Modelling Semiconductor Manufacturing Operations. In: Proceedings of the 9th ASIM Dedicated Conference Simulation in Production und Logistics. Berlin, pp 56-64

Tempelmeier H (2003) Material-Logistik. Springer, Berlin, Heidelberg, New York

Tran-Gia P (1996) Analytische Leistungsbewertung verteilter Systeme. Springer, Berlin Heidelberg New York

Tran-Gia P, Schömig A (1996) Discrete-time analysis of batch servers with bounded idle time. In: Proceedings of the European Simulation Multiconference 1996. Budapest

Sequencing and lot-size optimisation of a production-and-inventory-system with multiple items using simulation and parallel genetic algorithm

Michael Kämpf[1] and Peter Köchel[1]

University of Technology, Department of Computer Science, Chemnitz D-09107, Germany

Abstract Our paper is dealing with the Capacitated Stochastic Lot-Sizing Problem. In addition to the usual model assumptions as stochastic demand and manufacturing times, cost for setup, we also consider cost for waiting and lost demand. The goal is to find release and sequencing decisions with minimal expected cost per time unit. To solve the problem we use simulation optimisation, i. e., we combine a simulator with a Parallel Genetic Algorithm. Some numerical examples show the applicability of the proposed approach.

1 Introduction

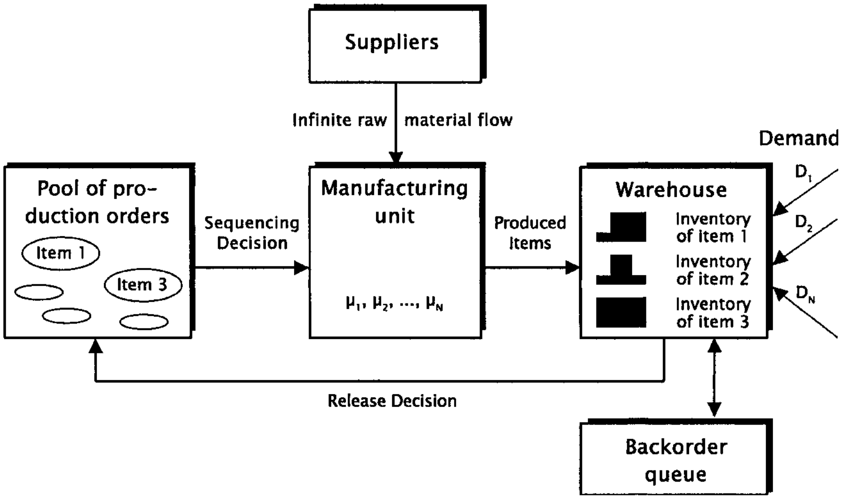
One of the most important problems for manufacturing firms is production lot-sizing and sequencing. At present different problem formulations exist: ELSP, Economic Lot Scheduling Problem; SELSP, Stochastic Economic Lot Scheduling Problem; CLSP, Capacitated Lot-sizing and Scheduling Problem; CSLSP, Capacitated Stochastic Lot-sizing and Scheduling Problem. For a literature overview for all problem formulations see Kuik et al. (1994).

In our paper we investigate the simulation optimisation approach to a generalised formulation of the CSLSP (Kämpf and Köchel (2004)). Some main characteristics are: N items, a single manufacturing unit, item dependend setup times and cost, arbitrary demand processes, finite space in the storage, backordering and rejection of demand.

2 Model and problem formulation

The model we will investigate consists of various parts (see Fig. 1). We assume continuous time and infinite horizon.

The ***demand process*** is modeled as a compound renewal process with λ_n as intensity of arriving clients and B_n as random variable for the demand size of a client, $\forall n$. If there is no inventory on hand arriving demand can be backordered in a queue with capacity $0 \leq b_n \leq \infty$, $\forall n$. Clients who meet a full backorder queue will be rejected.



Figurel. Scheme of the production-and-inventory system

Produced items are used to satisfy backordered demand. If there is no waiting client the produced items will be stored in the ***warehouse.*** Each item occupies a defined size p_n in the warehouse. The warehouse itselves has capacity P with $0 \leq P \leq \infty$. In dependence of the release policy the warehouse sends a production order to the production pool.

The ***manufacturing unit*** works without failures. Item n is produced with rate μ_n, where it needs a random time to manufacture one unit. The only reasons why the production process must stop are no place for the produced item in the warehouse or emptiness of the pool of production orders.

Next we introduce various policies for the release of manufacturing orders and their sequencing in the production process. The corresponding decisions are called *release decision* and *sequencing decision.*

The production ***sequencing decision*** chooses from the pool of manufacturing orders the next one for manufacturing. A concrete decision is defined by a sequencing rule or sequencing policy. Some examples are the following ones:

First come first serve (FCFS): The production order first inserted in the pool will be produced next. It's denoted by P_S^{FCFS}.

Random: The next production order will be chosen randomly (P_S^{Random}).

Cyclic: There exist a precalculated cyclic production sequence $P_S^{Cyclic} = \{c_1, c_2, \ldots, c_N\}$. If the pool containts no production order for the currently choosen item the next position in the cycle will be considered and so on. In our case the order is predefined as $P_S^{Cyclic} = \{1, 2, \ldots, N\}$ without any claim of optimality.

Longest Waiting Queue (LWQ): This policy chooses as the next one the item with the longest waiting queue (P_S^{LWQ}), i. e., maximal number of waiting demand orders.

Most Waiting Values (MWV): Similar to LWQ but from all production orders will be selected that with the highest backlogged demand (P_S^{MWV}).

In the literature various policies to define ***release decisions*** can be found. One possible classification is to distinguish between *make-to-order policies* and *make-to-stock policies.* Make-to-order policies are well known from queueing theory polling models (cf. Takagi (1986)). We adapted to our model three policy classes:

Exhaustive policy: the machine will manufacture all production orders of an item until all waiting or backlogged demand is served,

Gated policy: the machine will manufacture all production orders of an item which are backlogged when the queue was polled, and

Limited policy: an item is manufactured until either the backlogging queue is emptied or a specified number of backlogged order units is produced.

More flexible are make-to-stock policies. We use (s, Q) policies to generate release decisions, i. e., for item n the parameter Q_n defines the size of a manufacturing order. An order will be released if the inventory position for item n will fall below the order point s_n, i.e. if $I_n(t) < s_n$ at time t. To be more concrete we need the following definitions:

$O_n(t)$ – number of waiting demand orders,

$M_n(t)$ – amount of all outstanding manufacturing orders in front of the production unit,

$I_n^+(t)$ – available inventory or inventory on hand,

$I_n^-(t)$ – backordered demand, and

$I_n(t)$ – inventory position.

We have the usual relations $I_n(t) = I_n^+(t) - I_n^-(t) + M_n(t)$, $I_n^+(t) = max\{0, I_n(t) - M_n(t)\}$, $I_n^-(t) = max\{0, M_n(t) - I_n(t)\}$ and the two restrictions $\sum_{n=1}^{N} p_n \cdot I_n^+(t) \leq P$, $t \geq 0$ and $O_n(t) \leq b_n, n = 1, 2, \ldots, N$, $t \geq 0$.

We also adapted the (s, S) policy to our model. The order point s_n is as for the (s, Q)-policy. But now we have to decide how much of a given product to manufacture. For instance, the gated (s, S) policy means that we produce $S_n - I_n(t)$ parts of item n, where t is the time moment the manufacturing order was released.

Differentiating between release and sequencing decisions we separate the control of the system into two parts - the release of manufacturing orders (release policy) and the sequencing of manufacturing orders (sequencing policy). We call it *decomposed policy.* Let Π denote the set of all these policies.

As optimisation criterion we consider the total average gain $G(\pi)$ per time unit earned under policy π from the system in steady-state. We remark that in case of finite queueing capacities b_n the system is stable for arbitrary policies. Otherwise we have to restrict our considerations to an appropriate subset of Π. Regardless of the great importance of stability considerations they are behind our main topic and they will be neglected here. We define the following:

$Th_n(\pi)$ - throughput as the expected amount of item n, manufactured per time unit;
$H_n(\pi)$ - average inventory on hand;
$W_n(\pi)$ - average backordered demand;
$R_n(\pi)$ - expected amount of demand for item n rejected per time unit;
$U_{nm}(\pi)$ - average number of production switches from item n to item m per time unit;
h_n - holding costs per item n per time unit;
w_n - waiting costs per item n per time unit;
r_n - rejection costs per rejected client requested item n;
pc_n - production costs per item n per time unit;
c_{nm} - setup costs per setup switch from item n to item m.

Now for $G(\pi)$ we can write $G(\pi) = \sum_{n=1}^{N} g_n \cdot Th_n(\pi) - C(\pi)$ with

$$C(\pi) = \sum_{n=1}^{N} \Bigg[h_n \cdot H_n(\pi) + w_n \cdot W_n(\pi) + r_n \cdot R_n(\pi) + \qquad (1)$$
$$+ pc_n \,/\, \mu_n \cdot Th_n(\pi) + \sum_{m=1}^{N} c_{nm} \cdot U_{nm}(\pi) \Bigg].$$

The terms at the right-hand side of (1) express for the steady state the average per time unit occurring holding, waiting, rejection, production, and set-up cost respectively. We remark that $pc_n \,/\, \mu_n$ represents the expected cost to manufacture one unit of item n.

Since $Th_n(\pi) = \lambda_n - R_n(\pi)$ holds for $\pi \in \Pi$ and each n we can write $G(\pi) = \sum_{n=1}^{N} e_n \cdot \lambda_n - C_1(\pi)$, where $e_n = g_n - pc_n \,/\, \mu_n$ denotes the profit for selling one unit of item n and

$$C_1(\pi) = \sum_{n=1}^{N} \left[h_n H_n(\pi) + w_n W_n(\pi) + (r_n + e_n) R_n(\pi) + \sum_{m=1}^{N} c_{nm} U_{nm}(\pi) \right].$$

Because of maximization of $G(\pi)$ is equivalent to minimization of $C_1(\pi)$ the optimisation problem to be solved is

$$\begin{aligned} &\min_{\pi \in \Pi} C_1(\pi) \\ \sum_{n=1}^{N} p_n \cdot I_n^+(t) &\leq P, \text{ for } t \geq 0; \\ O_n(t) &\leq b_n, \text{ for } n = 1, 2, \ldots, N. \end{aligned} \tag{2}$$

Obviously the CSLSP mentioned in section 1 without constraints is not analytically tractable. So more in our formulation of problem (2). Simulation optimisation seems to be the only practicable approach. We will briefly discuss this approach in the next section.

3 Simulation optimisation approach

In optimisation problems, for which the calculation of an analytical solution is difficult or even not possible, we prefer a technique called simulation optimisation (see e.g. Fu (2001)). The basic idea is to couple a simulation system with an optimiser to get an optimal solution (see Fig. 2).

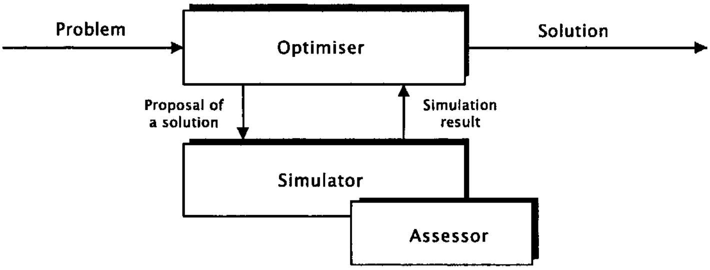


Figure2. Scheme of the simulation optimisation

We remark that once the optimiser is started the search process runs automaticaly without user interaction. When the optimisation stops the best of all solutions found up to now will be returned.

Optimisation. As optimiser we prefer Genetic Algorithms (GA) or Parallel Genetic Algorithms (PGA) because of different reasons. They are applicable to many other general optimisation problems, independend of the solution domain, can handle with different starting points (robustness), and can leave local optima to find the global one. The other advantage is the need of only a few input data. Köchel et al. (2003) showed the successful application of GA's to another complex optimisation problem.

PGAs are build up to compute the fitness of several individuals at the same time on different computers connected through a computer network. To control this technique different approaches are formulated. For more details and an other concrete implementation see Haase and Kohlmorgen (1996). We use a global master-slave PGA model were one master task holds the central population and the multiple clients evaluate the individuals.

Simulation. We use the event-oriented approach for the design of our simulation model. The main model element of this approach is the *future event list.* It is organized as heap and holds the different events characterized by its event time and type. In our simulation model we need only two groups of event types. One group consists of the two events "arrival of a demand order" and "completion of manufacturing of an item". These events are specific for the investigated production-and-inventory-system. They reflect the dynamics of the model. To the other group belong simulation specific events like "completion of the transition phase" and "end of simulation".

4 Numerical examples

This section contains empirical results on three numerical examples with (s, Q)-release-policies and all introduced sequencing polcies.
First we define the common settings for all examples with $n = 1, 2, \ldots, N$.

- Demand processes are compound Poisson processes with different arrival rates λ_n but identical order size distribution

$$B_n \sim \begin{pmatrix} 1 & 10 & 50 \\ 0.6 & 0.3 & 0.1 \end{pmatrix}$$

- Storage capacity is infinite, $P = \infty$.
- Queueing capacities for waiting demand orders are finite with $b_n = 10$,
- Cost factors are $h_n = 1$, $w_n = 1$, $(r_n + e_n) = 20$, and $pc_n = 1$.

The single-item model (Example 1) reflects the case that only average data are available. For Example 2 (two items) and Example 3 (five items) we assume that data about two resp. five groups of the production mix are known.

Example 1 (Single-item model). In the single-item model with $N = 1$ we set $\lambda = 5$ and $\mu = 50$ item units per time unit and setup costs to $c = 3$. Setup times are set to $t = 3$ time units.

Example 2 (Two-item model). Now $N = 2$, $\lambda_1 = 3$, and $\lambda_2 = 2$. The manufacturing rates are set to $\mu_1 = \mu_2 = 50$. The definition of the setup costs c and setup times t are as follows

$$(c_{nm}) = \begin{pmatrix} 1.5 & 4 \\ 1.5 & 4 \end{pmatrix} \text{ resp. } (t_{nm}) = \begin{pmatrix} 1.5 & 4 \\ 1.5 & 4 \end{pmatrix} \text{time units, for } n, m = 1, 2.$$

Example 3 (Five-item model). We assume $N = 5$, $\lambda_n = 1$, and $\mu_n = 50$, $\forall n$. For setup costs and setup times we define $c_{nm} = t_{nm} = m$, for $n, m = 1, 2, \ldots, 5$.

Optimisation results. As described in section 3 a special simulator and a parallel genetic algorithm were combined for optimisation. The search process was restricted to policy parameter values s and Q, each between 1 and 400. Each simulation experiment consists of three runs with simulated time of 200000 time units (days) including 40000 time units for the transition phase. Thus we simulate about 1 million demand arrivals.

Table 1 contains the values found by the applied PGA for the release policy parameters. The most interesting case is that with 5 items, which are all identical excluding setup data. As a consequence the PGA stopped at policies with nearly identical parameter values. This fact indicates that the PGA has "recognised" that the items are identical in some sense. Furthermore, it shows that the PGA moves in the "right" region (near the global optimum) of the decision space.

Table 2 shows the cost details: *manu* for the utilisation of the machine; *wait*, *reject*, and *hold* as the average costs for waiting, rejection, and holding respectively; *setup* denotes the amount of time needed for setup changeovers. Since in all examples we have $pc_n = 1$ and setup cost of one per time unit the columns in Table 2 for manu and setup represent also the corresponding costs per time unit. We see that the cyclic sequencing policy is not the best for the two-item case as well as for the five-item case. The increase of the cost with increasing number N of items is caused by the fact that waiting, rejection and holding costs are considered for each of the items as well as multiple backorder queues exist.

no. of	sequencing	(s, Q)-release policy parameters									
items	decision	release points (s)					order sizes (Q)				
		item	item	item	item	item	item	item	item	item	item
		1	2	3	4	5	1	2	3	4	5
1	FCFS/Random/ Cyclic/LWQ/MWV	75	-	-	-	-	49	-	-	-	-
2	FCFS	84	35	-	-	-	35	275	-	-	-
	Random	38	38	-	-	-	157	247	-	-	-
	Cyclic	210	194	-	-	-	168	236	-	-	-
	LWQ	22	37	-	-	-	245	162	-	-	-
	MWV	13	28	-	-	-	155	17	-	-	-
5	FCFS	166	166	166	164	161	161	166	166	167	166
	Random	141	138	137	138	134	154	147	141	141	134
	Cyclic	139	136	135	136	130	160	148	145	145	139
	LWQ	201	202	172	147	133	132	149	182	172	177
	MWV	105	102	108	101	105	101	101	101	102	105

Tablel. Optimisation results for the problems ((s, Q)-release policy)

no. of items	sequencing decision	C_1	manu	wait	reject	hold	setup
1	FCFS/Random/ Cyclic/LWQ/MWV	61.93	0.78	32.07	9.40	19.50	0.18
2	FCFS	154.07	0.62	87.72	27.47	37.89	0.37
	Random	156.34	0.62	88.31	18.22	38.81	0.38
	Cyclic	162.68	0.59	94.02	30.74	36.91	0.41
	LWQ	156.89	0.62	92.77	28.55	34.57	0.38
	MWV	152.86	0.58	94.82	32.08	24.96	0.42
5	FCFS	397.00	0.54	244.24	37.44	114.32	0.46
	Random	369.46	0.50	199.65	41.38	127.43	0.50
	Cyclic	381.99	0.50	282.77	42.42	55.81	0.50
	LWQ	362.74	0.53	258.76	38.02	64.96	0.47
	MWV	267.10	0.42	194.40	51.63	20.07	0.58

Table2. Cost details for the policies in Table 1 (cost per time unit)

5 Conclusions

In this paper we investigated how to apply simulation optimisation to a general formulation of the CSLSP with multiple items. Our model consists of general demand processes, general cost and gain structure and is flexible enough to include and analyse other release and sequencing policies. Beyond it there exists the possibility to couple the implemented simulator with other optimisation tools.

Further research will go to complex production systems in order to replace the single manufacturing unit. Also, the coupling with a supply chain system in front of this new complex production system could be investigated.

References

1. Fu, M. C. (2001) Simulation optimisation. Proceedings of the 2001 Winter Simulation Conference. New York: ACM, 53–61
2. Haase, K., Kohlmorgen, U. (1996) Parallel genetic algorithm for the capacitated lot-sizing problem. In: Kleinschmidt et al. (1996), Operations Research Proceedings, pp. 370–375, Springer-Verlag
3. Kämpf, M., Köchel, P. (2004), Sequencing and lot size optimisation of a production-and-inventory-system with multiple items using simulation. Thirteenth International Working Seminar on Production Economics, Pre-Prints vol. 3, Igls/Innsbruck, 175–183
4. Köchel, P., Kunze, S., Nieländer, U. (2003) Optimal control of a distributed service system with moving ressources: Application to the fleet sizing and allocation problem. International Journal of Production Economics **81-82**, 443–459
5. Kuik, R., Salomon, M., van Wassenhove, L. N. (1994) Batching decisions: structure and models. European Journal of Operational Research **75**, 243–263
6. Takagi, H. (1986) Analysis of polling systems. MIT Press series in computer systems

Ein Dekompositionsverfahren zur Bestimmung der Produktionsrate einer Fließproduktionslinie mit Montagestationen und stochastischen Bearbeitungszeiten

Michael Manitz

Universität zu Köln, Seminar für Allgemeine Betriebswirtschaftslehre und Produktionswirtschaft, 50923 Köln, GERMANY,
Email: manitz@wiso.uni-koeln.de

Zusammenfassung Im folgenden werden flexible Fließproduktionssysteme mit Montagestationen untersucht. An solchen Stationen werden Komponenten von mehreren Zulieferstationen zur Bildung eines neuen Werkstücks zusammengefügt. Das ist die sog. Synchronisationsbedingung. Der Materialfluß ist asynchron. Die Puffer sind beschränkt. Die Bearbeitungszeiten sind beliebig verteilt. Der folgende Beitrag beschreibt ein Verfahren zur Abschätzung der Produktionsrate eines solchen Systems. Hierfür wird ein Dekompositionsansatz verwendet. Die betrachteten 2-Stationen-Subsysteme werden als Warteschlangenmodelle abgebildet, für die die virtuellen Ankunfts- und Bearbeitungsraten sowie die zugehörigen Variationskoeffizienten zu ermitteln sind. Die Approximationsgüte des Verfahrens wird mit einem Simulationsexperiment untersucht.

Abstract In this paper multi-stage assembly lines are studied. These production systems involve simple processing stations as well as assembly stations at which the workpieces from two input stations have to be joined to form a new one for further processing. Thus, asynchronous flow production systems with a convergent flow of material and synchronisation constraints are being analyzed. This paper describes an approximation procedure for determining the throughput of such systems. In addition, finite buffer capacities and generally distributed processing times are considered. Exact solutions are not available in this case. For performance evaluation the decomposition approach is used. The 2-station submodels are analyzed as G/G/1/Z stopped-arrival queueing systems. In this heuristic approach, the virtual arrival and service rates as well as the squared coefficients of variation of these subsystems have to be determined. A system of decomposition equations which are solved iteratively is being presented. The solution indicates estimated values for the unknown subsystems' parameters. The quality of the presented approximation procedure is investigated by carrying out various simulation experiments.

1 Fließproduktionssysteme mit Montagestationen

Flexible Fließproduktionssysteme werden für Erzeugnisse eingerichtet, die in hohen Stückzahlen und verschiedenen Varianten eines Grundmodells in beliebiger Reihenfolge gefertigt werden sollen. Die Arbeitssysteme werden räumlich so angeordnet, daß ein einheitlicher Materialfluß entsprechend der Reihenfolge der Arbeitsgänge im Arbeitsplan entsteht. Sie sind in der Lage, jeweils eine gewisse Menge von ähnlichen Arbeitsverrichtungen an den Werkstücken durchzuführen, ohne daß dabei aufwendige Umrüstvorgänge erforderlich werden. Vgl. [15].

Ist die Anordnung der Arbeitssysteme linear, so sind für Systeme mit stochastischen, beliebig verteilten Bearbeitungszeiten die Verfahren in [2] und [16] geeignet, die Produktionsrate abzuschätzen. Bei exponentialverteilten oder deterministischen Bearbeitungszeiten können die 2-Stationen-Subsysteme einschließlich eventueller operationszeitabhängiger Stationsausfälle als Markow-Ketten modelliert und ausgewertet werden, s. [4], [9]. Unterscheiden sich die deterministischen Bearbeitungszeiten von Station zu Station, so muß approximativ das Verfahren von [1] für kontinuierliches Material verwendet werden. Vgl. auch [5].

In der betrieblichen Praxis, z. B. im Karosserierohbau (vgl. z. B. [13] und [14]), findet man häufig keine lineare, sondern eine konvergierende Materialflußstruktur vor. Solche Montagelinien unterscheiden sich von linearen Systemen dadurch, daß es sog. Montagestationen (*assembly stations*) gibt, an denen Werkstücke als Komponenten, Baugruppen oder Einzelteile zu einem neuen Werkstück zusammengefügt bzw. -montiert werden.

Montagestationen sind Stationen mit mehr als einer direkten Vorgängerstation. Sie sind charakterisiert durch eine Synchronisationsbedingung, die in Abb. 1 illustriert wird. An einer Montagestation treffen Werkstücke von mehreren Zulieferstationen ein. Unter Synchronisation (*matching*) versteht man in diesem Zusammenhang, daß sich die jeweils ℓ-ten Werkstücke aus jeder Warteschlange vor einer Montagestation von einem bestimmten Zeitpunkt an synchron durch das System bewegen, wenn sich zu diesem Zeitpunkt in jeder Warteschlange mindestens n Werkstücke befinden, $\ell = 1, 2, \ldots, n$. Nur zuvor synchronisierte Werkstücke können durch einen nachfolgenden Bearbeitungsvorgang physisch zusammengefügt werden (Montage, *assembly*). Dieser

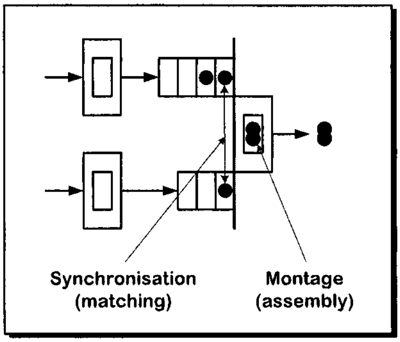


Abb. 1. Synchronisation und Montage an einer Montagestation

kann erst dann beginnen kann, wenn in jedem Eingangspuffer Werkstücke verfügbar sind.

Verfahren zur Leistungsanalyse von Fließproduktionssystemen mit Montagestationen müssen die Synchronisationsbedingung berücksichtigen und abbilden. Für Montage-Demontage-Systeme, in denen die Bearbeitungszeiten deterministisch sind, an deren Stationen aber operationszeitabhängigen Störungen auftreten, werden in [8], [6] und [10] entsprechende Analyseverfahren präsentiert. In [11] und [12] werden exponentialverteilte Bearbeitungszeiten betrachtet. Im folgenden sollen Montagesysteme mit beliebig verteilten Bearbeitungszeiten betrachtet werden. Operationszeitabhängige Stationsausfälle können mit dem Completion-Time-Konzept gemäß [7] berücksichtigt werden.

2 Der Dekompositionsansatz

Wir zerlegen das Modell eines M-Stationen-Montagesystems in insgesamt $M - 1$ pufferbezogene Zwei-Stationen-Subsysteme. Ein solches Subsystem soll den Materialfluß und die Belegung des betrachteten Puffers nachbilden. Die dem Puffer zwischen den Stationen i und m vorgelagerte Station heißt Upstream-Station $M_u(i,m)$ des Subsystems (i,m). Und die dem Puffer $B(i,m)$ nachgelagerte, materialflußabwärts gelegene Station heißt Downstream-Station $M_d(i,m)$. Montagestationen tauchen in mehreren Subsystemen als Downstream-Station auf. In Abb. 2 ist die pufferbezogene Dekomposition eines Montagesystems für einen Ausschnitt aus dem Montagesystem veranschaulicht. Gleichzeitig wird die verwendete Notation illustriert. Die Bezeichnung v_{mj} bezieht sich auf die j-te Vorgängerstation der Station m, n_m auf die Nachfolgerstation. Die Menge der Vorgängerstationen der Station m wird mit $\mathcal{V}_m$ bezeichnet, die Menge der Nachfolgerstationen mit $\mathcal{N}_m$.

Jedes Subsystem läßt sich als (G/G/1/$C_{i,m}$ + 2)-Warteschlangensystem modellieren, $m = 1, 2, \ldots, M$, $\mathcal{V}_m \neq \emptyset$. Eine Blockierung der Station $M_u(i,m)$ tritt ein, wenn ein fertig bearbeitetes Werkstück nicht in den nachfolgenden Puffer mit der Kapazität $C_{i,m}$ transferiert werden kann, weil sich bereits insgesamt $C_{i,m} + 1$ Werkstücke in der Bearbeitungseinrichtung (Server) an der Station $M_d(i,m)$ und im Puffer $B(i,m)$ aufhalten (Blockierung nach Bearbeitung, *blocking after service*). Aus der Sicht des repräsentierenden Warteschlangensystems füllt das blockierte Werkstück das System, das somit eine Kapazität von insgesamt $C_{i,m} + 2$ Plätzen aufweist. Im Fall einer Blockierung wird der Ankunftsprozeß unterbrochen, d. h., es liegt ein sog. Stopped-arrival-Ankunftsprozeß vor.

Aus dem Prinzip des Dekompositionsansatzes folgt, daß die Subsysteme unabhängig voneinander analysiert werden. Tatsächlich vorhandene Abhängigkeiten müssen sich in einer Modifikation der die Subsysteme beschreibenden Modellparameter niederschlagen. Die eigentliche Problemstellung eines Dekompositionsverfahrens zur Leistungsanalyse eines Montagesystems besteht deshalb darin, virtuelle Ankunfts- und Bearbeitungsraten für die äqui-

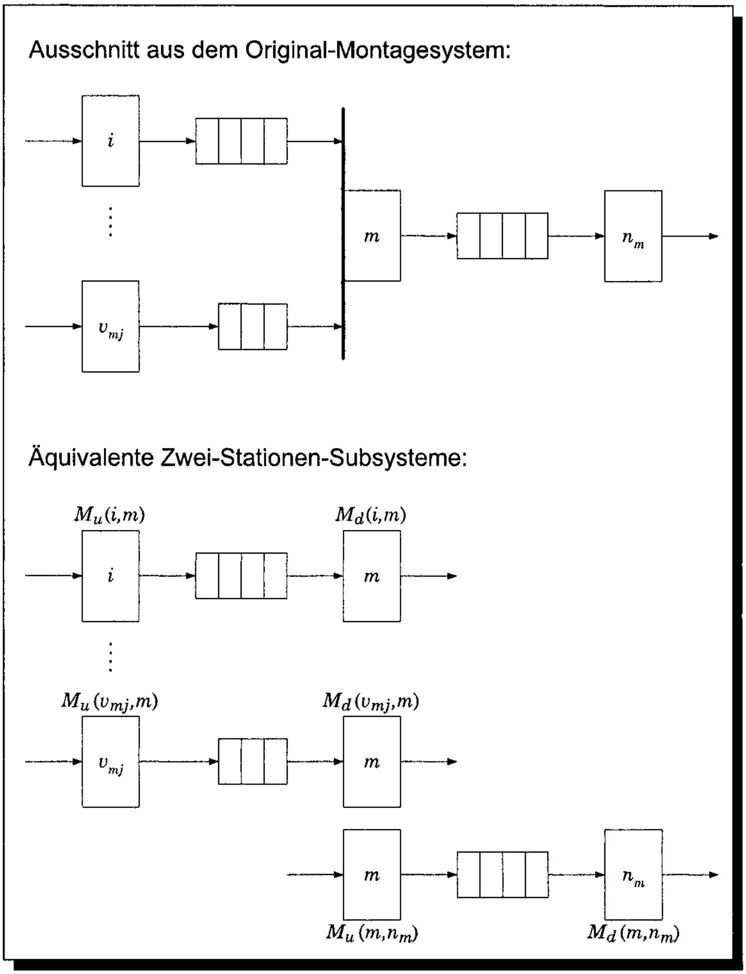


Abb. 2. Pufferbezogene Dekomposition eines Montagesystems in Zwei-Stationen-Subsysteme (Ausschnitt um die Station m herum)

valenten Warteschlangenmodelle zu bestimmen, so daß sie den Materialfluß im entsprechenden Abschnitt des Originalsystems nachempfinden.

Die Modellierung eines Subsystems als Warteschlangensystem impliziert, daß die Downstream-Station niemals blockiert ist, wogegen die von ihr repräsentierte Originalstation durchaus blockiert sein kann. Das Blockiert-Sein der Station m wird daher als Verlängerung der Bearbeitung an der Downstream-Station $M_d(v_{mj}, m)$ des äquivalenten Warteschlangensystems aufgefaßt. Die Blockierzeit entspricht der Restverweilzeit eines Werkstücks an der Nachfolgerstation n_m, deren Wahrscheinlichkeitsverteilung bei Annahme der Gedächtnislosigkeit der Verteilung der virtuellen Bearbeitungszeit im nachfolgenden Subsystem entspricht. Analog wird die Wartezeit einer Komponente auf Synchronisation berücksichtigt: Das Werkstück der Zulieferstation v_{mj} muß – bei unabhängiger Bestückung – am Server der Station m solange warten, bis alle anderen Zulieferstationen mindestens ein Werkstück geliefert haben. Die Wartezeit auf Synchronisation entspricht damit der längsten Restfertigstellungszeit an den Zulieferstationen, die durch das Maximum der Zwischenankunftszeiten in den parallelen Subsystemen approximiert werden kann. Die virtuelle Bearbeitungsrate an der Station $M_d(v_{mj}, m)$, $\mu_d(v_{mj}, m)$, ist damit kleiner als die Originalbearbeitungsrate μ_m. Man ermittelt den Erwartungswert der virtuellen Bearbeitungszeit wie folgt:

$$\begin{aligned}\frac{1}{\mu_d(v_{mj},m)} = {} & \frac{1}{\mu_m} + p_B^*(m,n_m)\cdot\frac{1}{\mu_d(m,n_m)} \\ & + P^*\left(\bigcup_{i\in\mathcal{V}_m,\, i\neq v_{mj}} \{\text{„}(i,m)\text{ leer“}\}\right)\cdot\max_{i\neq v_{mj}}\left\{\frac{1}{\mu_u(i,m)}\right\} \\ & \left(m \in \{\{1,\ldots,M\} \big| |\mathcal{V}_m| \geq 2, \mathcal{N}_m \neq \emptyset\}; j = 1,\ldots,|\mathcal{V}_m|\right)\end{aligned} \tag{1}$$

Dabei ist $p_B^*(m,n_m)$ die zeitpunktbezogene Blockierwahrscheinlichkeit des Subsystems (m,n_m). $P^*(\cdot)$ ist die zeitpunktbezogene Wahrscheinlichkeit dafür, daß mindestens eines der zu (v_{mj},m) parallelen Subsysteme leer ist. Als eine weitere Approximation kommt hinzu, daß das Maximum der Momente anstelle des entsprechenden Moments eines Maximums von Zufallsvariablen verwendet wird.

Die Station $M_u(m,n_m)$, die ebenfalls die Originalstation m repräsentiert, ist verantwortlich für den Zugangsprozeß in das Subsystem (m,n_m), d. h., sie erzeugt den abzubildenen Ankunftsprozeß in das äquivalente Warteschlangensystem. Die virtuelle Zwischenankunftszeit (bei aktivem Ankunftsprozeß) umfaßt aber nicht nur die Bearbeitungszeit an der Station m, sondern auch die Leerzeit dieser Station. Die Wartezeit der Station m auf die Komponente j kann durch die Zwischenankunftszeit im Subsystem (v_{mj},m) modelliert werden. Hinzu kommt wieder die Wartezeit der j-ten Komponente auf Synchronisation. Es ergibt sich

$$\begin{aligned}\frac{1}{\mu_u(m,n_m)} = {} & \frac{1}{\mu_m} + p_S^*(v_{mj},m)\cdot\frac{1}{\mu_u(v_{mj},m)} \\ & + P^*\left(\bigcup_{i\in\mathcal{V}_m,\, i\neq v_{mj}} \{\text{„}(i,m)\text{ leer“}\}\right)\cdot\max_{i\neq v_{mj}}\left\{\frac{1}{\mu_u(i,m)}\right\} \\ & \left(m \in \{\{1,\ldots,M\} \big| |\mathcal{V}_m| \geq 2, \mathcal{N}_m \neq \emptyset\}; j = 1,\ldots,|\mathcal{V}_m|\right)\end{aligned} \tag{2}$$

mit $p_S^*(v_{mj},m)$ als zeitpunktbezogene Starving- bzw. Leerwahrscheinlichkeit des Subsystems (v_{mj},m).

Aus den ermittelten virtuellen Ankunfts- und Bearbeitungsraten und ggf. analog approximierten Variationskoeffizienten können die Kenngrößen der

Warteschlangensysteme bestimmt werden. Für beliebig verteilte Zwischenankunfts- und Bearbeitungszeiten läßt sich mit dem Ansatz von [3] z. B. die stationäre Zustandswahrscheinlichkeit P_n abschätzen, d. h. die Wahrscheinlichkeit dafür, daß sich im Warteschlangensystem genau n Werkstücke aufhalten, $n = 0, 1, \dots, C_{i,m} + 2$. Für die Produktionsrate des Subsystems (i, m) folgt dann

$$X(i,m) = \mu_d(i,m) \cdot (1 - p_S(i,m)) \qquad (m \in \{1, 2, \dots, M\};\ i \in \mathcal{V}_m) \quad (3)$$

mit $p_S(i,m) = P_0\big(\mu_u(i,m), \zeta_u(i,m), \mu_d(i,m), \zeta_d(i,m), C_{i,m} + 2\big)$ als Leer- bzw. Starvingwahrscheinlichkeit der Station $M_d(i,m)$. Ist die Produktionsrate $X(i,m)$ bekannt, dann können mit Hilfe der Gleichungen (1) und (2) die zeitpunktbezogenen Blockier- und Leerwahrscheinlichkeiten der Warteschlangensysteme, $p_B^*(i,m)$ und $p_S^*(i,m)$, ermittelt werden.

3 Das Verfahren und Rechenergebnisse

Die Bestimmungsgleichungen (1) und (2) sowie ggf. entsprechende Ausdrücke für die Variationskoeffizienten beschreiben die Beziehungen zwischen den Parametern benachbarter Subsysteme und werden als Dekompositionsgleichungen bezeichnet. Sie bilden ein Gleichungssystem, das iterativ – initialisiert mit den Original-Bearbeitungsraten – zur Abschätzung der unbekannten, virtuellen Parameter ausgewertet wird. In jeder Iteration werden für jedes Subsystem zunächst in einer der Materialflußrichtung entsprechenden „Vorwärtsrechnung“ die Upstream-Parameter aktualisiert. Alle dafür benötigten Upstream-Parameter von Vorgängersubsystemen sind in der laufenden Iteration bereits aktualisiert worden. Für die Downstream-Parameter setzt man die Werte aus der letzten Iteration ein. Anschließend werden in einer „Rückwärtsrechnung“ die Downstream-Parameter bestimmt. Als Abschätzung für die Produktionsrate des gesamten Montagesystems verwendet man die Produktionsrate des Subsystems mit der Station M als Downstream-Station. Wegen der Conservation-of-Flow-Eigenschaft, nach der alle Subsysteme (zumindest annähernd) die gleiche Produktionsrate aufweisen, ist diese Auswahl jedoch beliebig.

Rechentests haben ergeben, daß das beschriebene Dekompositionsverfahren gute Abschätzungen der Produktionsrate liefert. Die Ergebnisse liegen für die hier untersuchte konvergierende Struktur des Materialflusses in dem Bereich, den [2] bereits für lineare Systeme dokumentiert haben. Für ein Montagesystem gemäß Abb. 3 konnten Abweichungen der analytisch ermittelten Produktionsrate von einem entsprechenden Simulationsergebnis von unter 4 % festgestellt werden. Es läßt sich beobachten, daß die analytischen Abschätzungen mit zunehmenden Puffergrößen besser werden, vgl. für balancierte Systemkonfigurationen Abb. 4. Voraussetzung dafür ist, daß die Variationskoeffizienten der Bearbeitungszeiten kleiner als 2 sind.

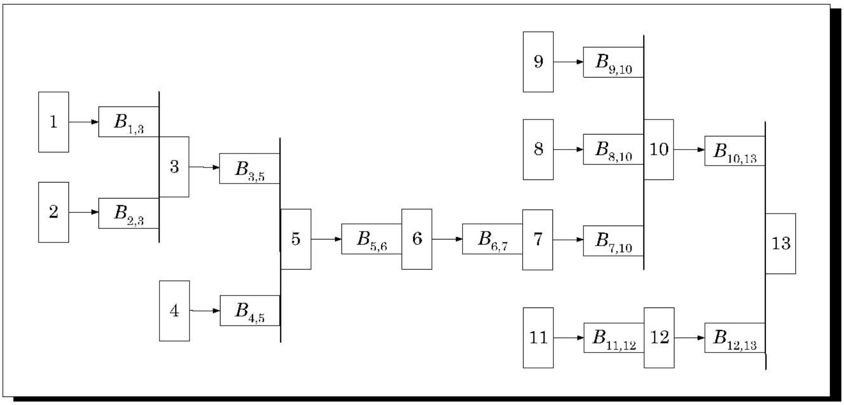


Abb. 3. Ein 13-Stationen-Montagesystem

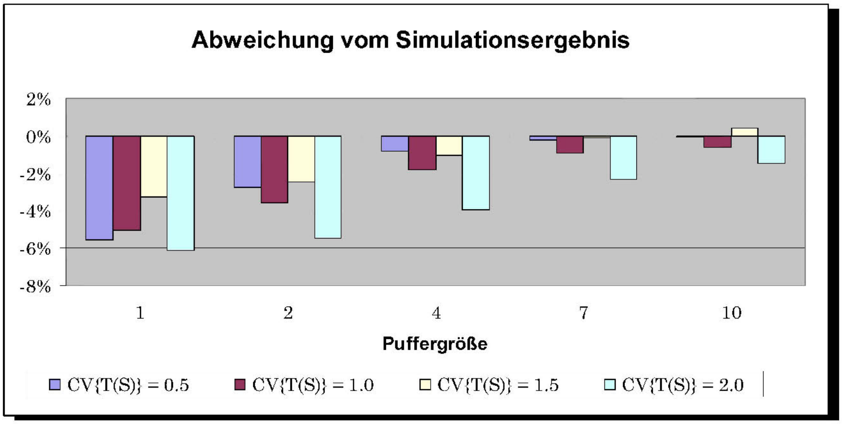


Abb. 4. Durchschnittliche Abweichungen vom Simulationsergebnis

Im Gegensatz zum beschriebenen analytischen Ansatz liefert das Simulationsmodell mit 20 Replikationen und einer Laufzeit von 101 000 Zeiteinheiten nicht „auf Knopfdruck", sondern erst nach durchschnittlich 25 Minuten eine ausreichend genaue Abschätzung der Produktionsrate für eine konkrete Konfigurationsalternative des Montagesystems. Aus Rechenzeitgründen ist daher im Rahmen von Optimierungsverfahren zur Lösung des BWAP (*buffer and workload allocation problem*) die analytische Vorgehensweise bei der Leistungsbewertung vorzuziehen.

Literatur

1. Burman, M. H. (1995). New Results in Flow Line Analysis. Ph. D. thesis, Massachusetts Institute of Technology.

2. Buzacott, J. A., X.-G. Liu und J. G. Shanthikumar (1995). Multistage flow line analysis with the stopped arrival queue model. IIE Transactions **27**, 444–455.
3. Buzacott, J. A., und J. G. Shanthikumar (1993). Stochastic Models of Manufacturing Systems. Englewood Cliffs: Prentice-Hall.
4. Dallery, Y., R. David und X.-L. Xie (1988). An effincient algorithm for analysis of transfer lines with unreliable machines and finite buffers. IIE Transactions **20**(3), 280–283.
5. Dallery, Y., und S. B. Gershwin (1992). Manufacturing flow line systems – A review of models and analytical results. Queueing Theory **12**, 3–94.
6. Di Mascolo, M., R. David und Y. Dallery (1991). Modeling and analysis of assembly systems with unreliable machines and finite buffers. IIE Transactions **23**(4), 315–330.
7. Gaver, D. P. (1962). A waiting line with interrupted service, including priorities. Journal of the Royal Statistical Society **24**(2), 73–90.
8. Gershwin, S. B. (1991). Assembly/disassembly systems: An efficient decomposition algorithm for tree-structured networks. IIE Transactions **23**(4), 302–314.
9. Gershwin, S. B. (1994). Manufacturing Systems Engineering. Englewood Cliffs: Prentice-Hall.
10. Gershwin, S. B., und M. H. Burman (2000). A decomposition method for analyzing inhomogeneous assembly/disassembly systems. Annals of Operations Research **93**, 91–115.
11. Helber, S. (1999). Performance Analysis of Flow Lines with Non-Linear Flow of Material, Band 473 von Lecture Notes in Economics and Mathematical Systems. Berlin, Heidelberg, New York: Springer.
12. Jeong, K.-C., und Y.-D. Kim (1998). Performance analysis of assembly/disassembly systems with unreliable machines and random processing times. IIE Transactions **30**(1), 41–53.
13. Liu, X.-G. (1990). Towards Modeling Assembly Systems – Application of Queueing Networks with Blocking. Ph. D. thesis, University of Waterloo.
14. Spieckermann, S., K. Gutenschwager, H. Heinzel und S. Voß (2000). Simulation-based Optimization in the Automotive Industry – A Case Study on Body Shop Design. Arbeitsbericht, SimPlan Gesellschaft für Simulation betrieblicher Abläufe mbH und Technische Universität Braunschweig, Institut für Wirtschaftswissenschaften, Maintal und Braunschweig.
15. Tempelmeier, H. (1996) Flexible Fertigungstechniken. In: Kern, W., H.-H. Schröder und J. Weber (Hrsg.), Handwörterbuch der Produktionswirtschaft (2. Aufl.). Stuttgart: Schäffer-Poeschel. S. 502–512
16. Tempelmeier, H., und M. Bürger (2001). Performance evaluation of unbalanced flow lines with general distributed processing times, failures and imperfect production. IIE Transactions **33**, 293–302.

A dynamic model for strategic supplier selection

Eric Sucky

Department of Supply Chain Management, Goethe-University, Mertonstr. 17,
60054 Frankfurt am Main, esucky@wiwi.uni-frankfurt.de

Abstract. Supplier selection decisions at the strategic level are focused on strategic items with both a high supply risk and a high profit impact. Therefore, strategic supplier selection decisions have to be long-term orientated considering (i) mutual commitments between the partners involved, (ii) fixed costs upon selection of a new supplier in the form of investment in training, and technology, as well as (iii) significant costs of switching from one supplier to another. Existing approaches of supplier selection neglect the interdependencies in time arising from investment costs of selecting a new supplier and costs of switching from one supplier to another. Moreover, it is assumed that the set of employed suppliers can be changed each period without cost. These shortcomings of current approaches motivates the research presented in this paper. A stochastic dynamic model for supplier selection based on hierarchical planning approaches will be presented. This model enables the evaluation of alternative dynamic supplier selection strategies.

1 Introduction

A supply chain (SC) can be considered as a network of different geographically dispersed facilities, where products are transformed and stored, and transportation links that connect these facilities. The major task of supply chain configuration (SCC) – the strategic level of supply chain management (SCM) – is the design of the SC in such a way, that a given set of objectives is achieved. SCC decisions concern the entire SC process. On the production side decisions about the location and capacity of plants are the typical tasks of SCC. On the distribution side, in general, the locations of warehouses have to be determined. This paper focuses on the major task on the procurement side: strategic supplier selection. Each potential supplier is equipped with his own production facilities and warehouses. Therefore, by selecting suppliers the network's structure will be influenced.

Supplier selection decisions are complicated by the fact that multiple criteria must be considered in the decision making process [12]. Supplier selection decisions are further complicated by the fact that some suppliers may have different performances in regard to the different criteria. For example, the supplier who has the best quality performance may not have the best delivery performance [13]. Since the 1960's when *Dickson* [2] suggested 23 different selection criteria, the analysis of these criteria has been the focus of a large number of publications. Quantitative approaches supporting supplier selection range from simple linear weighting models to complex mathematical programming models. Extensive literature overviews on available approaches to supplier evaluation and selection are given by *De Boer/Labro/Morlacchi* [1] and *Weber/Current/Benton* [12].

Supplier selection decisions at the strategic level of SCM are focused on strategic items with both a high supply risk and a high profit impact. Here, the buyer aspires to a long-term orientated strategic partnership type relationship with his suppliers [3]. The considered long-term planning environment is characterized by only a few suppliers, long-term contracts and substantial switching costs, either because of a high unique specification or because of the scarcity of the material [1]. Therefore, strategic supplier selection decisions have to be long-term orientated considering (i) mutual commitments between the partners involved, (ii) fixed costs upon selection of a new supplier in the form of investment in training, and technology [6], as well as (iii) significant costs of switching from one supplier to another. Besides the price, the quality, and the delivery time of the supplier's products, these switching and investment costs have to be taken into account selecting one ore more alternative suppliers for a strategic item.

Existing approaches of supplier selection neglect these interdependencies in time arising from investment costs of selecting a new supplier and costs of switching from one supplier to another. Moreover, it is assumed that the set of employed suppliers can be changed each period without cost. These shortcomings of current approaches motivates the research presented in this paper. We propose a stochastic dynamic model for supplier selection based on hierarchical planning approaches. Our model enables the evaluation of alternative dynamic supplier selection strategies. The remainder of this paper is organized as follows: In section 2 we first present a generic single-period supplier selection model. With it the impacts of the supplier selection decisions on medium-term supply chain performance can be evaluated. Using this model we generate the base-level of our hierarchical approach, presented in section 3. On the top-level of the proposed hierarchical approach we model supplier selection alternatives and the corresponding investment and switching costs. In section 4, we show, how this hierarchical approach can provide valuable decision support for strategic supplier selection decisions.

2 A deterministic one-period supplier selection model

In this paper, we utilize a modified version of the deterministic single-period supplier selection model presented by *Jayaraman/Srivastava/Benton* [5]. The model simultaneously determines the set of suppliers and allocates the demand of the buyer's plants among them. For a more detailed formulation, which splits the regarded period in multi sub-periods, considering time-varying prices dictated by time-varying quantity discounts, and the possibility of carrying forward products to a future sub-period, incurring holding costs, see *Tempelmeier* [11] and *Reith-Ahlemeier* [8]. In our model, the buyer of a specific product selects one or more suppliers from a set of potential suppliers *H*, pre-selected by using a scoring-model. The supplier selection decision is constrained (i) by the supplier's specific lead-times with regard to the buyer's different geographically dispersed plants, and (ii) by the supplier's specific production and supply capacities. Therefore, a supplier $h \in H$ can be selected only, if he can meet the required lead-time and if he has the capacity to achieve the demands, as well. The following notation is used:

H Set of potential suppliers $h \in \{1,\dots,H\}$
J Set of the buyer's plants $j \in \{1,\dots,J\}$
d_j Demand per period at buyer's plant $j \in J$
cap_h Production and supply capacity of supplier $h \in H$
sup_h Minimum supply agreement with supplier $h \in H$
l_{hj} Lead-time for supplier h to produce and supply the product to plant $j \in J$
L_j Maximum lead-time accepted for buyer's plant $j \in J$
c_{hj} Price per unit from supplier $h \in H$ to buyer's plant $j \in J$
C_h Fixed transaction cost associated with utilizing supplier $h \in H$
c_h Cost per non accepted unit of supplier $h \in H$ if the buyer does not fulfill the minimum supply quantity agreement sup_h
c_j Penalty cost per unit if the suppliers' deliveries do not fulfill the demand per period at buyer's plant $j \in J$
$\hat{x}_h$ Quantity of the minimum supply quantity agreement with supplier $h \in H$ non fulfilled by the buyer
$\breve{x}_j$ Quantity of the demand at plant $j \in J$ non fulfilled by the suppliers
x_{hj} Quantity supplied from supplier h to buyer's plant $j \in J$

$$y_h = \begin{cases} 1 & \text{if supplier h is employed for purchases} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{hj} = \begin{cases} 1 & \text{if supplier h is employed to supply plant j} \\ 0 & \text{otherwise} \end{cases}$$

The supplier selection problem can be formulated as follows:

$$\min \ TRC^{ope} = \sum_{h \in H} \sum_{j \in J} c_{hj} \cdot x_{hj} + \sum_{j \in J} c_j \cdot \breve{x}_j + \sum_{h \in H} c_h \cdot \hat{x}_h + \sum_{h \in H} C_h \cdot y_h \qquad (1)$$

subject to

$$\sum_{h \in H} x_{hj} + \breve{x}_j = d_j \qquad \text{for all j} \qquad (2)$$

$$\sum_{j \in J} x_{hj} + \hat{x}_h \geq sup_h \qquad \text{for all h} \qquad (3)$$

$$\sum_{j \in J} x_{hj} \leq cap_h \qquad \text{for all h} \qquad (4)$$

$$l_{hj} \cdot y_{hj} \leq L_j \qquad \text{for all h and j} \qquad (5)$$

$$y_{hj} \leq y_h \qquad \text{for all h and j} \qquad (6)$$

$$x_{hj} \leq y_{hj} \cdot d_j \qquad \text{for all h and j} \qquad (7)$$

$$x_{hj}, \breve{x}_j, \hat{x}_h \geq 0 \;;\; y_{hj}, y_j \in \{0,1\} \qquad \text{for all h and j} \qquad (8)$$

The objective is to minimize the buyer's total relevant cost. In this formulation, the objective function (1) has four components: the purchasing costs, the costs for not fulfilling the supply agreements, the penalty costs for not fulfilling demands, and the fixed transaction costs associated with utilizing a supplier. The feasible solutions are subject to the constraints (2)-(8). Constraints (2) ensure that the delivered quantities do not exceed the demands of the buyer's plants. Constraint set (3) is used to ensure that the minimum supply agreements with the suppliers are satisfied. Each supplier has a certain fixed capacity that he can use to produce and supply products. Constraints (4) are the supplier's capacity constraints. Each supplier needs a specific lead-time in which to fulfill an order of the buyer. The supplier's lead-time may vary for each of the buyer's plants. Constraint set (5) ensures that a supplier will only be selected to supply a specific plant if he satisfies the required lead-time. Constraints (6) and (7) require that an order can only be placed with a supplier if he is employed. Constraint set (8) ensures nonnegativity of the decision variables and defines the binary decision variables.

This supplier selection model may be adopted in a short-term planning environment focusing on the operational problems of order sizing and supplier selection for routine items with low supply risk and low profit impact [11]. Applying this model in a multi-period planning problem implies that the selected set of suppliers can be modified in every period without cost. However, supplier selection decisions for strategic items are characterized (i) by long-term contracts associated with contractual penalties in case of premature termination of an existing relationship, and (ii) high costs of investment in training and technology as well as joint costs of development if a new relationship is established. Therefore, switching from one supplier to another induces significant costs, which have to be considered in the selection process. To incorporate these time interdependencies a dynamic stochastic model for supplier selection based on hierarchical planning approaches will be developed. On the top-level of this hierarchical approach we model supplier selection alternatives and the corresponding investment and switching costs. The model presented above will be used as the base-level to evaluate the impacts of the supplier selection decisions on medium-term performance. In the following we first describe the buyer's dynamic decision problem.

3 The dynamic supplier selection problem

The planning period of the dynamic supplier selection problem is assumed to be [0,T] and can be divided into T sub-periods [0,1],...,[T-1,T], whereas the beginning point of a sub-period [t,t+1] is $t \in \{0,...,T-1\}$, and the end point of [t,t+1] is t+1. The state s_t of the buyer's supply side – the buyer's supply system – at the beginning of a sub-period [t,t+1] is characterized by the set of employed suppliers, the potential supply linkages, and the buyer's plants. For t=0, we assume, there is still no supplier employed for supplying the regarded product. Therefore, the state $\mathbf{s}_0$ of the buyer's supply system at the point of time t=0 can be characterized by the set of potential suppliers *H* and the associated set of potential supply linkages. Each supplier $h \in H$ can be characterized by his supply capacity per period, an agreed on minimum supply quantity, his plant specific lead-times, and the relevant costs. The buyer can identify the set of alternatives for supplier selection $A_0(\mathbf{s}_0)$. An alternative $\mathbf{a}_{i0} \in A_0(\mathbf{s}_0)$, with $i \in \{1,...,I\}$, determines a specific combination of suppliers $h \in H$ employed for supplying the regarded product in sub-period [0,1]. We assume that the suppliers selected by choosing the alternative $\hat{\mathbf{a}}_{i0} \in A_0(\mathbf{s}_0)$ are employed in [0,1]. Determining the set of employed suppliers for [0,1], induces costs of investment in training, technology, and development. Note, that the decision $\hat{\mathbf{a}}_{i0}$ can be considered as an investment into buyer-supplier relationships. Whether an employed supplier is used for supplying or not depends on the plants specific demands in sub-period [0,1]. Therefore, in sub-period [0,1], the buyer has to allocate the plant's demand to the employed suppliers. The buyer determines the activated supply linkages and the associated order quantities. These decisions depend on the expected plants specific demands $\tilde{\mathbf{d}}'_0 = (\tilde{d}_{10},..., \tilde{d}_{J0})$. We assume that the discrete random variable $\tilde{\mathbf{d}}_0$ has the possible realizations $\mathbf{d}_{10},...,\mathbf{d}_{R0}$, with associated probabilities $w(\mathbf{d}_{r0})$, $w(\mathbf{d}_{10}) + ... + w(\mathbf{d}_{R0}) = 1$.

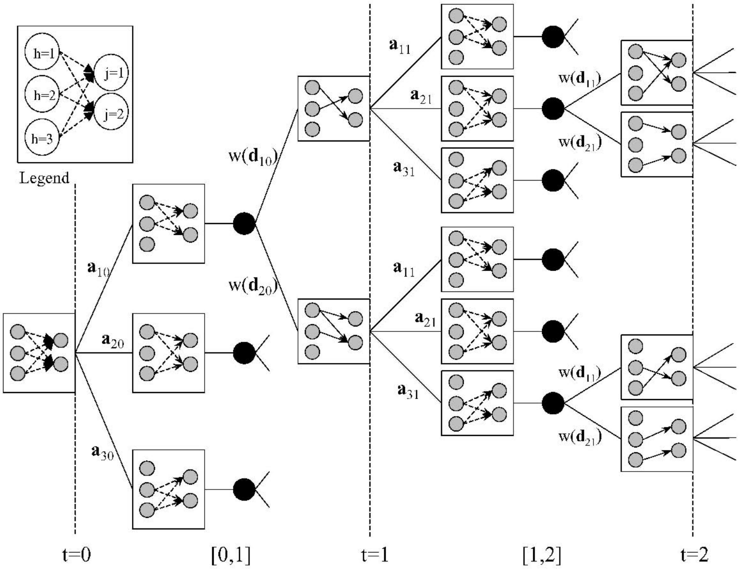


Fig. 1. Dynamic supplier selection process

The state of the buyer's supply system $\mathbf{s}_1$ depends on $\mathbf{s}_0$, $\hat{\mathbf{a}}_{i0} \in A_0(\mathbf{s}_0)$, and $\tilde{\mathbf{d}}_0$, i.e. the state of the supply system at the end of period [0,1] is a random variable $\tilde{\mathbf{s}}_1 = f_0(\mathbf{s}_0, \hat{\mathbf{a}}_{i0}, \tilde{\mathbf{d}}_0)$. At the beginning of each sub-period t=1,...,T-1 the buyer can choose from a set of alternatives $A_t(\mathbf{s}_t)$ which depends on the realized state $\mathbf{s}_t$, with $\tilde{\mathbf{s}}_t = f_{t-1}(\mathbf{s}_{t-1}, \hat{\mathbf{a}}_{it-1}, \tilde{\mathbf{d}}_{t-1})$. Thus, for the planning period [0,T], the buyer has to determine a supplier selection strategy, i.e. a series of interdependent decisions $(\hat{\mathbf{a}}_{i0}, \hat{\mathbf{a}}_{i1}, \hat{\mathbf{a}}_{i2,}..., \hat{\mathbf{a}}_{iT-1})$. For that, first, the buyer has to generate all realizable dynamic decision sequences. Second, the decision sequences have to be evaluated in regard to the associated relevant cost, and finally, the buyer has to choose precisely one alternative $\hat{\mathbf{a}}_{i0} \in A_0(\mathbf{s}_0)$ at point of time t=0. The decision tree in figure 1 exemplarily illustrates the dynamic decision process.

In this example a buyer has two geographically dispersed plants j=1,2. The buyer has identified three potential suppliers h=1,2,3 for a specific strategic item which has to be procured. It is assumed that the buyer follows a double sourcing strategy, i.e. in each period he wants to select two of the potential suppliers. The set of alternatives $A_0(\mathbf{s}_0)$ is given by $\{\mathbf{a}_{10}, \mathbf{a}_{20}, \mathbf{a}_{30}\}$. Regarding the sub-periods [0,1] and [1,2], one realizable supplier selection strategy $(\hat{\mathbf{a}}_{10}, \hat{\mathbf{a}}_{21})$ may be: employ the suppliers h=1 and h=2 for [0,1] and substitute supplier h=2 by supplier h=3 in [1,2] (see figure 1). The state of the supply system $\mathbf{s}_1$, i.e. the set of employed suppliers, the activated supply linkages, the buyer's plants, and the supply quantities depends on the plant specific stochastic demands in [0,1]. We assume $\tilde{\mathbf{d}}_0 \in \{\mathbf{d}_{10}, \mathbf{d}_{20}\}$, with associated probabilities $w(\mathbf{d}_{10}) > 0$, $w(\mathbf{d}_{20}) = 1 - w(\mathbf{d}_{10})$.

4 A hierarchical approach supporting supplier selection

Based on the dynamic decision process, described in the preceding section, we now introduce a hierarchical planning approach supporting the dynamic supplier selection. According to the framework of *Schneeweiss* [9] and *Huchzermeier/ Cohen* [4] we propose the following hierarchical approach in order to evaluate supplier selection strategy alternatives.

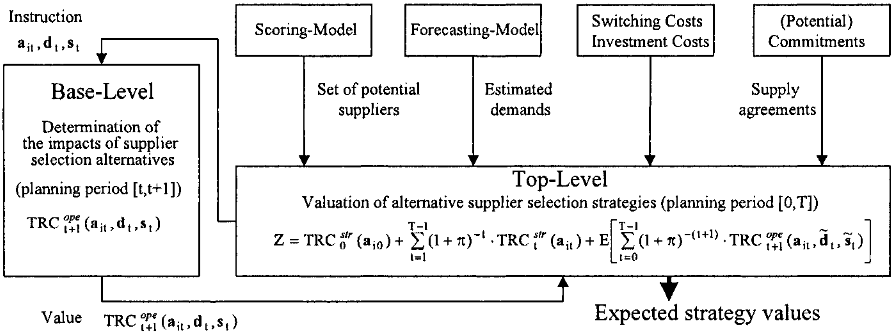

Instruction
a_{it}, d_t, s_t
Base-Level
Determination of the impacts of supplier selection alternatives
(planning period [t,t+1])
$TRC_{t+1}^{ope}(a_{it}, d_t, s_t)$
Value $TRC_{t+1}^{ope}(a_{it}, d_t, s_t)$
Scoring-Model
Forecasting-Model
Switching Costs Investment Costs
(Potential) Commitments
Set of potential suppliers
Estimated demands
Supply agreements
Top-Level
Valuation of alternative supplier selection strategies (planning period [0,T])
$Z = TRC_0^{str}(a_{i0}) + \sum_{t=1}^{T-1}(1+\pi)^{-t} \cdot TRC_t^{str}(a_{it}) + E\left[\sum_{t=0}^{T-1}(1+\pi)^{-(t+1)} \cdot TRC_{t+1}^{ope}(a_{it}, \tilde{d}_t, \tilde{s}_t)\right]$
Expected strategy values

Fig. 2. Modeling framework

Figure 2 shows the overall modeling framework for dynamic strategic supplier selection. First, at the top-level, we define a set of supplier selection strategies. Second, the value of a particular supplier selection strategy is determined by solving the deterministic single-period supplier selection model, presented in section 2, for every possible realization of the stochastic demand. Third, a multi-period stochastic dynamic programming formulation uses these values as inputs for determining the buyer's expected present value of relevant costs for each alternative at t=0. Thus, the stochastic dynamic programming model is formulated as follows:

$$\begin{aligned} \min \ Z = {} & TRC_0^{str}(a_{i0}) + \sum_{t=1}^{T-1}(1+\pi)^{-t} \cdot TRC_t^{str}(a_{it}) \\ & + \sum_{t=0}^{T-1}(1+\pi)^{-(t+1)} \cdot E\left[TRC_{t+1}^{ope}(a_{it}, \tilde{d}_t, \tilde{s}_t)\right] \end{aligned} \tag{9}$$

subject to

$$a_{i0} \in A_0(s_0) \tag{10}$$

$$a_{it} \in A_t(\tilde{s}_t) \qquad \text{for t=1,2,...,T-1} \tag{11}$$

$$\tilde{s}_{t+1} = f_t(s_t, \hat{a}_{it}, \tilde{d}_t) \qquad \text{for t=0,1,...,T-1} \tag{12}$$

$$\tilde{d}_t \in \{d_{1t}, d_{2t}, ..., d_{Rt}\} \qquad \text{for t=0,1,...,T-1} \tag{13}$$

The decision maker, the buyer, is assumed to be risk-neutral. The objective of the buyer is to minimize the expected present value of his total relevant cost in the planning period [0,T]. The objective function on the top-level (9) is composed of two components. The strategic total relevant cost $TRC_t^{str}(a_{it})$ describes the switching and investment costs resulting from an alternative $a_{it} \in A_t(s_t)$. We refer to them as costs of (re-) configuring the supply system. We assume that $TRC_t^{str}(a_{it})$ occur at the beginning of period [t,t+1]. Using the discount rate π, the buyer's time preferences are considered. Thus, for every realizable supplier selection strategy $(a_{i0}, a_{i1}, a_{i2,}..., a_{iT-1})$ the buyer's strategic total relevant costs can be calculated. We assume, without loss of generality, that a supplier selected in t is able to supply the regarded product in sub-period [t,t+1], i.e. no switching time arise. In case switching time has to be considered, e.g. a switching time of one period, the (re-) configuration cost has to be taken into account one period earlier: $\mathrm{TRC}_{t}^{str}(\mathbf{a}_{it+1})$.

The operative total relevant cost $\mathrm{TRC}_{t+1}^{ope}(\mathbf{a}_{it}, \mathbf{d}_{t}, \mathbf{s}_{t})$ for t=0,...,T-1 are calculated by solving the deterministic, single-period model presented in section 2. For each combination $(\mathbf{s}_{t}, \mathbf{a}_{it}, \mathbf{d}_{rt})$ the optimal solution of this supplier selection model is determined. Considering the probabilities $w(\mathbf{d}_{rt})$ the expected value will be calculated for each strategy $(\mathbf{a}_{i0}, \mathbf{a}_{i1}, \mathbf{a}_{i2,}..., \mathbf{a}_{iT-1})$. For discounting we assume that the operative total relevant costs occur at the end of period [t,t+1]. With the sets of constraints (10)-(13) the dependencies between the alternatives are taken into consideration. The objective function (9) is decomposable. Thus, the optimal supplier selection alternative $\mathbf{a}_{i0}^{*}$ at point of time t=0 as well as the strategies $(\mathbf{a}_{i1},..., \mathbf{a}_{iT-1})^{*}$, which are contingent on the realized states can be determined by applying standard stochastic dynamic programming techniques [10], [7].

It is easy to see that the model complexity as well as the solution time depend on the regarded number of (i) periods, (ii) the possible realizations of the expected plants specific demands in each period, and (iii) the sets of alternatives for supplier selection in each period. Regarding the example presented in section 3 in each period two of three potential suppliers could be selected and two possible realizations of the plants specific demands have to be considered. Therefore, in order to evaluate the realizable supplier selection strategies, for each period [t,t+1], with t=0,...,T-1, 6^{t+1} deterministic, single-period models have to be solved at the base-level. In order to reduce the solution time, these single-period models may be solved parallel on different computers using a standard optimization software. *Pibernik* [7] solves a similar problem of dynamic capacity configuration in a production network with more than four thousand models on the base-level within ten hours. However, using the presented hierarchical approach, the decision maker – the buyer – is able to calculate the optimal supplier selection alternative for different estimated demand scenarios. Therefore, our hierarchical approach can provide valuable decision support for the strategic supplier selection.

References

[1] De Boer L, Labro E, Morlacchi P (2001) A review of methods supporting supplier selection. In: European Journal of Purchasing & Supply Management 7: 75-89

[2] Dickson G W (1966) An analysis of vendor selection systems and decisions. In: Journal of Purchasing 2 (1): 5-17

[3] Ellram L M (1990) The supplier selection decision in strategic partnerships. In: Journal of Purchasing and Materials Management 26 (4): 8-14

[4] Huchzermeier A, Cohen M A (1996) Valuing operational flexibility under exchange rate risk. In: Operations Research 44 (1): 100-113

[5] Jayaraman V, Srivastava R, Benton W C (1999) Supplier selection and order quantity allocation: a comprehensive model. In: The Journal of Supply Chain Management 35 (2): 50-58

[6] Lewis T R, Yildirim H (2003) Managing switching costs in multi period procurements with strategic buyers. Working Paper, Duke University, Department of Economics. http://faculty.fuqua.duke.edu/bio/lewis/SwitchingIER.pdf

[7] Pibernik R (2001) Flexibilitätsplanung in Wertschöpfungsnetzwerken. Wiesbaden

[8] Reith-Ahlemeier G (2002) Ressourcenorientierte Bestellmengenplanung und Lieferantenauswahl. Leichlingen

[9] Schneeweiss C (2003) Distributed decision making. Second Edition, Heidelberg

[10] Schneeweiss C (1974) Dynamisches Programmieren. Würzburg

[11] Tempelmeier H (2002) A simple heuristic for dynamic order sizing and supplier selection with time-varying data. In: Production and Operations Management 11: 500-515

[12] Weber C A, Current J R, Benton W C (1991) Vendor selection criteria and methods. In: European Journal of Operational Research 50: 2-18

[13] Weber C A, Current J R, Desai A (1998) Non-cooperative negotiation strategies for vendor selection. In: European Journal of Operational Research 108: 208-223

Functional Analysis of Process-Oriented Systems*

Peter Buchholz and Carsten Tepper

Dept. of Computer Science IV, University of Dortmund, D-44221 Dortmund
e-mail:{peter.buchholz,carsten.tepper}@udo.edu

Abstract. A major problem in modelling and subsequent simulation of process-oriented systems (ProC/B models), is the functional correctness of the model. Therefore a model should be first analysed for its functional correctness before it is analysed by simulation. Petri nets are well suited for model based and state based functional analysis, but are often not adequate or not used for the specification of process models. We present in this paper a transformer for an automatic mapping from ProC/B models onto PNs. The resulting PN-models can be analysed with PN-algorithms and the results from the PN-analysis can be interpreted at the ProC/B level.

1 Introduction

To find optimal or at least good designs and strategies for large logistic networks often a model based approach is necessary. Usually the networks are modelled using some process oriented approach like process chains [5]. Ideally the model should be usable for the automatic analysis of performance, reliability and correctness of the logistic network. One formalism which allows an automatic mapping of the process chain model onto a simulator is the ProC/B formalism [1] which has been developed together with a corresponding toolset within the Collaborative Research Center 559 'Modelling of Large Logistics Networks'.

One major problem of modelling and subsequent simulation is the functional correctness of the model. Simulation can only hardly detect functional errors like deadlocks or livelocks in a system such that the simulative analysis of an incorrect model can result in strange and incorrect simulation results. Therefore a model should be first analysed for its functional correctness before the simulation is started. Unfortunately, process oriented descriptions like ProC/B often do not support functional analysis. On the other hand Petri nets are well suited for functional analysis of discrete event systems and a large number of analysis algorithms are available. Thus, a natural solution is to map the process chain model onto a PN. In this paper, we present an automatic mapping of ProC/B models onto Petri nets (PNs) show how different functional analysis algorithms can be applied to this model and indicate which results for the process chain can be derived from the PN and where are the limits of the approach.

The paper is structured as follows. In the next section we give a brief overview of the ProC/B formalism. Before introducing the translation of

* This research is supported by DFG, collaborative research centre 559 'Modelling of Large Logistic Networks'

ProC/B models onto PNs in section 4, some basic definitions for PNs are briefly introduced in section 3. Section 5 explains the functional analysis of ProC/B models via PN analysis. In section 6 we show an example of a store for a freight village and conclude the paper with a summary and an outline of future work.

2 ProC/B formalism

A modelling language which is especially designed to the needs of logistic networks is the ProC/B formalism, which is accompanied by a corresponding software toolset including a graphical interface for specification and a set of analysis tools [1]. Table 1 includes the core ProC/B modelling elements. The ProC/B formalism has proved its value in several industrial projects and it is the joint specification language in a large, interdisciplinary, and longterm collaborative research centre (SFB 559) at the university of Dortmund [1].

The main structuring elements of ProC/B-models are *Functional Units* (FUs), which encapsulate one or more *Chains.* A chain can be viewed as a structured and measured set of activities starting with *Sources* for process creations (demand), denoted by a circle with midpoint ⊙, followed by a chronological sequence of *Process Chain Elements* (PCEs) describing activities, denoted by arrow-like hexagons, and the chain is accomplished by a sink, denoted by ⊗. Horizontal connections between PCEs indicate a sequential behavioural pattern of processes. Branches into and merges from alternative sub-chains are allowed and represented by vertical bars. *Process Chains* can be hierarchically structured. PCEs may invoke sub-chains from so-called *subordinated FUs.* Subordinated FUs and sub-chains are described in the same manner as their super-ordinated FUs and chains (self-similarity). Actually, two types of subordinated FUs are distinguished: user-defined FUs and simple, predefined FUs of type *Server* or *Counter.* Servers are used to model active, possibly shared resources, i.e. machines, assembly lines, workers. In principle, servers correspond to single stations in queueing networks. Counters are used to describe passive resources, i.e. stores and waiting areas of usually restricted capacity.

3 Basic Definitions of Petri nets

First, a brief introduction to Petri nets (PNs) and some associated terminology is provided before we describe the translation of ProC/B onto PNs. Only some concepts which are relevant for understanding the rest of the paper are addressed. More complete material can be found in the literature e.g. [6].

A Petri net is a bipartite directed graph composed of places, drawn as circles, and transitions, drawn as rectangles. As usual, a directed graph is formally given by the description of its elements and functions or matrices specifying their interconnection.

Definition 1. A Petri Net (PN) is an 5 tuple $PN = (P, T, I^-, I^+, M_0)$, where $P = \{p_1, \ldots, p_n\}$ is a finite and non-empty set of places, $T = \{t_1, \ldots, t_m\}$

[1] see `http://www.sfb559.uni-dortmund.de/index.php?lang=eng`

ProC/B element	**Graphical Denotation**	**Semantics of Dynamic Behaviour**
Source	⊙	process creation
Sink	⊗	process termination
Process Chain Element (PCE)	arrow-like hexagons	activity involved in process, semantics: (stochastic) delay, resource demand/release
Process Chain	all denotations in between *Source-Sink* pair	description how process evolves over time
Server	rectangle with curved corners	familiar queueing station
Counter	trapezoid	familiar passive resource (semaphore etc.)
OR-Connector	vertical bar	probab./boolean branching into subprocesses, or combination of alternative subprocesses
AND-Connector	two vertical bars in parallel	synchronisation of incoming and outgoing processes
Functional Unit (FU)	rectangle	structuring element, a form of container, semantics given by included *Process Chains*, *Server*, *Counter* etc.

Table 1. Overview: Core ProC/B modelling elements: graphical denotation and semantics of dynamic behaviour

is a finite and non-empty set of transitions ($P \cap T = \emptyset$), $I^-, I^+ : P \times T \longrightarrow \mathbb{N}_0$ are backward, forward incidence functions, $M_0 : P \longrightarrow \mathbb{N}_0$ is the initial marking.

The initial marking M_0 is a special case of a marking $M : P \longrightarrow \mathbb{N}_0$. A marking M can be interpreted as an integer (row) vector which includes per place p one integer value describing the number of tokens on place p.

Definition 2. Let $PN = (P, T, I^-, I^+, M_0)$ be a Petri Net.

- Input places of transition t: $\bullet t := \{p \in P | I^-(p,t) > 0\}$,
- Output places of transition t: $t \bullet := \{p \in P | I^+(p,t) > 0\}$,
- Input transitions of place p: $\bullet p := \{t \in T | I^+(p,t) > 0\}$,
- Output transitions of place p: $p \bullet := \{t \in T | I^-(p,t) > 0\}$,

In the sequel we consider only nets where the undirected graph of the net is strongly connected. If also the directed graph is strongly connected, then the PN can potentially have a cyclic behaviour which means that it can be live (see the explanation below).

The dynamic behaviour results from the firing of transitions. A transition t is enabled in a marking M (denoted by $M[t>$) iff $\forall p \in P : M(p) \geq I^-(p,t)$. Any transition $t \in T$ with $M[t>$ can fire, changing the marking of any $p \in P$ to marking $M'(p) = M(p) + I^+(p,t) - I^-(p,t)$. This will be indicated by $M[t > M'$, $M[>$ describes the set of enabled transitions in M. The incidence matrix C of a PN is a $n \times m$ matrix with $C(p,t) = I^+(p,t) - I^-(p,t)$. Typical functional results for PNs include safety, boundedness and liveness. A net is safe if in every reachable marking the number of tokens at each place is bounded by one, it is bounded if the number of tokens is finite and it is live, if in every reachable marking every transition can eventually fire (i.e., a marking is reachable where the transitions is enabled). Especially liveness is important since non-live nets often indicate a design error.

For functional analysis two classes of analysis techniques exist. First, model based techniques that use the structure of the model for analysis. These techniques include invariant analysis and several specific techniques for restricted net classes. Alternatively, results can be derived from the set of reachable states in a so called state based analysis. State based analysis techniques allow a very detailed analysis of the functional behaviour, but are faced with the so called *state space* explosion indicating that the state space often grows exponentially with the size of the model such that even with contemporary computers and advanced data structured only small models with some hundreds of millions of states can be analysed [2].

The following classes of PNs are commonly distinguished and allow a more efficient models analysis (for details see [6]) which is, if possible, also exploited for the analysis of ProC/B models.

Definition 3. Let $PN = (P, T, I^-, I^+, M_0)$ be a Petri Net with $I^-(p,t)$, $I^+(p,t) \in \{0,1\}, \forall p \in P, t \in T$. PN is called a

1. State Machine (SM) iff $\forall t \in T : |\bullet t| = |t \bullet| \leq 1$.
2. Marked Graph (MG) iff $\forall p \in P : |\bullet p| = |p \bullet| \leq 1$.
3. Free Choice net (FC-net) iff $\forall p, p^{\text{‘}} \in P : p \neq p^{\text{‘}} \Rightarrow (p \bullet \cap p^{\text{‘}} \bullet = \emptyset$ or $|p \bullet| = |p^{\text{‘}} \bullet| \leq 1)$.
4. Extended Free Choice net (EFC-net) iff $\forall p, p^{\text{‘}} \in P : p \bullet \cap p^{\text{‘}} \bullet = \emptyset$ or $p \bullet = p^{\text{‘}} \bullet$.

4 Translation of ProC/B onto Petri nets

In the ProC/B toolset, an automatic mapping of ProC/B models onto PNs are employed in order to make existing functional analysis techniques for PNs available for ProC/B models. Along general lines, the mapping of ProC/B modelling elements onto PNs is fairly natural. Clearly, all core elements of the ProC/B formalism (Table 1) need to be expressed by PN places and transitions.

The general idea is to map a PCE to a place-transition pair in order to account for the number of processes that currently reside in the PCE by the marking of the place, and to account for processes that move away from a PCE by firing of the transition, cf. Figure 2. This construction allows us to map a sequence of PCEs to a PN in a straightforward manner, by connecting the transition of a PCE with the input place of its successor PCE.

The model mapping seems to be straightforward as long as we only focus on how core modelling constructs of the ProC/B formalism are mapped onto their corresponding PNs. However, the actual burden of implementation is to link together "local replacements" in order to achieve a consistent PN (assuming the ProC/B model is consistent) which has to be bounded whenever state based analysis techniques should be or have to be applied. Furthermore the resulting state space has to be manageable in this case. Many process chain models result in PNs which are unbounded, but by restricting the number of processes that can be active concurrently, the net becomes bounded.

Being aware of the consequences for state space dimension and since low level and not high level PNs are used, modelling of data and data dependent decisions is only possible in very limited and special cases (*Counters* are modified only by constant values, parameter and variables are excluded, data dependent decision needs to be randomised). This decision reduces the state space size significantly and seems to be mandatory for state based analysis. One could think of using high level PNs which allow a detailed modelling of parameters and decisions, but if model based or state based techniques are applied, then high level PNs are usually also mapped onto low level PNs. Nevertheless, one should keep in mind that the neglection of details results in a model which includes all possible behaviours of the process, but may include additional behaviour patterns that are not possible in the original system and possibly have to be ruled out by a more detailed analysis of the ProC/B model.

Another key decision in mapping *Process Chains* was whether each individual process is mapped onto a single safe PN or whether all processes of a *Process Chain* are mapped onto the same and the token population corresponds to the number of active processes. The latter solution reduces the state space size drastically and is therefore used. The price for this state space reduction is that fork/join structures cannot be described exactly since in a join operation the subprocesses arriving first and not the subprocesses resulting from the previous fork operation are joined. This problem was identified as Min-Match Problem in [4].

In the mapping each FU is described by an acyclic net where a pair of source and sink places describes one process behaviour pattern (see fig. 2). By joining the source and the corresponding sink place, the net becomes cyclic and boundedness and liveness can be analysed. By putting a finite number of tokens into the source/sink place, a finite number of parallel process incarnations is described. This description is similar to the specification of workflows by workflow nets [7]. However, in contrast to workflow nets, PNs resulting from ProC/B models are more complex and have to consider resorces which might be shared by different PCEs or FUs.

Clearly, our transformer turns out to be somehow restrictive regarding the subset of ProC/B models that can be treated or requires a more abstract model. Nevertheless, experience shows that many practically relevant ProC/B models fulfill all restrictions. Observe that all required restrictions can be statically checked efficiently at the ProC/B model level. The resulting PNs can be easily classified according to the net classes defined above. The intoduced restricted PN structures are observed only in some cases. Nevertheless, even if a complete ProC/B model does not result in such a restricted structure, it is not unusual that a PN for a FU results in such a restricted net.

5 Functional Analysis of ProC/B models via PNs

In this section we present the algorithm that is used for the functional analysis of ProC/B models (cf. algorithm 1). To reduce the effort of functional analysis, it is preferable to analyse the PCEs of the system in isolation to avoid state space explosion. As already mentioned only in some cases, the

PCE	PN class
Source	SM/MG
Sink	SM/MG
Server with one access	MG
Counter with one allocation and one release	MG
Opening and-connector	MG
Closing and-connector	MG
Opening or-connector	SM
Closing or-connector	SM

Table 2. List of live PCE and classification of the PN description of this PCE

whole PN has a specific simple structure such that liveness of the whole net can be decided from the liveness of the components. Nevertheless, in a first step model based analysis is used for all PCEs where it is possible.

Algorithm 1 *Functional analysis of ProC/B models via PNs*

```
if (ProC/B model contains only live PCE) // Model based analysis
 ProC/B model live;
else
  {
    if (ProC/B model can be translated onto PN)
    {
      Translate ProC/B model onto PN model;
      Check liveness of PN model; // State based analysis
      if (PN model is live)
        ProC/B model live;
      else
      {
        Transform PN trace onto ProC/B trace;
        Detect deadlock in ProC/B visualisation tool;
      }
    }
    else
      Functional analysis via PN impossible;
  }
```

For the model based analysis step of algorithm 1 we consider only the PN descriptions of PCEs. These descriptions can be classified in the PN classes SM, MG, FC-net or EFC-net and afterwards a liveness check for this PN class (cf. section 3 and [6]) can be done. Both the classification and the liveness check for the PN classes are realized in the APNN-Toolbox [2]. So if a model only consists of live PCEs, then the whole model is live because the connection of live PCEs doesn't change the liveness. Due to space restrictions we list in table 2 the live PCEs and the PN class in which the PN description of the PCEs falls.

If the ProC/B model contains one non-live PCE, then a translation onto a PN model and a state based functional analysis are necessary. First, the ProC/B model must be checked if a translation onto a PN model is possible (cf. restrictions of the translation, sec. 4). A state based functional analysis via PNs is impossible when the ProC/B model contains one PCE which cannot be translated by the transformer. In this cases we cannot decide if the ProC/B model is functional correct. State based functional analysis requires the generation of the whole transition system and reaches its limitations due to the size of the state space. Although advanced data structures are used in the APNN-Toolbox which allow us to analyse fairly large state spaces [3], many practically relevant models cannot be analysed at the state space level.

6 Example

This section demonstrates, how functional analysis can be applied to a ProC/B models of a Freight Village (FV). Fig. 1 shows the ProC/B model of the store.

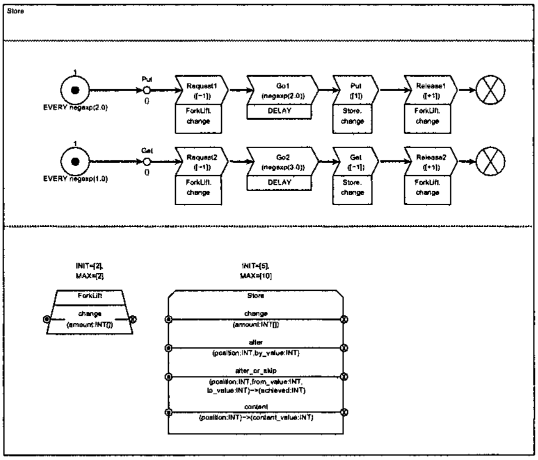


Fig. 1. ProC/B model 'Store' of a FV

This model contains two process chains. Processes which are created at the source of the upper process chain put goods into the store. The bottom process chain creates at the source processes which want to get goods out of the store. Both processes need the resource 'ForkLift' which is allocated before the processes can put or get goods and afterwards the resource is released. The model based analysis step at ProC/B level detects non-live PCEs so a translation onto a PN and afterwards a state based functional analysis is necessary. Fig. 2 depicts the PN model of the ProC/B model of fig. 1. The state based analysis of the PN model detects a deadlock in the PN model and generates a PN trace. This trace is a firing sequence of the transitions of the PN. The PN trace can be transformed onto a ProC/B trace which can be visualised by a ProC/B tool. The ProC/B trace can be observed in forward and backward direction, such that the deadlock in the ProC/B model can be located and modified into a functional error free model. The model in fig. 1 contains a deadlock, which can be easily explained. Processes in the upper

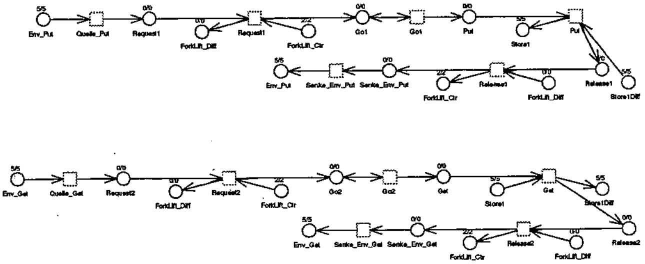


Fig. 2. PN model of the ProC/B model 'store'

process chain put goods into the store as long as the store has free capacity. When the store is full, then these processes are blocked inside the store. Before deblocking, goods have to be removed, but since processes with goods are blocked inside the store still holding the fork lifts, no new processes to remove goods can enter the store.

7 Conclusions

In this paper we present some results on the functional analysis of ProC/B models by using PNs and related analysis algorithms. A model based analysis approach checks the liveness of the system without translating the ProC/B model onto a PN. If the ProC/B model does not fall into the class of models which can be structurally analysed, it is translated onto a PN and analysed via state based analysis algorithms. In future research we plan to improve of the transformer from ProC/B onto PN to reduce the existing restrictions and we plan to investigate reduction rules to reduce the complexity of the resulting PN and also the size of the state space.

References

1. F. Bause, H. Beilner, M. Fischer, P. Kemper, and M. Völker. The Proc/B Toolset for the Modelling and Analysis of Process Chains. *In T. Field, P. G. Harrison, J. Bradley, U. Harder (eds.) Computer Performance Evaluation Modelling Techniques and Tools (Proc. Performance TOOLS 2002)*, Springer, LNCS 2324:51–70, 2002.
2. P. Buchholz, M. Fischer, P. Kemper, and C. Tepper. Modeling checking of CTMCs and Discrete Event Simulation integrated into the APNN Toolbox. *In F. Bause (ed): Tools of the 2003 Illinois International Multiconference on Measurement, Modelling, and Evaluation of Computer-Communication Systems*, 2003.
3. P. Buchholz and P. Kemper. Efficient computation and representation of large reachability sets for composed automata. *Discrete Event Dynamic Systems Theory and Applications 12 (3)*, pages 265–286, 2002.
4. S. Donatelli and J. Franceschinis. The PSR Methodology: Integrating hardware and software models. *in: J. Billington and W. Reisig (Ed.) Application and Theory of Petri Nets 1996*, Springer, LNCS 1091:133–152, 1996.

5. A. Kuhn. *Prozessketten in der Logistik, Entwicklungstrends und Umsetzungsstrategien (in German).* Verlag Praxiswissen, Dortmund, 1995.
6. T. Murata. Petri nets: properties, analysis and applications. *Proc. of the IEEE 77 (1989)*, pages 541–580.
7. W. M. P. van der Aalst. The application of Petri nets to workflow management. *The Journal of Circuits, Systems and Computer 8 (1)*, pages 21–66, 1998.

Finding Delay-Tolerant Train Routings through Stations

Gabrio Caimi, Dan Burkolter, and Thomas Herrmann

Institute For Operations Research, Swiss Federal Institute of Technology,
Clausiusstrasse 47, 8092 Zurich, Switzerland
{caimig, danb, herrmann}@ifor.math.ethz.ch

Abstract. Currently, many railway operators are increasing the frequencies of their trains. By condensing the timetable, routing trains becomes increasingly difficult as the chosen routes not only have to meet safety restrictions, but also guarantee some stability if delays occur.

We address the problem of routing trains through railway stations for a given timetable and outline two algorithms. The first algorithm searches for a feasible solution for the train routing problem based on an independent set modeling that is solved using a fixed-point iteration method. The initial solution is then amended by applying the second algorithm in order to increase the time slot of a chosen route, i.e. the time interval during which a train may arrive and find its designated route open. This algorithm is based on a local search optimization scheme.

Results showed that the fixed-point iteration found feasible solutions within minutes even for difficult cases, i.e. tight timetables. Though more time-consuming, the second algorithm allowed the average time slot length to be doubled, thus implying that it is possible to find routings which are more delay-tolerant. This helps to decrease impacts of late trains.

1 Introduction

We consider the problem of routing trains through stations. As many railway operators are currently decreasing the cycle-time of their periodic timetables, finding a feasible and good train routing will become increasingly difficult. In particular, the problem of scarce capacity will be accentuated in major stations where it is desirable to provide for connections. As a consequence, trains tend to arrive and leave during a short interval.

We assume for the problem that a timetable and a layout of the station are given. The goal then is to find for each train a route from an entry point to a leaving point of the station area via its platform. Zwaneveld et al. have studied this problem in [1], [2], and [3]. They show that the problem is NP-complete and propose an independent set modeling that they solve with a Branch and Cut algorithm. We apply the same modeling, yet propose a fixed-point iteration heuristic for solving the problem as some of our test-instances are too large for a Branch and Cut algorithm.

To our knowledge, other researchers have so far only addressed the question of finding feasible routings. However, by choosing a good routing, the impact of late trains on the timetable can be decreased. Ideally, the planned route for a train stays clear even if it has some delay and, as a consequence, operators need not intervene actively into the system. We introduce the concept of the time slot of a train route which shows how much time prior or later than the actual schedule a train may arrive and still use its planned route without coming into conflict with other running trains; see [4] for further information. We propose a local search optimization algorithm that increases the length of the time slots of routings starting from an initial feasible solution.

The paper is organized as follows. In section 2 an algorithm for finding an initial feasible solution and an algorithm for increasing the time slots are described. In section 3 computational results for a major station of the Swiss Federal Railways are given. Finally conclusions are drawn.

2 Model and Algorithm

We are given a set T of n trains each having a set $R_i = \{r_{i_1}, \ldots, r_{i_{m(i)}}\}$, $i = 1, \ldots, n$ of $m(i)$ possible routings connecting its entry and leaving point into the station area. Additionally, we have the conflict set C of incompatible routes, i.e. pairs of routes of different trains that at some place in the station topology use the same track segment at the same time (including safety time). We use the following notation for conflicting routes: $r_{p_q} \not\asymp r_{u_v}$. Analogously, $r_{p_q} \asymp r_{u_v}$ implies that the routes are compatible. Additionally, all routes of the same train are "in conflict" since only one route for each train is needed. Therefore, the conflict set is described by $C = \{\{r_{i_k}, r_{j_l}\} | i = j \vee r_{i_k} \not\asymp r_{j_l}\}$.

The routing problem can be seen as an independent set problem. The vertices are the elements of $R_1, \ldots, R_n$ and the edges correspond to the elements of C. Note that within the resulting graph all routes of the same train build a clique. A feasible solution to the problem is now given by a maximum independent set. A tight upper bound on the maximum cardinality is known due to the clique structure of the graph. The maximum cardinality is equal to the number of trains if and only if it is possible to find a conflict-free routing for all trains.

2.1 Finding an Initial Solution

In order to find a maximum independent set, we apply a specialized version of a heuristic for solving *Constrained Semi-Assignment Problems* [5]. The basic idea of the heuristic is to make a continuous relaxation of the boolean decision variables and then evolve starting from an interior point towards an extremal point corresponding to a feasible assignment. Burkard et al. present theoretical results for the general algorithm and show by empirical comparisons with Tabu Search that the algorithm finds solutions to the *Constrained*

Semi-Assignment Problem of similar target function value in less computation time [5]. The main advantages of this heuristic for solving the train routing problem are that it allows to exploit the clique structure of the graph and that many varying solutions can be found by including randomization (see below).

We introduce new variables p_{i_k} where $p_{i_k} = 1$ if train i receives its k-th route and $p_{i_k} = 0$ otherwise. Allowing all values $p_{i_k} \in (0, 1)$ we conduct the following fixed-point iteration:

Initialization. Choose for each p_{i_k} a value according to some distribution such that $0 < p^0_{i_k} < 1$ and $\sum_{k=1}^{m_i} p^0_{i_k} = 1 \;\; \forall i \in T$.

Iteration.

$$p^{t+1}_{i_k} := \frac{p^t_{i_k} \prod_{r_{j_l} \not\asymp r_{i_k}} (1 - p^t_{j_l})}{\sum_{s=1}^{m_i} p^t_{i_s} \prod_{r_{j_l} \not\asymp r_{i_s}} (1 - p^t_{j_l})} \quad i \in T \quad k \in \{1, \dots, m(i)\}$$

The intuition behind the iteration is as follows. The p_{i_k} can be interpreted as probabilities for finding a feasible solution to the routing problem when route k is chosen for train i. In each iteration step, the probability of selecting route r_{i_k} is adjusted by $(1 - p^t_{j_l})$ for all conflicting routes $r_{j_l} \not\asymp r_{i_k}$, i.e. the probability of choosing no route conflicting with r_{i_k}. The probability of choosing a route is also adjusted by the probability of not choosing the alternative routes of the same train due to the clique structure of the graph. Thus, there exists a feedback that increases the probability of choosing a likely route which speeds up convergence considerably. The denominator preserves $\sum_{k=1}^{m_i} p^t_{i_k} = 1 \;\; \forall i \in T$.

In theory, attractive fixed-points $p^{t+1}_{i_k} = p^t_{i_k}$ of the iteration correspond to solutions of the routing problem assuming that a solution exists. Yet, non-attractive fixed-points may exist which do not meet all restrictions, i.e. conflicting routes are chosen. This is shown for a more general setting in [6]. Due to limited precision of computing, fixed-points not corresponding to solutions occur in practice. Therefore, we stop iterating after a small number of iterations, say 200–400. The resulting values p_{i_k} are then used as probabilities to randomly choose routings for trains. A feasible solution is found if they are conflict-free. If no feasible solution is found after a bounded number of randomizations, a new fixed-point is calculated from a new starting point. Note that due to randomization many varying solutions may be generated.

2.2 Local Search Optimization

As already mentioned it is not good enough to have an arbitrary feasible solution. In a satisfactory solution, a planned routing for a train should be free even if it arrives with some delay in order to increase stability of railway operations. The aim is therefore to find routings that are not only conflict-free but also remain conflict-free for as large as possible perturbations of train arrivals or departures.

Assume that a solution L of the routing problem has been found. L is an assignment of conflict-free routes $r_i := r_{i_k}$; one for each train $i \in T$. We call the interval in which train i may arrive for its planned route r_i its time slot. The length is calculated for a given solution L assuming that all other trains are running exactly on time in the following way:

$$F_i(L) := \min_{j \in T, j \neq i} f(r_i, r_j) + \min_{h \in T, h \neq i} f(r_h, r_i)$$

where $f(r_i, r_j)$ is the amount of time train i may be delayed and still not be in conflict with the designated route of train j. Note that the value is symmetric in the sense that $f(r_i, r_j)$ stands also for the amount of time that train j may arrive prior to schedule if train i is on time. The goal is then to maximize a weighted sum of all time slots—weighted because primarily the shortest time slots should be increased. Formally:

$$\max_{L \in \mathcal{L}} z(L) := \max_{L = \{r_i | i=1,\ldots n\} \in \mathcal{L}} \sum_{s=1}^{n} \omega_s F_{\sigma(s)}(L)$$

where $\mathcal{L}$ is the set of all feasible solutions to the train routing problem and $\omega_1 \geq \cdots \geq \omega_n \geq 0$ are weights. $\sigma(s)$ is a permutation such that the time slots are brought into ascending order, i.e. $F_{\sigma(1)}(L) \leq \cdots \leq F_{\sigma(n)}(L)$. Therefore, the shortest time slot receives the largest weight in the target function.

It is not possible to introduce the length of the time slots as edge or node weights in the independent set graph since the time slot of a routing cannot be calculated before the complete routing has been made. As a consequence, the problem of maximizing time slots of routings cannot be directly modeled into the graph.

However, the sum of the time slots of a partial solution is always larger or equal to the value of a complete routing, i.e. an incomplete routing has an equal or better objective value. To state this more formally additional definitions are needed. U is called a subsolution to the routing problem with train set $T_U \subseteq T$ if all trains $i \in T_U$ have received conflict-free routings. Note, that it is not required that there exists an extension of the subsolution to a feasible solution of the routing problem. The objective value of a subsolution U is defined by

$$z(U) := \sum_{s=1}^{n} \omega_s F_{\sigma(s)}(U)$$

where

$$F_i(U) := \begin{cases} \min\left\{\min_{j \in T_U, j \neq i} f(r_i, r_j) + \min_{h \in T_U, h \neq i} f(r_h, r_i), \widehat{F}\right\} & \text{if } i \in T_U \\ \widehat{F} & \text{else} \end{cases}$$

Thus, $F_i(U)$ is the length of the time slot of train i in the subsolution if $i \in T_U$. The constant $\widehat{F}$ is some suitable upper bound on time slots such as the timetable period. It is needed to be able to compare two subsolutions.

We denote $U \subseteq V$ for two subsolutions if $T_U \subseteq T_V$ and all trains in T_U have the same routings in T_V, i.e. V is an extension of U.

We can now state the main lemma that is necessary for a local search optimization algorithm:

Lemma 1. *Let $U \subseteq V$ be two subsolutions. Then*

$$z(V) \leq z(U)$$

Proof: It is sufficient to show that $F_i(V) \leq F_i(U)$ $\forall i \in T$. We divide T into three parts: $T = T_U \cup (T_V \setminus T_U) \cup (T \setminus T_V)$.

$i \in T_U$:

$$\begin{aligned} F_i(U) &= \min_{j \in T_U, j \neq i} f(r_i, r_j) + \min_{h \in T_U, h \neq i} f(r_h, r_i) \\ &\geq \min_{j \in T_V, j \neq i} f(r_i, r_j) + \min_{h \in T_V, h \neq i} f(r_h, r_i) \\ &= F_i(V) \end{aligned}$$

$i \in (T_V \setminus T_U)$: $F_i(U) = \widehat{F}$ since $i \notin T_U$. On the other hand $F_i(V) \leq \widehat{F}$, therefore $F_i(V) \leq F_i(U)$.

$i \in (T \setminus T_V)$: i is neither in T_U nor in T_V, therefore $F_i(U) = F_i(V) = \widehat{F}$. ◇

Lemma 1 implies that the objective value will not increase by successively constructing a solution by adding trains. Hence, a subsolution gives an upper bound on the objective value of its extensions.

In order to be able to implement a local search optimization scheme a notion of solution adjacency is needed. We define a solution neighborhood as follows. Let $L \in \mathcal{L}$ then the *k-Neighborhood* is

$$N_k(L) := \left\{ \widetilde{L} \,\middle|\, \widetilde{L} \in \mathcal{L}, \, |\widetilde{L} \setminus L| \leq k \right\}$$

and L is an *k-optimal* solution if

$$z(L) \geq z(L') \qquad \forall L' \in N_k(L).$$

This neighborhood allows us to exploit Lemma 1. In addition, empirical evidence shows that usually only a small number of trains have a short time slot. Therefore, only a few trains need to be rerouted in order to substantially increase the target function value. Hence, good local optima can be expected already for a neighborhood size of $k = 1$ or 2.

We propose the following local search optimization:

Initialization. Choose a feasible solution $L \in \mathcal{L}$ and a neighborhood size k. Set `BestValue` $:= z(L)$ and $y := 1$.

Iteration. While $y \leq k$ create subsolution U by removing y trains from L. If $z(U) > z(L)$ then try to find extension L' of U such that $z(L') > z(L)$. If no such pair U and L' exists then increase $y := y + 1$; otherwise set the current solution to $L := L'$ and `BestValue` $:= z(L')$.

Lemma 1 is used when the objective value of the subsolution is compared with the value of the previous full solution. Extensions of the subsolution need only be considered if the subsolution has a better objective value than the full solution. As a consequence only a fraction of the neighborhood has to be examined.

3 Computational Results

The algorithms described in the previous section were implemented and tested for the station of Bern. Bern has 12 platforms and trains arriving from six major directions. The station region has a radius of roughly 5 km and has about 600 switches within. In the current timetable 19 trains pass Bern in half an hour each having an average of 290 possible routes while the maximum is 860. The tests were conducted on a Pentium 4 with a clock speed of 2.4 GHz.

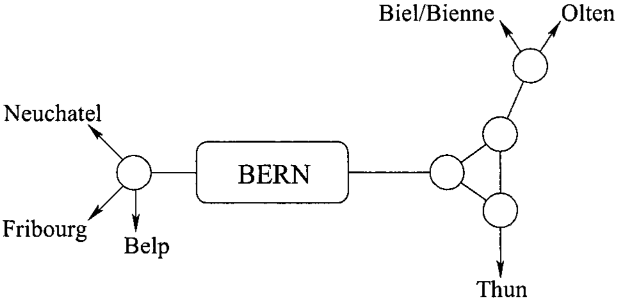


Fig. 1. Station area of Bern and its major directions.

First the fixed-point iteration for finding an initial solution was tested for the current timetable as well as for a condensed hypothetical timetable in which 16 trains pass Bern within a quarter of an hour. This timetable was specially constructed in order to test the limits of the system. In order to reduce the problem size the western and eastern part of the station region were computed separately. 200 fixed-point iteration steps were conducted prior to randomization.

	# Trains	# Routes	# Conflicts	Graph Density	Computation Time
Current Timetable					
East	19	5500	740 000	5%	< 1 min
West	19	1400	70 000	8%	< 1 min
Condensed Timetable					
East	16	6800	7 100 000	34%	40 min
West	11	1800	300 000	23%	< 1 min

Table 1. Problem size and computation time for initial solutions. Period time of current timetable 30 min and of condensed timetable 15 min. 100 test runs each.

Table 1 shows that the fixed-point iteration was able to find feasible solutions within reasonable time even in difficult scenarios (graph density for the condensed timetable > 30%). Usually, up to 50 restarts of the iteration were needed before the randomization led to feasible solutions for the scenario east side of Bern with the condensed timetable. For the other test instances, the first fixed-point iteration almost always led to feasible solutions.

In a next step tests were conducted to see if it is possible to increase arrival time slots for trains by changing initial routings applying the local search optimization heuristic. The following weight was chosen for the objective function: $\omega = (0.4, 0.3, 0.2, 0.1, 0, \ldots, 0)$, i.e. only the four shortest time slots were—with decreasing importance—considered.

	Initial Sol.	1-opt Sol.	2-opt Sol.	Improvement	Computation Time
Current Timetable					
East	19.2	89.6	149.2	680%	15 h
West	117.0	140.8	145.9	25%	8.5 h
Condensed Timetable					
East	1.0	2.1	2.2	120%	3 h
West	23.5	45.4	51.5	120%	2.5 h

Table 2. Average objective values in seconds. The average corresponds to the average of the four shortest time slots. Period time of current timetable 30 min and of condensed timetable 15 min. 20 test runs each.

Table 2 shows that it is possible to increase the average time slot length considerably. The longer computation time for the current timetable is a consequence of its lower density which implies that the solution space and the neighborhoods become larger. The very short time slots of the condensed timetable show that it indeed reaches the capacity limit of the current topology of the station of Bern.

Tentative tests seem to imply that the 2-optimum is already very near the global optimum since no or only tiny improvements are found by increasing the search space to the 3-neighborhood. Yet, computation time becomes prohibitive for conclusive investigations.

4 Conclusions

In this paper we considered the problem of routing trains with a given timetable through a station. Two algorithms were outlined and implemented. The first algorithm finds an initial solution for the train routing problem based on a fixed-point iteration. The second algorithm improves this initial solution based on a local search optimization scheme—improve in the sense that the designated routes for the trains remain open as long as possible even if train arrival in the station region is delayed. Tests showed that it was possible to find feasible solutions within seconds or minutes. Improving the routings was more time consuming, yet it was possible to double the time a designated route was open for a train. This implies that it is indeed possible to find a good routing which can help decrease impacts of singular delays on the overall timetable.

5 Acknowledgements

We would like to thank the Swiss Federal Railways for funding and providing data and in particular Thomas Graffagnino and Dr. Felix Laube for insightful discussions.

References

1. P.J. Zwaneveld, S. Dauzère-Pérès, S.P.M van Hoesel, L.G. Kroon, H.E. Romeijn, M. Salomon, and H.A. Ambergen. Routing Trains through Railway Stations: Model Formulation and Algorithms. In *Transportation Science*. 30(3):181–194, 1996.
2. L.G. Kroon, H.E. Romeijn, P.J. Zwanefeld. Routing Trains through Railway Stations: Complexity Issues. In *European Journal of Operational Research*. 98:485–498, 1997.
3. P.J. Zwanefeld, L.G. Kroon, S.P.M van Hoesel. Routing Trains through a Railway Station based on a Node Packing Model. In *METEOR, Research Memoranda*. 015, 1997.
4. G. Caimi. Routing mit Zielfunktion für den Bahnhof Bern. In *IFOR, Diploma Thesis*. 2004.
5. M. Burkard, M. Cochand, A. Gaillard. A Dynamical System Based Heuristic for a Class of Constrained Semi-Assignment Problems. In *Operations Research Proceedings*. 182–191, Springer 1998.
6. M. Cochand. A Fixed Point Operator for the Generalised Maximum Satisfiability Problem. In *Discrete Applied Mathematics*. 46:117–132, 1993.

Router: A Fast and Flexible Local Search Algorithm for a Class of Rich Vehicle Routing Problems

Ulrich Derigs and Thomas Döhmer

Department of Information Systems and Operations Research (WINFORS),
University of Cologne, Pohligstr. 1, 50969 Cologne, Germany
derigs@winfors.uni-koeln.de, thomas.doehmer@uni-koeln.de

Abstract. We describe a flexible indirect search procedure which we have applied for solving a special pick-up and delivery vehicle routing problem with time windows. The heuristic is based on an encoding of a solution as a sequence/permutation of tasks, a cheapest insertion decoding procedure, and, a threshold-accepting like local search meta-heuristic.

1 Introduction

During the last years there has been extensive research on extensions of the classical Vehicle Routing Problem (VRP) with respect to additional constraints which occur in real-world applications. The family of those "new" and difficult VRP-variants is often referred to as *Rich Vehicle Routing Problems (RVRP)*. Algorithms have been developed which build upon construction and improvement principles which have been used for the classical VRP already. For a survey of different classes of rich vehicle routing problems as well as traditional heuristic approaches see Toth and Vigo [3]. Our research on RVRP has been motivated by a real-world routing and scheduling problem which had nearly all of the complexities which are discussed in the literature: multiple time-windows, pick-up and delivery, heterogeneous fleet, multiple use of vehicles and multiple objectives.

2 Problem Description

The application for which we have developed a Decision Support System (DSS) is the daily routing and scheduling problem for a removal firm which, using a small (sub-) fleet of dedicated vehicles, supplies services like packing cases etc. and which arranges the reservation of parking lots in a metropolitan area. Yet, in contrast to the benchmark situations studied in the literature not all of the constraints are valid for all tasks. For instance, there are only some tasks which have time-windows and these time windows are of different nature, i.e. they can be a fixed time interval or they may exclude certain time intervals. Also, we have some tasks which are deliveries from the depot, some which collect goods and some which are pick-up and delivery tasks. Another significant difference from the standard models is that most of the services have to be scheduled on fixed days (in some cases within a specific time interval), but some specific services and service types could be scheduled flexibly several days ahead. Thus in the solution process we have to distinguish between optional and required tasks. Also, due to miscalculations on the demand and additional requirements, the plan has to be revised during the day, i.e. during operation.

Now a service plan consists of a number of routings which are associated to vehicles such that every task is contained in at most one route, every route is feasible with respect to capacity, time-window, coupling and precedence constraints. Also, every single route is feasible for the vehicle which it is assigned to and the set of routes assigned to a vehicle which represents the service plan for this vehicle and the associated crew are feasible in the sense that the vehicle is able to sequentially operate these routes with intermediate stops at the depot.

The quality of a service plan is evaluated by several criteria. Since the service is performed by a division which is a cost-center within the logistic firm these criteria are related to operational cost and service quality. Here we distinguish between so-called primary criteria and secondary criteria. Primary criteria are the distance and time for routing based on the geographic data, the waiting time at a customer, overtime cost, i.e. the cost if a vehicle/crew is operating more than its duty time, opportunity cost for not servicing a customer on the respective day and a fixed cost per tour which is assigned to a vehicle. Secondary criteria evaluate the "fairness" of a service plan and measure the difference between the longest and the shortest deployment time of the vehicles/crews and the difference between the largest and smallest service time of the vehicles/crews.

In our DSS-model these 8 criteria are aggregated into one objective function to measure the "cost" of a service plan. This objective function can be controlled by the user through appropriate parameterization over the DSS-dialog component, and the user can have the DSS construct alternative service plans under variations of the parameters. Thus we had to develop a rather flexible algorithmic approach as the core of a highly interactive DSS which is able to react on modifications with respect to incorporating constraints and objectives. In the following we shortly describe the solution approach for ROUTER, the solver component of the DSS.

3 The ROUTER-Heuristic

Respecting the complexity of the application and the necessity of the system to be able to generate alternate solutions in relatively short time it became obvious that the DSS had to be based on a heuristic which is outmost flexible with respect to possible combinations of criteria, i.e. configurations of the objective function and alternate requirements with respect to running time. The heuristic approach which we have implemented is an indirect search procedure (cf. Gottlieb [2]) which is based on

- a simple encoding of a service plan as a sequence/permutation of tasks,
- a cheapest insertion construction procedure which decodes the permutations into feasible service plans, and,
- a threshold-accepting like local search meta-heuristic.

3.1 The Cheapest Insertion Decoder

Let n the number of the required and optional tasks to be scheduled and p be a permutation of the numbers from 1 to n. Then the decoder-function constructs a (feasible) service plan using a deterministic insertion heuristic. Starting from the empty service plan the tasks are considered one by one. The cost value of the empty service plan is the sum of the opportunity cost values. The heuristic either inserts a task into an existing route at its best "insertion point" or opens a new tour with the task if by this insertion/opening the overall cost of the service plan is not increased. Otherwise the task remains unscheduled. Thus we generate the final solution through a sequence of (partial) service plans. Then this schedule/service plan, $X = X(p)$ say, can be evaluated with respect to the different criteria and we can assign an objective function value $C(p)$ say. Figure 1 summarizes this cheapest insertion strategy.

3.2 The Local Search Approach

Since the construction procedure is deterministic we can treat p as representation of X and the set P of all permutations of the numbers 1 to n represents the set of all service plans which can be constructed. Now we define a neighborhood function on P as follows. For $p \in P$ let $N(p)$, the neighborhood of p, be the set of all permutations $q \in P$, where in comparison to p two positions of numbers are exchanged. Now our local search procedure is a variant of the so-called *threshold accepting method* (cf. Dueck and Scheuer [1]): Let p be the actual permutation (solution) and p^* be the best permutation found so far. Then we randomly generate a neighbor solution $q \in N(p)$ and we accept q if

$$Z(q) < Z(p) + \text{THRESHOLD} \cdot Z(p^*) \tag{1}$$

with THRESHOLD being a fixed constant in the interval $[0, 1]$.

If this local search has failed to yield a better solution after a fixed number maxITER of trials then the local search procedure is restarted with another randomly generated initial permutation p. This is repeated a specified number RESTARTS of times. The local search procedure is summarized in Fig. 2.

Optionally, we apply a simple post-processing routine to improve the best solution found. This procedure works as follows: For every scheduled task we

```
Input: Permutation $p \in P$.
Output: Feasible service plan $X(p)$ and total cost $C(p)$.
Start with the empty service plan and $C(p)$ the sum of the opportunity costs
for every task $i$ in order of permutation $p$ do
  for every vehicle $l$ do
    if task $i$ may be scheduled on vehicle $l$ then
      Find the position(s) of task $i$ on vehicle $l$ which leads to the smallest cost increase, $\Delta_{il}$ say, with respect to operating cost, waiting cost and overtime cost
      if feasible position(s) found then
        If vehicle $l$ is not utilized yet, add vehicle cost,
        If a new tour has to be opened, add cost for an additional tour,
        Subtract opportunity cost for not serving task $i$,
      else
        $\Delta_{il} := \infty$
      end if
    else
      $\Delta_{il} := \infty$
    end if
  end for
  if $\min_l \{\Delta_{il}\} \leq 0$ then
    Insert task $i$ at the best position(s) found.
    $C(p) := C(p) + \Delta_{il}$
  end if
end for
return service plan and total cost.
```

Fig. 1. The ROUTER cheapest insertion decoder

check whether the objective function can be decreased by deleting the task from its tour and then reinserting it at its best insertion point. If the solution has been changed (improved) by such a reassignment then all unscheduled tasks are checked sequentially for feasible insertions by which the objective function value can be reduced further. If more than one such task exists, the one which leads to the highest decrease is chosen. This is repeated until no improving insertion is possible. Then the procedure restarts from the new service plan. The procedure stops if no such improvement is possible.

4 Implementation

The efficiency and flexibility of this approach is highly dependent on the implementation of the construction heuristic. For representing and manipulating service plans we maintain for every route a linked list of its locations in the order of service with the depot being the first and last entry in the list. For every location which is already scheduled in a route we calculate and

```
Input: $0 \leq$ THRESHOLD $\leq 1$, maxITER $> 0$, RESTARTS $\geq 1$
    a permutation $p \in P$, POST $\in$ {true, false},
Output: Feasible service plan $X(p)$ and cost $C(p)$.
 1: Set $p^* = p$
 2: Generate service plan $X = X(p)$ by calling the ROUTER-construction
    procedure and set $C = C^* = C(p)$.
 3: Set counter = 0
 4: repeat
 5:   Choose a random permutation $q \in N(p)$, i.e. swap two randomly chosen
      indices in $p$.
 6:   Generate service plan $X' = X(q)$ by calling the construction procedure
      and set $C' = C(q)$.
 7:   if $C' \leq (C + \text{THRESHOLD} \cdot C^*)$ then
 8:     Set $C = C'$ and $p = q$, i.e. accept permutation $q$.
 9:     if $C' < C^*$ then
10:       Set $C^* = C'$ and $p^* = q$
11:       Set counter = 0, i.e. reset counter
12:     end if
13:   end if
14: until counter++ < maxITER
15: if POST = true then
16:   Apply the post optimization routine POST to $X(p^*)$.
17: end if
18: if RESTARTS −− > 1 then
19:   Choose an new random permutation $p \in P$, generate $X = X(p)$, set
      $C = Z(X)$ and continue with step 3
20: end if
```

Fig. 2. The local search procedure

maintain values for the following additional attributes: the capacity-buffer for pick-ups, a time buffer giving the number of time units by which the arrival time at the location can be shifted forward such that for each location on the same route the arrival time remains within the time-window which is selected for this route, a time buffer giving the number of time-units by which the arrival time at the location can be shifted forward such that for each location on the same route the arrival time lies within at least one of the associated time-windows and the accumulated waiting time.

Based on this information the feasibility of an insertion and the consequences with respect to the criteria/the objective function can be calculated. When inserting (or deleting) a location into (from) a tour this information has to be updated. Note that when discussing a simple pick-up or simple delivery task one location has to be tested for insertion. If a pick-up and delivery task is discussed then two locations, the associated pd-pair, have to be tested simultaneously.

We have integrated the ROUTER-routine into a Decision Support System for solving the specific (rich) VRP of the removal firm. Here the flexibility of the solver allowed the user to decide on quick optimization versus long planning runs, i.e. from one run with the default permutation in which tasks for which finding a feasible position is probably difficult, i.e. tasks with few and narrow time windows and tasks which can only be served by a small number of vehicles, get earlier positions in the permutation than tasks for which finding a feasible position is probably easy, to more time-consuming intensive applications from many randomly generated start solutions. The routine showed to be quite effective since even with very low running time acceptable service plans could be generated. Dynamic aspects like additional tasks during operations can be supported in the DSS by freezing certain tours and calling the insertion heuristic.

The heuristic can easily be parameterized to solve special VRPs which are "poorer" in a sense. On the other hand the simple representation gives the DSS the flexibility to also incorporate additional constraints/measures associated with tours and routes by just modifying the insertion routine which plays somewhat the role of a work-horse in our approach.

References

1. G. Dueck and T. Scheuer (1990), *Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing.* Journal of Computational Physics 90: 161–175.
2. J. Gottlieb (2001), *Evolutionary Algorithms for Constrained Optimization Problems.* Shaker Verlag, Aachen.
3. P. Toth and D. Vigo (2002), *The Vehicle Routing Problem.* SIAM Monograph on Discrete Mathematics and Applications.

Integrated Optimization of School Starting Times and Public Bus Services

Armin Fügenschuh[1], Alexander Martin[1], and Peter Stöveken[2]

[1] Darmstadt University of Technology, Schlossgartenstr. 7, D-64289 Darmstadt
[2] BPI-Consult GmbH, Röttelnweiler 22, Alte Vogtei, D-79541 Lörrach

Abstract. In many rural areas, the public bus service is demand-oriented: By far the biggest group of customers are pupils who are transported to their schools within certain strict time limits. Usually, all schools start around the same time, which causes a morning peak in the number of deployed buses. However, schools are allowed to change their starting times within some interval. The question is, how to simultanenously rectify the starting times for all schools and bus trips in a certain county so that the number of scheduled buses is minimal. This problem can be formulated as a vehicle routing problem with coupled time windows (VRP-CTW), which is an extension of the vehicle routing problem with time windows (VRP-TW), where additional coupling constraints on the time windows are introduced. We give a mixed-integer programming formulation for VRP-CTW, and present solutions and lower bounds for randomly generated and real-world instances.

1 Introduction

In rural areas pupils on their ways to school and back home are usually the biggest group of customers for public means of transportation. In Germany it has become custom that schools start in a small time interval around 8:00. On the transportation side, this leads to a high number of deployed vehicles (i.e., buses) that have to transport the pupils in time, with a peak from 7:00 to 8:00. After having served the morning rush hour, most of the buses are sent back to the depot, for there is nearly no further demand. The afternoon peak at the end of school is usually much lower, for schools do not release all pupils at the same time. Hence the main focus for the optimization, i.e., a reduction of the total number of deployed buses, is the morning peak. Responsible for the transportation of pupils to their schools and back are the respective county administrations. Since they are funding bus companies with public money, they are interested in a cost-efficient organisation.

Changing the starting time of all schools in some county generally creates a strong opposition from the people affected by the change. To give an example, if some school doesn't start at, say, 7:50, but at 8:30, then all its pupils leave home more than half an hour later, which might cause troubles for working parents. In the afternoon, the children return later, and therefore might no longer be able to attend a sports club. Here the consulting company *BPI-Consult*, a subsidiary of the Finnish *Jaakko Pöyry*, enters the stage. Appointed from the county administration, BPI suggests new school starting times, accordingly adjusts the trip time tables, and plans new bus schedules using fewer buses than before. Then, BPI accompanies the whole process of embedding their solution into real-life, which includes negotiations with all participants (schools, bus companies, parents, county government officials) and potentially re-optimization, when new constraints appear that make previous solutions infeasible. Up to now, BPI successfully consulted four counties and one city, where the number of buses was reduced by 15 – 20%, which yields a yearly cost saving of 10 – 15%, see [7] for details. In each of these cases, the solutions were generated manually, which in the past turned out to be a tedious task for a human planner.

A wide range of transportation problems involving public bus transit, pupils and/or schools were already studied before, see [2–4], to name just a few. However, none of the presented models completely fits to our problem, mainly for some or all of the following reasons: In all previous modelling approaches, the time windows of school starting times are fixed and cannot be changed to save buses. Pupils are always transported directly to school, and changing the bus is not allowed. Some models focus on scheduling drivers, which is not an issue for us: Since our time horizon is small (mainly from 5:00 till 9:00), no breaks are needed, thus the drivers do not change the bus. Relocating bus stops, designing routes (trips) and assigning pupils to routes is sometimes part of the optimization, but for us these are input figures. BPI-Consult is not touching these issues, even if the number of deployed buses could be lowered in theory. From a mathematical point of view, this of course reduces the combinatorial complexity of the problem significantly. From BPI's point of view it is mainly a political decision: Changing only the starting times of trips and schools already creates enough opposition. The mathematical model we present in this article reflects these restrictions. It was designed to become part of a software tool supporting the planners at BPI.

2 A Mixed-Integer Programming Model

In general, our problem belongs to the class of vehicle routing problems with time windows (VRP-TW), see [5], for instance. In a generic formulation of this problem, a fleet of vehicles starting at a depot is sent to customers, picks up some commodities up to a maximum load and drives back to the depot. Each customer has a time window, in which the arrival time of the vehicle must be. In case of an earlier arrival of the vehicle, waiting is permitted. The lexicographic bi-criterial objective is: 1) minimize the number of deployed vehicles, and 2) minimize the total driving time (or distance) of the entire fleet. The model we developed to solve BPI's integrated planning problem is a modification of VRP-TW. We start with a detailed description of the sets and parameters needed as input figures.

2.1 Sets and Parameters

Let $\mathcal{S}$ be the set of all schools in the given county. For every school $s \in \mathcal{S}$, a time window $\underline{\tau}_s, \overline{\tau}_s \in \mathbb{Z}_+, \underline{\tau}_s \leq \overline{\tau}_s$, for the school starting time is given.

A *trip* t is a list of bus stops, where an arrival and a departure time is assigned to every bus stop of this list. Let $\mathcal{V}$ be the set of all bus trips in the county under consideration. The starting time of trip $t \in \mathcal{V}$, i.e., the time of departure at the first bus stop, is allowed to be shifted within the time window $\underline{\alpha}_t, \overline{\alpha}_t \in \mathbb{Z}_+, \underline{\alpha}_t \leq \overline{\alpha}_t$. The time a bus needs to serve trip t, i.e., the time difference between the departure at the first and the arrival at last bus stop, is denoted by $\delta_t^{\text{trip}} \in \mathbb{Z}_+$. We distinguish four different types of bus trips, called school-, feeder-, collector-, and free trips.

Let $\mathcal{P} \subset \mathcal{S} \times \mathcal{V}$, where $(s,t) \in \mathcal{P}$ if and only if trip t transports pupils to a bus stop of school s. In this case trip t is called *school trip* for school s. The driving time from the first bus stop of the trip till the school's bus stop is settled by parameter $\delta_{st}^{\text{school}} \in \mathbb{Z}_+$. For the pupils there is given a minimal and maximal waiting time $\underline{\omega}_{st}^{\text{school}}, \overline{\omega}_{st}^{\text{school}} \in \mathbb{Z}_+, \underline{\omega}_{st}^{\text{school}} \leq \overline{\omega}_{st}^{\text{school}}$, relative to the starting time of the school in which they must arrive at the school bus stop. The lower bound on this time interval $\underline{\omega}_{st}^{\text{school}}$ actually reflects the walking time from the school bus stop to the classroom in school, whereas the maximum waiting time $\overline{\omega}_{st}^{\text{school}}$ is specified by law.

Not all pupils arrive at their school by using only one trip. Let $\mathcal{C} \subset \mathcal{V} \times \mathcal{V}$, and $(t_1, t_2) \in \mathcal{C}$ if and only if there are pupils who start their journey with trip t_1 and then transfer to some other trip t_2 at a so-called *changing bus stop*. Then we call trip t_1 a *feeder trip* for *collector trip* t_2. The driving time for feeder trip t_1 from the first bus stop of the trip till the changing bus stop is settled by parameter $\delta_{t_1 t_2}^{\text{feeder}} \in \mathbb{Z}_+$, for collector trip t_2 this is $\delta_{t_1 t_2}^{\text{collector}} \in \mathbb{Z}_+$. The minimum and maximum waiting time at this bus stop is given by the time window $\underline{\omega}_{t_1 t_2}^{\text{change}}, \overline{\omega}_{t_1 t_2}^{\text{change}} \in \mathbb{Z}_+, \underline{\omega}_{t_1 t_2}^{\text{change}} \leq \overline{\omega}_{t_1 t_2}^{\text{change}}$. Note that a trip can have more than one type, p.e., it can be a school and feeder trip.

All other trips that are not school, feeder or collector trips are called *free trips*. Obviously free trips don't play a role for the transport of pupils. Nevertheless, they also have to be served.

When a bus finishes a trip, it either starts serving another trip or it is sent back to the depot. The connection of several trips that are served by the same bus is called *schedule* or *block*. The set $\mathcal{A} \subset \mathcal{V} \times \mathcal{V}$ consists of all pairs of trips that might be connected in some schedule. The intermediate trip from the last bus stop of trip t_1 to the first bus stop of trip t_2, where no passangers are transported, is called a *shift* or a *deadhead trip*. The driving time for a shift is denoted by $\delta_{t_1 t_2}^{\text{shift}} \in \mathbb{Z}_+$ for all $(t_1, t_2) \in \mathcal{A}$.

2.2 Variables and Bounds

For every bus trip $t \in \mathcal{V}$ we introduce a decision variable $v_t \in \{0, 1\}$ with $v_t = 1$ if and only if trip t is the first trip in some block. In the same manner,

$w_t \in \{0,1\}$ is a decision variable with $w_t = 1$ if and only if trip t is the last trip in a block. For the connection of two trips $(t_1, t_2) \in \mathcal{A}$ in a block we make use of a decision variable $x_{t_1 t_2} \in \{0,1\}$, with $x_{t_1 t_2} = 1$ if and only if some bus serves trip t_2 directly after trip t_1.

The starting time of trip $t \in \mathcal{V}$ (i.e., departure of a bus at the first bus stop of trip t) is settled by the integer variable $\alpha_t \in \mathbb{Z}_+$ which is bounded by the respective time window, $\underline{\alpha}_t \leq \alpha_t \leq \overline{\alpha}_t$. For every school $s \in \mathcal{S}$, the starting time is modeled by the integer variable $\tau_s \in \mathbb{Z}_+$ with $0 \leq \tau_s \leq \frac{\overline{\tau}_s - \underline{\tau}_s}{5}$. It is required that the new starting time is a multiple of 5 (minutes), thus the starting time of school s is given by $\underline{\tau}_s + 5\tau_s$.

2.3 Objective and Constraints

The main goal is the saving of buses, a secondary goal is to use the deployed buses in an efficient way and avoid long deadhead-trips where no customers are transported. Thus for sufficient big values of C_t, an objective function reflecting this goal can be stated as

$$z = \text{minimize} \sum_{t \in \mathcal{V}} C_t \cdot v_t + \sum_{(t_1,t_2) \in \mathcal{A}} \delta^{\text{shift}}_{t_1 t_2} \cdot x_{t_1 t_2}. \tag{1}$$

In a feasible solution, every trip must be served by exactly one bus. The trip can either be the first one in a block or it has a unique predecessor:

$$\sum_{(t_1,t_2) \in \mathcal{A}} x_{t_1 t_2} + v_{t_2} = 1, \qquad \forall\, t_2 \in \mathcal{V}. \tag{2}$$

Furthermore, it can either be the last one in a block or it has a unique successor:

$$\sum_{(t_1,t_2) \in \mathcal{A}} x_{t_1 t_2} + w_{t_1} = 1, \qquad \forall\, t_1 \in \mathcal{V}. \tag{3}$$

If trips t_1 and t_2 are connected, then trip t_2 can only start after the bus has finished trip t_1 and shifted from the end of t_1 to the start of t_2. Waiting is permitted, if the bus arrives there before the start of t_2. Using a sufficient big value for M, these constraints can be formulated in terms of linear equations:

$$\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1 t_2} - M \cdot (1 - x_{t_1 t_2}) \leq \alpha_{t_2}, \qquad \forall\, (t_1, t_2) \in \mathcal{A}. \tag{4}$$

So far, the presented formulation is a model for the the classical VRP-TW, where the time windows are static. Next, we extend this model with respect to restrictions due to school starting times and pupils who have to change the bus on their way to school. The new aspect, in contrast to the classical VRP-TW, is that the time windows for the trips are not longer fixed from the beginning, but depend on other time windows. For this reason, we call our problem *VRP-CTW*, where the character "C" represents the coupling aspect of the time windows.

For $(t_1, t_2) \in \mathcal{C}$, bus trip t_2 must arrive at the changing bus stop after trip t_1 within a small time window specified by $\underline{\omega}^{\text{change}}_{t_1t_2}$ and $\overline{\omega}^{\text{change}}_{t_1t_2}$. This synchronization is done by the following inequalities:

$$\begin{aligned}
\alpha_{t_1} + \delta^{\text{feeder}}_{t_1t_2} + \underline{\omega}^{\text{change}}_{t_1t_2} &\le \alpha_{t_2} + \delta^{\text{collector}}_{t_1t_2}, \quad &&\forall\ (t_1, t_2) \in \mathcal{C}, \\
\alpha_{t_1} + \delta^{\text{feeder}}_{t_1t_2} + \overline{\omega}^{\text{change}}_{t_1t_2} &\ge \alpha_{t_2} + \delta^{\text{collector}}_{t_1t_2}, \quad &&\forall\ (t_1, t_2) \in \mathcal{C}.
\end{aligned} \tag{5}$$

Moreover, we have to add the following inequalities to the model in order to synchronise the start of bus trips and schools:

$$\begin{aligned}
\alpha_t + \delta^{\text{school}}_{st} + \underline{\omega}^{\text{school}}_{st} &\le \underline{\mathcal{T}}_s + 5\tau_s, \quad &&\forall\ (s, t) \in \mathcal{P}, \\
\alpha_t + \delta^{\text{school}}_{st} + \overline{\omega}^{\text{school}}_{st} &\ge \underline{\mathcal{T}}_s + 5\tau_s, \quad &&\forall\ (s, t) \in \mathcal{P}.
\end{aligned} \tag{6}$$

Summing up, our MIP model consists of optimizing the linear objective function (1) subject to the set of linear constraints (2) – (6).

3 Input Data

Real-world input data for the model was provided by BPI-Consult. Testscenario `ct_1` is from a county where a manually generated solution was implemented in the last year. The county is located in Nordrhein-Westfalen, in the mid-west of Germany. The instance is too large to be solved to optimality with state-of-the-art MIP solvers (we used CPlex and XPress).

We developed a generator which is able to create random instances of any given size, while maintaining the characteristics of real-world data. Several sufficient small instances coming out of this generator were used to compare heuristic solutions with exact results of a MIP solver. Three of them, **rnd_1**, **rnd_2**, and **rnd_3**, are presented here. Table 1 gives an overview of the sizes of sets in the various instances. The number of buses (b) and the length of deadhead trips (d) for the current schedules (before the optimization) are shown in the last column of Table 1.

Table 1. The test instances

	$\mathcal{V}$	$\mathcal{A}$	$\mathcal{S}$	$\mathcal{P}$	$\mathcal{C}$	$b.d$
rnd_1	25	600	10	36	19	24.0002
rnd_2	25	600	10	35	15	25.0000
rnd_3	25	600	10	37	17	24.0002
`ct_1`	192	36,672	76	263	133	96.0382

4 Solving the Model

Checking feasibility for VRP-CTW with a fixed number of vehicles is $\mathcal{NP}$-complete, for it contains VRP-TW with static time windows as a special case, which is known to be NP-complete, see Savelsbergh [6]. Thus we cannot expect a polynomial algorithm for its solution, unless $P = NP$. We developed a greedy-type construction heuristic to quickly obtain good feasible solutions. For the small random instances the quality of the heuristic solutions is later compared with the exact values found by a MIP solver.

4.1 A Primal Heuristic

Our construction heuristic consists of two major steps. First, the selection of a suitable deadhead-trip, and second, a reduction of the state space by propagating all time windows onto each other.

In each step of our heuristic, a local "best" deadhead-trip is identified:

$$(t_1^*, t_2^*) = \operatorname{argmin}\{s_{t_1 t_2} : (t_1, t_2) \in \mathcal{A}\}, \tag{7}$$

where $s_{t_1 t_2}$ is a scoring function measuring how good two trips t_1, t_2 would fit together in some schedule. For example, setting $s_{t_1 t_2} := \delta^{\text{shift}}_{t_1 t_2}$ gives the nearest-neighbour insertion heuristic, which is well-known for TSP and VRP. However, due to the coupling of time windows, the selections made by the nearest-neighbour rule frequently lead to solutions of poor quality. We obtained much better results by the following scoring function:

$$s_{t_1 t_2} := \delta^{\text{shift}}_{t_1 t_2} + |\underline{\alpha}_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1 t_2} - \underline{\alpha}_{t_2}| + |\overline{\alpha}_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1 t_2} - \overline{\alpha}_{t_2}|. \tag{8}$$

The (heuristic) idea behind this is the following: There is no contribution of the 2nd and the 3rd summand to the score $s_{t_1 t_2}$ if and only if the time windows of trips t_1 and t_2 perfectly coincide in the sense that the time window for trip t_1 at the first bus stop of trip t_2 equals the starting time window of trip t_2, i.e., $\underline{\alpha}_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1 t_2} = \underline{\alpha}_{t_2}$, and $\overline{\alpha}_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1 t_2} = \overline{\alpha}_{t_2}$.

Then trips t_1^*, t_2^* are connected by setting $x_{t_1^* t_2^*} := 1$. Because of this, the time windows on the starting time of some trips and schools might change. Note that the inequalities (4) (for those $(t_1, t_2) \in \mathcal{A}$ with $x_{t_1 t_2} = 1$), (5), and (6) involving time variables α and τ are of a special structure: They form an IP2-system, i.e., an integer program where each inequality has exactly two non-zero coefficients. Inequality systems of this type where examined by several researchers in the past three decades. Recently it was shown that constraint propagation can detect infeasibility of IP2-systems in pseudo-polynomial time, see [1]. Thus, we either obtain by constraint propagation that (t_1^*, t_2^*) was an infeasible deadhead-trip (and thus we have to select and check the second-best), or we get strengthened bounds on the time variables.

In the latter case, further binary variables v, w, x can be fixed to their bounds. For example, if $\underline{\alpha}_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1 t_2} > \overline{\alpha}_{t_2}$ then $x_{t_1 t_2} = 0$. Moreover it is tested, if variables v, w can be fixed to 1 using inequalities (2) and (3).

The above steps are repeated iteratively, until all binary variables v, w, x are either fixed to zero or one. Then, we end up with a schedule for all buses and strengthened bounds on the time variables. In a final step, new starting times within these strengthened bounds are assigned to the schools and the trips, by solving the corresponding IP2-system with constraint propagation.

4.2 Dual Bounds

From a theoretical as well as from a practical point of view it is interesting to know how far away from optimality the heuristic solutions are. To answer this question, the dual bounds found by LP-relaxation turned out to be practically worthless. It is known that set covering/partitioning based reformulations for VRP-TW frequently lead to much better lower bounds in comparison to LP-relaxations. We adapted this reformulation technique for VRP-CTW.

The avarage trip duration is about 45 minutes. Since we focus on the morning peak from 5:00 till 9:00, the schedules of the buses typically do not contain more than four or five trips. Thus it is possible to enumerate all feasible schedules within reasonable time. The most time-consuming step within this enumeration is the feasibility-check of each schedule, for it involves each time the solution of a suitable IP2-system by constraint propagation. To obtain lower bounds for VRP-CTW, we then solve the LP-relaxation of the set partitioning problem

$$\begin{aligned} \min\ & c^T y \\ \text{s.t.}\quad & Ay = 1 \\ & y \in \{0,1\}^{\mathcal{R}}, \end{aligned} \tag{9}$$

where $\mathcal{R}$ is the set of all schedules, c_j is the cost of schedule $j \in \mathcal{R}$ (i.e., the cost for assigning a new vehicle plus the cost for all deadhead-trips within), and $A_{ij} \in \{0,1\}$ with $A_{ij} = 1$ if trip i belongs to schedule j. If the set $\mathcal{R}$ is too large (as in ct_1), we start with a small subset $\mathcal{R}' \subset \mathcal{R}$, and iteratively solve the LP-relaxation of (9), where after each iteration new columns from $\mathcal{R}$ are added to $\mathcal{R}'$, until all remaining columns in $\mathcal{R}$ have non-negative reduced cost. Note that integer solutions of (9) are in general infeasible solutions for VRP-CTW, since the local feasibility of each column (each schedule) does not automatically imply global feasibility, due to the coupled time windows.

5 Results

For each of the three random instances, we specified three different school starting time windows. In rnd_Xa, schools may start between 7:45 and 8:15, in rnd_Xb between 7:30 and 8:30, and in rnd_Xc between 7:15 and 8:45. The lower and upper bounds of the MIP solutions are given in the 2nd column of Table 2 (in the "*b.d*" form as in Table 1). If upper and lower bound differ, then the time limit of 3,600 sec. was reached (3rd column; CPlex9, 2.6GHz P-IV). Clearly, the wider the school starting time window, the lower the number of deployed vehicles. On the other hand, the wider the time windows, the more time is needed for solving the MIPs. Although the three random instances are more or less of the same size, their solution times differ significantly.

The solutions found by our primal heuristic are shown in the 4th column, the computation times are neglectably small (below 2 seconds each). Compared to the optimal solutions (where available), the heuristic is good enough for practical purposes. On the large real-world instance, it even outperforms the solution found by the MIP solver, while using much lesser time.

The solutions of the LP-relaxation of (9) in the 5th column clearly improve the LP-relaxation of the VRP-CTW model, especially for the real-world instance ct_1. The last column of Table 2 shows the total number of schedules in $\mathcal{R}$, and the number of columns in $\mathcal{R}'$ used within the column generation.

Table 2. Computational results

	MIP (upper – lower)	time	heuristic	dual bound	columns (used)
rnd_1a	19.0052 (optimal)	1	20.0050	17.5069	67 (48)
rnd_1b	13.0292 (optimal)	27	15.0216	10.4022	377 (69)
rnd_1c	10.0343 (optimal)	124	12.0319	10.0221	863 (84)
rnd_2a	16.0083 (optimal)	4	18.0040	14.0051	143 (44)
rnd_2b	11.0155 (optimal)	947	15.0098	9.2427	647 (55)
rnd_2c	10.0170 – 6.0394	3,600	12.0137	7.3874	2348 (62)
rnd_3a	14.0101 (optimal)	37	15.0092	11.5610	215 (75)
rnd_3b	11.0139 – 9.0195	3,600	12.0134	9.0202	939 (81)
rnd_3c	10.0158 – 5.0324	3,600	10.0188	9.0155	3,340 (84)
ct_1	87.0633 – 9.9167	3,600	82.0715	49.9008	5,090,907 (784)

References

1. Bar-Yehuda R., Rawitz D. (2001), Efficient algorithms for integer programs with two variables per constraint. *Algorithmica* **29** (4), 595 – 609.
2. Bowerman R.L., Hall G.B., Calamai P.H. (1995), A multi-objective optimisation approach to school bus routing problems. Transp. Research A **28** (5), 107 – 123.
3. Braca J., Bramel J., Posner B., Simchi-Levi D. (1997), A Computerized Approach to the New York City School Bus Routing Problem. IIE Transactions **29**, 693 – 702.
4. Corberan A., Fernandez E., Laguna M., Marti R. (2000), Heuristic Solutions to the Problem of Routing School Buses with Multiple Objectives. Technical Report TR08-2000, Dep. of Statistics and OR, University of Valencia, Spain.
5. Cordeau J.-F., Desaulniers G., Desrosiers J., Solomon M., Soumis F. (2002), VRP with Time Windows. In: Toth P., Vigo D. (Eds.), The Vehicle Routing Problem. SIAM Monogr. on Disc. Math. and Applications. SIAM, Philadelphia.
6. Savelsbergh M. (1986), Local search for routing problems with time windows. Annals of Operations Research **4**, 285 – 305.
7. Stöveken P. (2000), Wirtschaftlicherer Schulverkehr: ÖPNV-Optimierung mit erfolgsabhängiger Honorierung. Der Nahverkehr **3**, 65 – 68. (In German).

A Decision Support Framework for the Airline Crew Schedule Disruption Management with Strategy Mapping

Yufeng Guo

Decision Support & OR Laboratory, and
International Graduate School of Dynamic Intelligent Systems,
University of Paderborn,
Warburger Str. 100, D-33098 Paderborn, Germany
Email: guo@dsor.de

Abstract. Disruption management for airline crew schedules is important for the airline industry, since an increasing amount of disruptions to the regular operations occur frequently. The emphasis of this task is put on quickly obtaining one or more reasonable, at best optimal, recovery solutions from current disruptions, which has to be achieved within an acceptable time period. In this work, we propose a decision support framework that combines exact optimization methods and meta-heuristics for solving real-life practical problems. An exact method, based on a Column Generation type of procedure, is studied and tested, while a dedicated Genetic Algorithm working with a local improvement procedure provides the capability to solve the problem alternatively. Notably, a so-called *strategy mapping* procedure is applied to customize solution methods.

1 Introduction

Due to the uncertain operating environment, numerous factors potentially influence the actual operations, such as aircraft breakdowns, severe weather, crew sickness, air congestion etc. As a result, frequent perturbations of flight and crew schedules occur everyday in airlines. *Crew Recovery* hence is the process by which airlines react to disruptions, and recover the crew schedule in a way that minimal 'impact' is produced by the disruption situation.

Similar to the airline crew scheduling (CSP) process, the recovery attempts to reschedule all the flights that are disrupted by irregular events. However, the airline Crew Recovery Problem (CRP) has been studied comparably little from a research point of view. Lettovský et al. [4] developed a framework using a pairing generation method that incorporates special branching strategies. A systematic study has been conducted by Stojković et al. [7], in which they solved it as an integer nonlinear multi-commodity network flow model with time windows and additional constraints. In addition, Yu et al. [8] described an award-winning real-life application employed by Continental Airline in the U.S., which treated the problem as a set covering problem, and a so-called *generate-and-test* heuristic was applied for roster generation. Moreover, some studies about airline irregular operations are also discussed in [3] and [5].

In this approach we propose a decision support framework. It provides a bundle of methods. A preprocessing procedure is developed to identify the suitable strategy, and customize it for the given situation. It shows that the selected strategy produces normally better solutions, and the solution time can be reduced.

In Section 2 we describe the airline CRP in detail and the relevant model is presented. The framework and solution methods are described in Section 3. In Section 4 a case study is presented to test the mapping strategy and the proposed solution methods.

2 Crew Recovery Problem

The process associated with the CRP takes care of disrupted situations in which original crew schedules require several, sometimes major, modifications to keep the airline's operations running after an unplanned occurrence. When disruptions happen, a series of flights have to be delayed and even canceled, and additional crews and flights are required in order to have enough resources to service all the flights that need to be operated.

2.1 Problem Description

At least three kinds of resources must be recovered during a disrupted time period: aircraft, crew, and passengers. Each of them implies large impact to the new schedule. For example, a shortage of aircrafts will result in not only unexpected delays and cancelations, but also some additional difficulties which makes the crew recovery more difficult to solve. Due to the complexity of the problem, common approaches will normally be decomposed into several sequential sub-problems, whereas each of them will handle with one resource type respectively. The way to decompose the problem differs from airline to airline regarding the heterogeneous company rules. One reason for applying a sequential approach is the fact that a complete integration is unrealistic from a practical point of view. However, a tightly integrated approach is obviously the work that needs to be done in the near future, because of allowing interactive collaboration among them.

2.2 Problem Formulation

In this approach the airline CRP is formulated as a set partitioning type model, where a set of affected flights caused by disruptions needs to be assigned or reassigned. These disrupted flights grouped with previously planned flights (flights set F) are chained into a set of rosters R, which represent all the possible individual schedules for a set of crew members W within a certain time period. Each crew member, therefore, will be finally assigned at most one revised schedule for the examined period with respect to all the regulations and rules. In our approach, we do not create pairings as traditional CSP does, i.e., the problem is solved without addressing pairing generation prior to assignment phase. Thus we present a simplified model as below:

c_i^w, operational cost of roster i assigned to crew member w

u_f, additional cost, if the flight f is assigned to a standby crew member

v_i^w, bonus of assigning a roster i to crew member w, where the roster i stays unchanged as planned or few changes are made

a_{fi} = 1, if flight f is included in the roster i, 0 otherwise

b_{iw} = 1, if roster i belongs to crew member w, 0 otherwise

The decision variables are:

x_i^w = 1, if roster i is assigned to crew member w, 0 otherwise

y_f = 1, if flight f is assigned to a standby crew member, 0 otherwise

Therefore, the model can be expressed as:

$$\text{min.} \sum_{w\in W}\sum_{i\in R}(c_i^w - v_i^w)x_i^w + \sum_{f\in F} u_f y_f \tag{1}$$

$$\text{s. t.} \sum_{w\in W}\sum_{i\in R} a_{fi}x_i^w + y_f = 1 \;\; \forall f \in F \tag{2}$$

$$\sum_{i\in R} b_{iw}x_i^w \leq 1 \;\; \forall w \in W \tag{3}$$

The objective function denotes minimizing the total operational cost c_i^w and the additional cost u_f for those flights which are assigned to standby crew members. Furthermore, it minimizes the variation from the planned schedule by calculating changes v_i^w in the monetary sense. Certain bonus is attached to the rosters that are same or similar to their originally scheduled ones. Constraint 2 guarantees that each flight is covered exactly once, while constraint 3 ensures that each crew member takes at most one roster.

3 Solution Methods

As mentioned earlier, this type of problem is considerably difficult to solve due to the complexity of the problem itself and its real-time aspect. Only relatively small-sized problems are suitable to be solved directly by standard solvers, because of the huge amount of variables. A number of approaches follow the idea of column generation to implicitly construct the model. Consequently, the total solution time can be reduced to some extent. However, the difficulty still remains when large instances are being solved.

As shown in Figure 1, in order to avoid intensive computations a preprocessing procedure is implemented to identify a strategy based on an evaluation of disruptions. The chosen strategy hereby customizes the eligible method

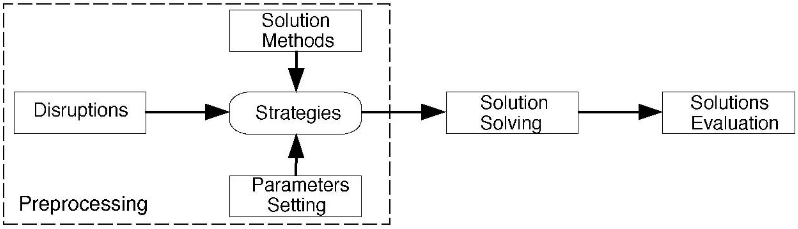


Fig. 1. General recovery process

with dedicated parameters. This makes it possible to reduce the problem size ahead of the solving.

3.1 Exact Optimization based Method

As introduced above, a Column Generation based method starts with a initial set of rosters as the basis, and solves it as a LP relaxation of the original IP. Then those rosters with negative reduced cost are gradually inserted into the basis until no more such roster available. Finally, the IP solution is obtained through ILOG CPLEX 8.0 [2] or a Branch-and-Bound method. In our approach, a constrained shortest path is implemented to achieve the task of roster generation. Accordingly, a multi-layer time space network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ is constructed, where departure and arrival events (nodes) are connected by flights (arcs) which are operated at multiple home bases (layers).

Generally, regarding the size of the problem two criteria may impact the performance: the length of the *recovery period* (the period within which the schedule needs to be recovered), and the number of the home bases involved in the process (i.e., how many crew members involve). The discussion of these criteria is given in Subsection 3.3.

3.2 GA-based Heuristic Method

In our approach, we customize the conventional genetic algorithm into a hybrid genetic algorithm which includes a set of heuristics with knowledge about this specific problem, together with a so-called *Local Improvement* procedure. This acts as a supplementary local search to the genetic algorithm. A two dimensional representation is developed which shows the direct assignment(see

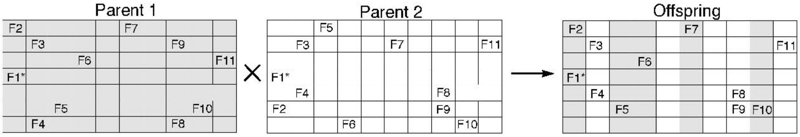


Fig. 2. *Column-based crossover* operator

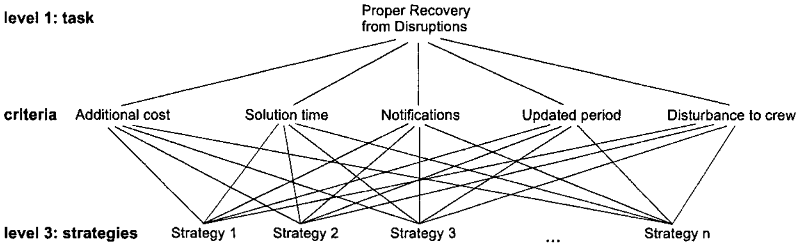


Fig. 3. AHP hierarchy for strategy mapping, with 5 criteria and n strategies

Figure 2). Various variation operators are implemented based on such encoding, such as *Row-based Crossover*, *Column-based crossover* (Figure 2), swapping mutation etc. Further details can be found in [1].

The performance of the algorithm is subject to numerous parameters. For example, the chosen variation operators and their corresponding probability have dramatic impacts. Additionally, a proper initialization of the population provides a faster convergence. The discussion of determining these parameters is described in next subsection.

3.3 Strategy Mapping

Many decision making processes involve preferential selection among a finite set of alternatives or courses of action. To handle such a situation, the Analytic Hierarchy Process (AHP) was developed by Saaty [6] to provide the prioritization of alternatives through evaluation of a set of criteria elements.

In the airline CRP, the main criteria are: *Additional cost*, the extra money that needs to be paid for repairing the schedule disrupted; *Solution time*, the computational time to find the final solution; *Notifications*, the number of crew members that need to be notified because of schedule updates; *Updated period*, the duration of the period starting from the time of the first updated flight to the last one; *Disturbance to crew*, the number of changes and unbalanced workload among crews.

There are several factors that influence the decision of choosing a proper strategy. Regardless the reason which causes disruptions, the following measurement may give ideas how serious a given situation is:

- C_d, the number of delayed flights
- C_c, the number of canceled flights
- C_n, the number of new flights that need to be operated
- C_m, the number of available crew members
- C_a, the number of daily flights in average

A finite strategy set S is built based on the experience of a huge amount of experiments. Basically, two strategy groups are considered, Column Generation based and GA based strategy. The finally selected strategy can be understood as the combination of solution methods and relevant parameters. Table 1 gives some example strategies, e.g., the strategy CGLO uses the Column Generation method, and it recovers the disrupted schedules of those home bases affected by disruption for a 5 days long recovery period.

Table 1. Example strategies for solving CRP

Strategies:	CGLO	CGLA	CGSA	GALO-SS	GALA-SS-LI
	.Col-Gen	.Col-Gen	.Col-Gen	.GA	.GA
Key aspects:	.LRP[a]	.LRP	.SRP[b]	.LRP	.LRP
	.OHBs[c]	.AHBs[d]	.AHBs	.OHBs	.AHBs
				.Solution Seeding[e]	.Solution Seeding
					.Local Improvement

[a] Long recovery period
[b] Short recovery period
[c] Only those home bases affected by disruptions
[d] All home bases in the instance
[e] Seed some individuals representing original schedule with little changes

With given criteria above, one can establish a hierarchy with 3 levels (Figure 3). By investigating occurred disruptions, the pairwise comparisons are conducted between each two criteria with respect to their importance or influence to the final decision. A criteria matrix $A = [a_{ij}]$ is built with the weight ratios a_{ij}. The normalized principal right eigenvector $\hat{C}$ of A represents the priority of those criteria. Then we compare the strategies on each of the criteria by examining how efficient one strategy may handle the given problem. Same to the number of available strategies, n strategy matrices $W_i, i = 1, 2, ..., n$ are produced, and the priority vector $\hat{W}_i, i = 1, 2, ..., n$ for each strategy can be calculated in the same way described above. The final priority vector $\hat{W} = [\hat{w}_i, i = 1, 2, ..., n]$ can be thus calculated as

$$\hat{W} = [\hat{W}_1\ \hat{W}_2\ \hat{W}_3 ... \hat{W}_n]\hat{C} \tag{4}$$

The highest priority value denotes the proper strategy for the examined situation. Even though the final result may still remain subjective, the most reasonable strategy can be chosen if the matrices are built by carefully setting those weight values.

4 Case Study

A case study is presented to demonstrate the strategy mapping and proposed solution method. The instance is from a European airline with multiple home bases inside Germany and more than 30 destinations spreading around Europe. In this case study, we consider three criteria: additional cost, solution time and notifications. Only two strategies are used for the purpose of simplicity, e.g., CGLA and GALO-SS-LI. Furthermore, a disruption scenario is given with following characteristics: $C_d = 2$, $C_c = 1$, $C_n = 2$, $C_m = 188$ and $C_a = 85$.

Table 2. Pairwise comparison matrix of criteria

	additional cost(AC)	solution time (ST)	notifications (N)	priority
AC	1	1/3	5	0.38
ST	3	1	5	0.54
N	1/5	1/5	1	0.08

Table 3. Pairwise comparison matrix of strategies on the criteron AC

AC	CGLA	GALO-SS-LI	priority
CGLA	1	2	0.667
GALO-SS-LI	1/2	1	0.333

Table 4. Pairwise comparison matrix of strategies on the criteron ST

ST	CGLA	GALO-SS-LI	priority
CGLA	1	1/3	0.25
GALO-SS-LI	3	1	0.75

Table 5. Pairwise comparison matrix of strategies on the criteron N

N	CGLA	GALO-SS-LI	priority
CGLA	1	1/3	0.25
GALO-SS-LI	3	1	0.75

The pairwise comparison matrix of criteria can be seen in Table 2, and the normalized priority vector, therefore, is $\hat{C} = [0.297, 0.617, 0.086]$. The comparisons between strategies are done based on criteria, with one matrix being created for each criterion (see Table 3, 4 and 5). With the above method (4), we calculate the final priority vector $\hat{W} = [0.371, 0.626]$ which clearly shows that the second strategy GALO-SS-LI is superior to CGLA by reflecting the priorities among the introduced criteria. After a variety of experiments were conducted, the results show that the strategy GALO-SS-LI produced a slightly worse solution than that the CGLA could find. But the solution time is dramatically reduced into the acceptable period of time, around 3 minutes.

5 Conclusion

Effective disruption management which enables efficient recovery is subject to various factors, and very cost intensive. On the one hand, powerful algorithms are needed to solve such complex problem. Some proper strategies should be also developed on the other hand, especially for situations where there is no method available that can easily find the 'perfect' solution, and where several criteria need to be considered simultaneously.

Besides our work on developing powerful algorithms solving the airline CRP, we present a strategy mapping method using AHP. It indicates the ability to differentiate a variety of solution methods with distinct parameters, and calculates their priority measurement in numerical values. Finally the selection of the strategy becomes easy and obvious. Noticeably, one prerequisite is the building of those precise comparison matrices, which may not be easily achieved. However, a large number of experiments may give one the insight how to set those weight values appropriately.

References

1. Y. Guo, L. Suhl, and M.P. Thiel. Solving the airline crew recovery problem by a genetic algorithm with local improvement. Technical report, WP0408, at DS&OR Lab., University of Paderborn, Germany, 2004.
2. ILOG. *Cplex v8.0 User's Manual.* ILOG, France, 2002.
3. M.E. Irrgang. Airline irregular operations. In D. Jenkins and C.P. Ray, editors, *Handbook of airline economics*, pages 349–365. McGrawHill Aviation Week Group, New York, 1st edition, 1995.
4. L. Lettovský, E.L. Johnson, and G.L. Nemhauser. Airline crew recovery. *Transportation Science*, 34(4):337–348, 2000.
5. J.M. Rosenberger, E.L. Johnson, and G.L. Nemhauser. Rerouting aircraft for airline recovery. *Transportation Science*, 37(4):408–421, 2003.
6. T.L. Saaty. *The Analytic Hierarchy Process.* McGraw-Hill, New York, 1980.
7. M. Stojković, F. Soumis, and J. Desrosiers. The operational airline crew scheduling problem. *Transportation Science*, 32(3):232–245, 1998.
8. G. Yu, M. Argüello, G. Song, S. M. McCowan, and A. White. A new era for crew recovery at continental airlines. *Interfaces*, 33(1):5–22, 2003.

Tail Assignment in Practice

Mattias Grönkvist[1] and Jens Kjerrström[2]

[1] mattias@carmensystems.com
[2] jensk@carmensystems.com, Carmen Systems AB, Odinsgatan 9, SE-411 03
Göteborg, Sweden

Abstract. Tail Assignment is the problem of assigning specific aircraft to flights, producing a fully operational, robust schedule which fulfills operational constraints, while minimizing a cost function. The costs are usually related to robustness and quality of the solution, e.g. to penalize short times between flights and reward "crew-friendly" connections. Carmen Systems' Tail Assignment system utilizes a combination of optimization methods and constraint programming to solve this problem. An interesting aspect of the system is that it can solve several fleet types at once, moving towards an integration of the traditional two-step approach of first solving the Fleet Assignment problem and then Tail Assignment. The article intends to explain some of the methods used.

1 Introduction

Resource optimization is becoming increasingly more important for airlines. Much work has been done in the crew planning area, e.g. Crew Pairing, and Crew Rostering, see [1] for references. This article will focus on a less explored problem in aircraft planning, the Tail Assignment problem.

A typical planning process for an airline begins with the construction of the timetable. The market is analyzed in order to find attractive destinations and times. After the timetable is done, it is decided which fleet should operate each flight, this process is called Fleet Assignment. In Fleet Assignment, among other things, the expected number of passengers for a specific flight is taken into consideration. A fleet with enough seats to accommodate all passengers has to be chosen, but the number of empty seats should be minimized. After Fleet Assignment is done, the crew can be planned, since the type of the aircraft decides what kind of crew is needed. This is done in two steps, Crew Pairing and Crew Rostering. Roughly, in Crew Pairing anonymous building blocks, called *pairings*, are created. Crew Rostering then creates a schedule, called *roster*, for each crew member. Tail Assignment, the process of creating a schedule for each aircraft, also called *route*, can be done at any stage after the Fleet Assignment is done, but is in many cases done late in the planning process. Tail Assignment is similar to the Crew Rostering step, in the aspect that schedules are created for specific individuals.

2 Tail Assignment

This section describes the Tail Assignment problem (TAS). For a thorough description and references, see [1]. TAS is the problem of constructing routes for specific aircraft, which are identified by a number on the tail fin, hence the name of the problem. In other processes of airline scheduling, as Crew Pairing and Fleet Assignment, the objective function is clearly associated with money. In TAS the costs are rather defined in terms of robustness and quality of the solution. It is for example desirable to avoid very short or medium time connections between flights. The short connections make the schedule sensitive to disruptions, while the medium time connections make the aircraft stand on the ground waiting for the next flight, but unable to do any flights in between. Longer connections makes it possible to use the aircraft as a *standby*, i.e. it can perform extra activities in the event of disruptions. If Tail Assignment can interchange information with Crew Pairing, it is possible to avoid unnecessary aircraft changes for the crew, making the schedule less sensitive for disruptions. An example where a where a good TAS solution is clearly related to direct savings, is when certain aircraft are leased, thus making them more expensive to use. It is possible to make flights more expensive for these aircraft, hence the optimizer will try to avoid them. The cost for leasing an aircraft is in the order of thousands of dollars per flight hour.

The most important operational constraints are maintenance constraints, which are not handled in the Fleet Assignment step. Two different kinds of maintenance are considered; *minor* and *major*. The major maintenance activities are planned in advance and given as input to TAS. They usually require that the aircraft is on the ground for a long period of time and occur at long intervals, e.g. every 500 days the aircraft needs a thorough check. The major maintenance is a *preassigned* task, an activity that is fixed to a tail and the optimizer is not allowed to change. The minor maintenance activities are not fixed in time, but require that the aircraft get maintenance at some point in a certain interval. Minor maintenance typically requires less time on ground, often one night at the home base, and occur at more regular intervals, e.g. every 100 flight hours. Another typical requirement is that the aircraft should return to the base after a number of calendar days, in order to take care of minor defects. In addition to the maintenance regulations there are constraints on individual aircraft. Some aircraft might not be allowed to land at certain airports due to e.g. noise level.

2.1 Model

TAS can be modeled as a set partitioning problem.

$$\min \sum_{r \in R} c_r x_r \tag{1a}$$

$$\sum_{r \in R} a_{fr} x_r = 1 \quad \forall f \in F \tag{1b}$$

$$x_r \in \{0,1\} \tag{1c}$$

R is the set of all routes respecting the rules, c_r is the cost of route r. F is the set of all *activities*, i.e. flights or group of sequential flights. a_{fr} is 1 if activity f is included in route r. x_r are binary decision variables that state if route r is part of the solution or not. Thus the columns in the coefficient matrix are routes, the rows correspond to flights. The partitioning constraints 1b state that each flight must be assigned to exactly one aircraft and that each aircraft can have only one route. Each aircraft must have at least one preassigned task in the period, representing the starting position of the aircraft, called a *carry-in activity*. The carry-in is included in every route for a vehicle. The constraint for the carry-in makes sure that an aircraft gets at most one route. To allow for unassigned tasks, routes including only one activity can be added to R, at a high penalty. These artificial routes can be interpreted as slack variables. The problem with this model is the size of the set R, which grows exponentially with the size of F. Problems with thousands of flights and about 80 aircraft are common for medium sized carriers, so enumeration is clearly not an option. Luckily the *column generation* method aims at solving large problems without explicitly enumerating all the variables.

3 Solution Methods

Column generation is chosen as the primary solution tool for the Carmen Systems' [3] Tail Assignment optimizer. It can be seen as an generalization of the simplex method. Instead of having all the columns present, and exchanging columns between the sets of basic and non-basic columns, the idea is to begin with an initial set of columns, and thus dual values, somehow generate one or more columns with negative reduced cost, re-evaluate the dual values and continue until no more columns can be generated. Column generation was first described by Dantzig and Wolfe in [2]. Column generation is solving an LP, so it is necessary to relax the binary decision variables to continuous variables between 0 and 1. After a sufficient degree of optimality is reached with column generation, one is left with the task of finding a good IP solution from the, in most cases, fractional solution. The LP containing the columns generated so far, and responsible for calculating dual values is called the *restricted master problem*, since it contains a subset of the columns. The part responsible for generating columns from the dual values is called the *pricing* *problem*. For the restricted master problem a standard commercial LP solver is used. The pricing problem is solved using a specialized algorithm, which will be described in the next section.

3.1 Pricing Problem

The costs in Tail Assignment are not so complicated, which allows us to make the assumption that costs are only defined on the level of activity and connections between activities. A way to describe the basic feasibility of routes is to use a *connection network*. The nodes in the connection network correspond to activities, while the arcs represent if one activity can follow directly after another. The connection network models the basic constraints in TAS, e.g. legal connection times between activities. The additive cost function resulting from the assumption on cost structure, in combination with the nice structure of the connection network (it is a directed acyclic graph), paves way for use of some *shortest path* algorithm for the pricing problem. A shortest path is the path between some nodes that has the lowest cost, defined as a sum of arc and node costs over the path. Each activity in F is corresponding to a row in the restricted master problem, so each activity will have a dual value. Each activity is also present in the connection network, along with costs for nodes and arcs. If the node costs are modified with the dual values, a path with negative cost through the connection network would give a negative reduced cost column. If no such path can be found, it would mean that the restricted master problem is solved to optimality. Since the shortest path problem can be solved in polynomial time, this scheme seems appealing. The purpose of the pricing problem is though to produce promising members of R. For a route to be legal it must fulfill the maintenance constraints, something which unfortunately is not captured by a pure shortest path formulation. The maintenance constraints can be modeled by introducing *resources* in the network. Each arc will not only have a cost, but also resource consumption associated to it. The consumption of a resource is not allowed to grow beyond a certain limit. The pricing problem is then a variation of the resource constrained shortest path problem (RCSPP), which is NP-hard. In a RCSPP it is normally not possible to reset the resource consumption, something that happens in the TAS case when maintenance is possible over a connection. It is important to solve the pricing problem swiftly, since it will be solved many times.

A standard method of solving shortest path problems is a dynamic programming algorithm referred to as *label setting*, see [6]. At each node a list is kept of the best paths so far. When the destination node is reached, the best path will be in the list. An entry in the list is called a label, and corresponds to one path. For the pure shortest path there will be only one label at each node, since it only makes sense to keep one cheapest path. When resources are involved, the number of labels at each node can grow dramatically. Each label has an entry not only for the cost, but for each resource. It is in many cases impossible to say that one intermediate path is better than another, since a path with good cost might become illegal at a later step. We use a combination of *dominance*, a criterion to determine if one label never can produce a path better than another, and heuristic decisions on how many labels are kept at each node, and at the destination node. Note that in the labeling case, the resetting of resources is not a problem. If maintenance is possible on a connection, the resource consumption will be reset to some appropriate value. Experiments has been made with an model where the resetting were made explicit, to allow for Lagrangian relaxation, see [4]. Unfortunately the added complexity made this model difficult to use in practice.

3.2 Reducing the Problem Size

The number of arcs in the the connection network grows very rapidly as the number of activities increases. The size of the connection network will naturally affect the performance of the pricing problem. One way of reducing the size of the network is to forbid very long connections, under the assumption that they wont be used anyway. There is some risk in such an approach, since the long connections might be needed in a solution. Also, it is a very simple assumption, which does not capture any redundant patterns in the connection network. In order to obtain a larger reduction of the network, without removing potentially useful arcs some pre-processing algorithms called *filters*, since they filter away connections, were developed. The *aircraft count filter* locates points in time when there are no aircraft on ground at an airport. This is done simply by looking at arrivals and departures. If such a point is found, there can be no connections between flights arriving before this point and flights departing after this point, if all flights are to be covered. This can be very powerful, depending on the problem structure. In some cases over 90% of the original connections are removed. The filtering procedure can have effect on the master problem too, since if connections are reduced so only one outgoing or incoming arcs exist for some flight, this connection must be used, and the two flights can be treated as one. This removes one constraint from the restricted master problem. In some cases, the number of activities are reduced by 60% by such aggregation of flights. Also, the aircraft filter can be applied *before* any connections are generated, avoiding unnecessary calculations altogether.

A constraint programming model of TAS is also available, see [5]. This can be used to solve TAS on its own, but also as a filter. The model demands that each flight has a unique follower, which is enforced by the `all_different` constraint. Efficient algorithms exist for this type of constraint. The constraint is propagated and the variable domains are reduced. Each flight will have a variable which domain is the possible following flights. Only connections between the flight and flights in the domain of its variable can be used in a solution. This *propagation filter* is very effective. Other heuristic filters also exist. The filtering significantly decreases the size of the problem, leading to better convergence and better solutions.

3.3 Integer Heuristic

When a good enough solution has been obtained for the restricted master problem through column generation, this is often fractional. One possibility is to go for branch-and-bound, but this has some serious disadvantages. Branch-and-bound can take a lot of time, since in the worst case the entire search tree might have to be explored. In the case of TAS not many variables have values above 0.5, which makes the situation even worse. Secondly, the columns created for the restricted master problem are for an LP. There is nothing that guarantees that they are good also in the IP case. A huge gap would exist between the restricted master solution and the final binary solution. Branch-and-price is the combination of branch-and-bound and column generation. At each node in the search tree the LP is solved to optimality through column generation. Branch-and-price guarantees the optimal IP solution, but has been decided too costly for TAS. The problem is more operational, proven optimality is not desired at any cost; shorter run times are prioritized. In order to avoid a huge gap between the restricted master problem and the IP solution, columns need to be generated in the integer phase. Instead of the possibly full exploration of the search tree through branch-and-price, different *fixing heuristics* have been developed. In the basic forms they are greedy in nature, since something is fixed and only the fixed branch is explored, without going back.

Variable Fixing. The first method is fixing variables. This means that an entire route is fixed for one aircraft. The fixing decisions are made on several criteria, e.g. primal value or the sum of the duals of the flights in the column. Columns that cover flights in the fixed column are removed from the restricted master problem. Connections to/from flights in the fixed column are removed from the pricing problem, and new columns are generated. Another column is then fixed, until the solution is integral. The downside of this method is that it is too greedy, in the end all flights often can not be covered. The fixing decisions are made on primal value.

Connection Fixing. Fixing connections is a more careful method than variable fixing. This can easily be seen as fixing a column is equivalent to fixing all the connections between the flights in that route. A connection corresponds to two flights and thus to two rows in the coefficient matrix. These rows are checked, and the values of the variables for routes that have both flights in it are added up. If this value is greater than some limit, the connection is fixed. The limit that decides if a connection should be fixed can vary during the fixing process, going from a value close to 1 and downward.

This gives significantly better results than the variable fixing, most often all flights are assigned in the end and often the lower bound is reached. The downside is it takes more time than the variable fixing.

Hybrid Fixing. From the sections above, the conclusion is that variable fixing is quick, but too greedy, and that connection fixing gives good quality solutions, but can take too much time. The thought of combining them and get the best of both worlds is tempting, and in this case rewarding. Since the variable fixing is most aggressive, it is done first. Some percentage of the variables are fixed initially, then connection fixing is performed until an integral solution is found. This approach gives a good trade-off between solution time and solution quality.

4 Solving Large Instances

The most common way to run the Tail Assignment optimizer is that it first creates an initial solution, which is subsequently improved. The output is not one single solution in the end of the run, but several solutions are output during the run. In this way the run can be monitored, to see if e.g. things are modeled correctly. This way of working also helps in solving larger instances. The typical planning period for TAS is one month. Solving the entire period with full column generation takes too long time for the larger instances arising at mid-size airlines (approximately 80 aircraft and 6000 flights), especially since it is preferred to get an initial solution as soon as possible. Instead an initial solution is created with an "aggressive" setting on the column generation process. After a full solution has been obtained, the full period is divided into smaller periods, which are improved sequentially with the full column generation setting. The aggressive setting involves e.g. heuristic connection filtering based on constraint programming, reducing the connection network substantially.

5 Mixing Several Fleet Types

The solution to the Fleet Assignment problem is a partitioning of the flights to different fleets. Tail Assignment in then traditionally done for each of these fleets. In our Tail Assignment model, not only the constraints, but also the costs can be aircraft specific, so nothing prevents Tail Assignment from being applied to several fleets at the same time. The system is used for this purpose in production. The Fleet Assignment is done, but then Tail Assignment is performed simultaneously for up to four fleets. Since Fleet Assignment does not consider all operational constraints, it is possible that the Fleet Assignment solution is impossible to operate. By planning several fleet at the same time introduces a way to get a feasible Tail Assignment solution. Another scenario is that the passenger demand for a flight can change as the departure date approaches. Fleet Assignment costs could then be incorporated into TAS and allow for *re-fleeting*, that a change of fleet type is demanded.

6 Rule Language

A special rule language called *Rave*, Rule And Value Evaluator, developed by Carmen Systems AB, is used to define all rules and costs. There are no rules hard-coded in the optimizer code and Rave is treated as a black box. This gives the user full freedom to try different scenarios, to see the impact on the solution if e.g. the minimum connection time between some activities are changed, or the maintenance interval for a minor maintenance is changed.

7 Conclusion

We have described parts of the Carmen Systems' Tail Assignment system. It can model constraints and costs on individual aircraft, which makes it possible to capture operational constraints, as well as doing Fleet Assignment decisions in Tail Assignment. The main solution approach is column generation, several refinements of the basic scheme were discussed. The system is used for production runs for two medium sized carriers.

References

1. Grönkvist M. (2003) Tail Assignment–A Combined Column Generation and Constraint Programming Approach. Lic. Thesis, Chalmers University of Technology, Gothenburg
2. Dantzig G. B., Wolfe P. (1961) The Decomposition Algorithm for Linear Programs. Econometrica. 29(4):767-778
3. Carmen Systems AB. `http://www.carmensystems.com`
4. Kjerrström J. (2003) A Model and Application of the Resource Constrained Shortest Path Problem with Reset Possibilities. Master's Thesis, Chalmers University of Technology, Gothenburg
5. Kilborn E. (2000) Aircraft Scheduling and Operation–A Constraint Programming Approach, Master's Thesis, Chalmers University of Technology, Gothenburg
6. Ahuja R.K., Magnanti T.L. and Orlin J.B. (1993) Network Flows. Prentice Hall, New Jersey

Ein praxistauglicher Ansatz zur Lösung eines spezifischen D-VRSP-TW-UC

Oliver Kunze

c/o PTV AG, D-76131 Karlsruhe, Stumpfstr. 1; Oliver.Kunze@ptv.de

Abstract

In der Regel wachsen die Transportkosten auf der letzten Meile mit der Enge des Lieferzeitfensters. Dies stellt insbesondere Anbieter vor ein Problem, die Aussagen über einen präzisen Liefertermin bei Bestellung abgeben müssen, obwohl noch nicht alle zu bedienenden Lieferungen bekannt sind.

Im folgenden Beitrag wird das Problem als D-VRSP-TW-UC-DO modelliert. Motiviert durch die raum-zeitliche Analyse der Solomoninstanzen wird der Begriff der Gleichzeitigkeit als wesentlicher Kostentreiber herausgearbeitet. Anschließend wird ein praxistaugliches Verfahren (ORA) vorgestellt, das es ermöglicht, eine hohe Avisgenauigkeit bei Bestellung zu erlangen. Schließlich werden typische Ergebnisse des Verfahrens dargestellt und diskutiert.

Einleitung

Mit Hilfe von Call-Centern und Internet-Bestellportalen ist es heute möglich, Güter online zu bestellen. Dabei ist es derzeit die Regel, dass die Ware online bestellt, und per Post oder KEP-Dienstleister zugestellt wird. Ein genaues Lieferzeitavis findet in der Regel nicht statt.

In manchen Bereichen (z.B. im online-Lebensmittelhandel) ist jedoch eine präzise Lieferauskunft zum Bestellzeitpunkt kaufentscheidend. Die Avisierung eines engen Zeitfensters bereits bei Annahme der Bestellung schränkt jedoch den zeitlichen Freiheitsgrad in der Tourenplanung erheblich ein, und hat somit i.d.R. signifikante Auswirkungen auf die Zustellkosten, wie im Folgenden gezeigt wird.

Problemdefinition

Zu lösen ist ein Tourenplanungsproblem mit Zeitfenstern mit der Besonderheit, dass die Aufträge sukzessive bzw. dynamisch eingehen, und weder die Lage noch die Anzahl der zu beliefernden Kunden ist ex ante bestimmt ist.

In Anlehnung an (Bent u. van Hentenryck 2003) lässt sich das Problem somit klassifizieren als dynamisches Tourenplanungsproblem mit Zeitfenstern und stochastischen Kunden mit Lieferzeitzusage zum Bestellzeitpunkt, oder kurz als D-VRSP-TW-UC-DO (dynamic vehicle routing and scheduling problem with time windows, uncertain customers and delivery notes at time of order placement). Außerdem kann angenommen werden, dass die Anzahl *m* der verfügbaren Fahrzeuge für die Lebensmittelheimzustellung beschränkt ist, da es sich hier i.d.R. um kleine Spezialfahrzeuge (mit 3 Temperaturzonen für Trocken-, Frische- und Tiefkühlsortiment) handelt, die nicht kurzfristig zugemietet werden können. Die Einhaltung der zugesagten Lieferzeitfenster kann durch die Definition von entsprechenden Nebenbedingungen (hier Lieferzeitfensterrestriktionen) hinreichend abgebildet werden. Es wird weiterhin die Annahme getroffen, dass die Anzahl der möglichen Lieferungen mit Lieferzeitzusagen zum Bestellzeitpunkt bei vorgegebener Anzahl *m* an Fahrzeugen zu maximieren ist (Zielfunktion). Diese Annahme ist dadurch motiviert, dass jede Lieferung einen Verkaufsertrag bringt[1]. Formal läst sich das Problem somit folgendermaßen definieren. Es sei:

n: Anzahl der besuchten Kunden

$E(n)$: Erwartungswert der Anzahl der besuchten Kunden

x_{ij}: Binärvariable zur Nachfolgerbeziehung mit
$x_{ij}=1$, falls Kunde j unmittelbar nach Kunde i besucht wird
$x_{ij}=0$, sonst

fz_{ij}: Fahrzeit von Kunde *i* zu Kunde *j*

sz_j: Bediendauer des Kunden *j*

d: Präzision der Lieferzeitzusage

D: Toleranz, d.h. der Kunde akzeptiert eine Lieferung gdw.
$ta_j \in [\, tp_j-D;\ tp_j+D]$

t: ein beliebiger Zeitpunkt

ta_j: Avisierter Lieferzeitpunkt beim Kunden *j*

$tf_j =$ ta_j-d: Früheste Ankunftzeit beim Kunden *j*

$ts_j =$ ta_j+d: Späteste Ankunftzeit beim Kunden *j*

t_j: Anfangszeitpunkt der Belieferung j in Bezug auf Tourstartzeit $t_0 = 0$

tp_j: bevorzugter Lieferzeitpunkt von Kunde *j*

$to(j)$: Zeitpunkt der Bestellung (order placement)

M^{∞}: sehr große Zahl

V: Menge aller zu besuchenden Kunden incl. Depot (Index 0); $/V/=n+1$

[1] Zwischen unterschiedlichen Erträgen pro Lieferung wird hier nicht unterschieden, da es langfristig darum geht, eine hohe Kundenbindung und einen großen Kundenstamm zu erzielen. Die Maximierung von Einmalerträgen tritt demgegenüber in den Hintergrund.

Q: beliebige Teilmenge von Kunden; $Q \subseteq V\text{-}\{0\}$
m: Anzahl verfügbarer Fahrzeuge
T(t): Anzahl Touren, die zu *t* bereits begonnen haben, aber nicht beendet sind
dur: maximale Einsatzzeit eines Fahrzeugs
X^t: Menge der unbekannten Aufträge zum Zeitpunkt *t*
$V\text{-}X^t$: Menge der bekannten Aufträge zum Zeitpunkt *t*
wz_i: Wartezeit beim Kunden j

$$\max E(n)\; u.d.N. \tag{1.1.}$$

$$\sum_{j=0}^{n} x_{ij} = 1 \quad \mathit{für}\; i = 1,\ldots,n \wedge \sum_{i=0}^{n} x_{ij} = 1 \quad \mathit{für}\; j = 1,\ldots,n \tag{1.2.}$$

$$\sum_{i\in Q}\sum_{j\in Q} x_{ij} \le |Q| - 1 \quad \mathit{für\ alle\ Teilmengen}\; Q \subseteq V - \{0\}\; \mathit{mit}\; |Q| \ge 1 \tag{1.3.}$$

$$tf_j \le t_j \le ts_j \quad \mathit{für}\; j = 1,\ldots,n \tag{1.4.}$$

$$t_j \ge t_i + sz_i + fz_{ij} - (1 - x_{ij})M^{\infty} \quad \mathit{für}\; i = 0,\ldots,n\; \mathit{und}\; j = 1,\ldots,n \tag{1.5.}$$

$$t_j + sz_j + fz_{j0} - (1 - x_{j0})M^{\infty} \le dur \quad \mathit{für}\; j = 1,\ldots,n \tag{1.6.}$$

$$T(t) \le m \quad \mathit{für}\; t \ge t_0 \tag{1.7.}$$

$$\left.\begin{cases} to(j) > to(i) \Leftrightarrow j \in X^{to(i)} \;\wedge \\ to(j) \le to(i) \Rightarrow j \in V - X^{to(i)} \end{cases}\right\} \forall i,j = 1\ldots n \tag{1.8.}$$

$$\left.\begin{cases} t = to(j) \Rightarrow ta_j \in [tp_j - D\,; tp_j + D] \;\wedge \\ t > to(j) \Rightarrow \quad t_j \in [ta_j - d\,; ta_j + d] \end{cases}\right\} \forall j = 1\ldots n \tag{1.9.}$$

(1.2.) gewährleistet, dass jeder Kunde genau einmal verlassen bzw. erreicht wird.
(1.3.) gewährleistet, dass keine Kurzzyklen ohne Depot entstehen.
(1.4.) gewährleistet, dass die avisierten Zeitfenster eingehalten werden.
(1.5.) gewährleistet, dass „zeitlich konsistente Routen“ entstehen, die sowohl Fahr- (fz), und Stand- (sz) als auch implizit Wartezeiten (≥) beinhalten können. Diese Bedingung greift nur für die Kundenpaare (i,j), die direkt hintereinander angefahren werden (d.h. $x_{ij}=1$). Für diese Kundenpaare wird der letzte Subtrahend $(1\text{-}x_{ij})M^{\infty}$ Null und somit fällt weg. Für alle anderen Kundenpaare wird dieser letzte Subtrahend mit hinreichend großem M^{∞} so groß, dass die Größer-Bedingung immer gilt und somit nicht greift.
(1.6.) gewährleistet, dass die maximale Tourdauer *dur* eingehalten wird.
(1.7.) gewährleistet, dass die Anzahl verfügbarer Fahrzeuge nicht überschritten wird.
Die in der Literatur an dieser Stelle oft verwendete Formulierung $\Sigma^j(x_{0j})=m$ ist zu scharf, falls Mehrfacheinsätze der Fahrzeuge zulässig sind, da diese Formulierung die Anzahl der Touren auf die Anzahl der verfügbaren Fahrzeuge beschränkt.
(1.8.) ist die Unbestimmtheitsaussage, die besagt, dass beim Eingehen des Auftrags i alle Aufträge j ≤ i bekannt, aber die Aufträge j > i noch unbekannt sind. Sie repräsentiert den dynamischen Aspekt des Optimierungsproblems.
(1.9.) zeigt den Freiheitsgradverlust bei der Zustellzeitzusage:
vor der Bestellung ist die mögliche Lieferzeit ein Intervall um den präferierten Lieferzeitpunkt tp_j mit Radius = Toleranz D,

nach der Bestellung ist die mögliche Lieferzeit ein Intervall um den zugesagten Lieferzeitpunkt ta_j mit Radius = Präzision d.

Aus OR-Sicht ist die Maximierung der Zielfunktion *max E(n)* schwierig, da die Bestellungen sequentiell eingehen (s. 1.8.). Es handelt sich also um ein Optimierungsproblem unter Unsicherheit in Hinblick auf Anzahl, geografische und zeitliche Lager der noch nicht eingegangenen Bestellungen. Das eigentliche Problem ist damit nicht ohne weiteres lösbar.

Stand der Forschung

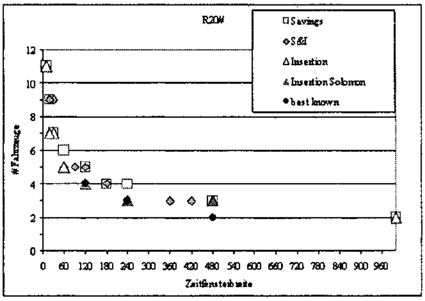
Die Literatur zur Lösung von VRSP-TW ist umfangreich. (Thangiah 1995), (Potvin u. Bengio 1996), (Berger et al. 1998) und (Bräysy 2001) untersuchen beispielsweise den Einsatz genetischer Algorithmen. (Gambardella et al 1999) befassen sich mit dem Einsatz von Ameisenalgorithmen, und (Cordeau et al 2000) mit Tabu Search Verfahren zur Lösung des VRSP-TW um nur einige wenige zu nennen. Sobald es jedoch um Unsicherheit (uncertainty) in Hinblick auf die Lager der zu erwartenden Kunden geht, wird es in der Literatur deutlich „dünner". Zur Lösung des VRSP-SCD haben sowohl (Gendreau et al 1996) als auch (Erera u. Daganzo 2000) ein auf den Anwendungsfall „Abholtransporte mit unbekannten Abholmengen von bekannten Abholpunkten" zugeschnittenes Verfahren entwickelt. Eine allgemeinere Form, bei der Kunden zu bedienen sind, deren Lage ex ante unbekannt ist, wird jeweils nicht betrachtet. Die einzige dem Autor bekannte Quelle, die einen Ansatz zur Lösung des D-VRSP-UC vorschlägt, ist (Bent u. van Hentenryck 2003)[2].

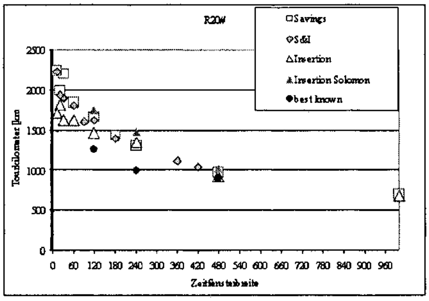
Auch in der Literatur zum Team Orienteering Problemen (TOP) werden Mehrfahrzeug-Tourenplanungsprobleme untersucht, bei denen nicht alle Kunden bedient werden müssen (Chao et al 1996). Es gibt jedoch nach derzeitiger Kenntnis des Autors noch keine entsprechenden Lösungsvorschläge zur Lösung des oben beschriebenen D-VRSP-TW-UC-DO in der Literatur zum Orienteering Problem.

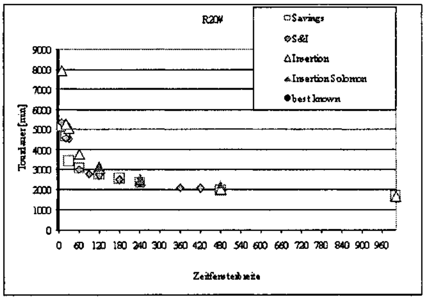
[2] Sie schlagen vor, für mehrere Szenarien Tourenpläne zu generieren, die jeweils sowohl die bereits bekannten, als auch „künftige" Aufträge enthalten. „Künftige" Auftrage werden dabei zufällig entsprechend der jeweiligen Verteilungsfunktion generiert. Das Verfahren ist in so fern „evolutiv", als mit fortschreitendem Auftragseingang immer wieder neue Szenarien generiert werden. Der aktuell „beste Plan" ist dabei jeweils der, der am meisten Freiräume für die zu erwartenden „künftigen" Aufträge lässt (least commitment strategy).

Analyse des D-VRSP-TW-UC-DO

Bevor wir uns dem D-VRSP-TW-UC-DO zuwenden, soll kurz auf den Zusammenhang zwischen der zugesagten Lieferzeitintervallbreite und den resultierenden Transportkosten eingegangen werden. Bei Simulation von VRSP-TW-Szenarien mit veränderlicher Lieferzeitintervallbreite ist erkennbar, dass die Transportkostenindikatoren (Anzahl benötigter Fahrzeuge, km, Fahr- & Wartezeiten) in der Regel mit abnehmender Lieferzeitintervallbreite steigen. Die folgenden Charts zeigen am Beispiel von Variationen der Solomon-Instanz R205 (ohne Mengenrestriktionen) diesen Effekt:







Savings
(Q: eigene Messung)

S&I: Savings mit anschließendem Einfügeverfahren
(Q: eigene Messung)

Insertion: Einfügeverfahren
(Q: eigene Messung)

Insertion Solomon
(Q: Solomon 1984; R201,R205,R211)

Best known – www. sintef.no (2004)
(Q: Homberger&Gehring 1999; R201)
(Q: Rousseau et al forthcoming; R205)
(Q: Bent & van Hentenryck 2001; R211)

Als unbewiesene Arbeitshypothese soll daher b.a.w. gelten:

Die Kostenindikatoren: Anzahl benötigter Fahrzeuge m, Gesamttourkilometer l_{ges} und Gesamttourdauer (inkl. Wartezeiten) t_{ges} wachsen mit abnehmender Lieferzeitintervallbreite (=2d) (H1)

Das Kernproblem beim D-VRSP-TW-UC-DO besteht jedoch in praxi darin, mit einer beschränkten Fahrzeugflotte möglichst viel Umsatz zu machen, ohne die jeweils zugesagten Lieferzeitintervalle zu verletzen. Die folgenden beiden Effekte haben dabei eine signifikante Auswirkung auf die Lösungsgüte des D-VRSP-TW-UC-DO.

Der Begriff der Gleichzeitigkeit

Die Anzahl der verfügbaren Fahrzeuge beschränkt die Anzahl der Lieferungen, die an verschiedenen geografischen Orten „zum selben Zeitpunkt“ zugesagt werden können. Dabei ist der Begriff „zum selben Zeitpunkt“ etwas ungenau und soll daher durch ein formales Maß zur „Gleichzeitigkeit“ ersetzt werden.

Seien die in einem beliebigen Lieferszenario zu bedienenden n Stops sortiert nach aufsteigender frühester Lieferzeit tf:

$$tf_i < tf_j \Leftrightarrow i<j \tag{2.1}$$

Sei dann das Gleichzeitigkeitsmaß G die maximale Anzahl der Belieferungen im Lieferszenario, die gleichzeitig stattfinden:

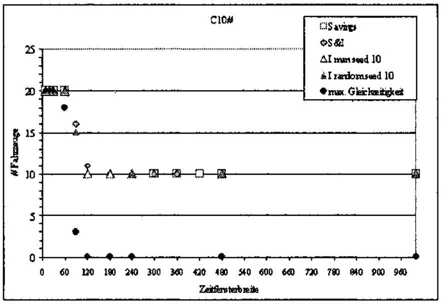
$$G = 1 + \max k \;\; \Big| \;\; tf_{j-k}+sz_{j-k}>ts_j \;\; \forall k=1..j-1, \;\; \forall j=1..n \tag{2.2}$$

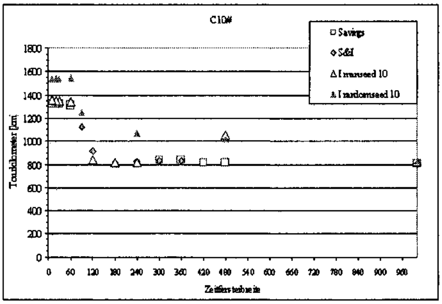
D.h. von keinem der k Vorgänger i von j in der nach tf sortierten Liste kann j ohne Lieferzeitintervallverletzung angefahren werden, selbst wenn die Fahrzeit von i nach j vernachlässigt wird. Da G definitionsgemäß eine Untergrenze der benötigten Anzahl von Fahrzeugen m für ein definiertes Lieferszenario ist, dürfen keinesfalls Lieferzeitintervalle so zugesagt werden, so dass $G>m$ ist (Vermeidung zu vieler gleichzeitiger Lieferungen).

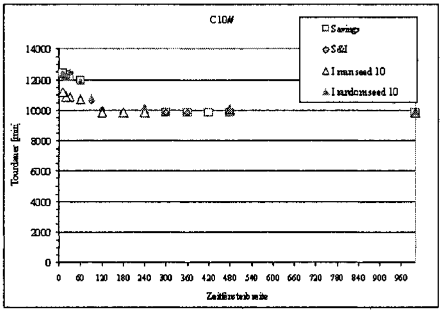
Der Begriff der „Impliziten Tourstruktur“

Die Betrachtung von Variationen C10# der Solomon-Instanz C107 zeigt einen interessanten Effekt in Hinblick auf die Abhängigkeit der Anzahl benötigter Fahrzeuge und der Transportkosten von der Lieferzeitintervallbreite. Bis zur Erreichung der „Gleichzeitigkeitsrestriktion“ (s.o.) verlaufen alle 3 Kenngrößen m, l_{ges} und t_{ges} konstant (d.h. unabhängig von der Lieferzeitintervallbreite). Dieser Effekt lässt sich leicht erklären, da Solomon die Instanz entlang einer vordefinierten „impliziten Tourstruktur“ definiert hat, von der er annimmt, sie sei optimal.

„Implizite Tourstruktur“ sei eine Eigenschaft eines Transportauftragsbestands, die einem raum-zeitlichen Tourverlauf durch den Transportauftragsbestand entspricht. Diese implizite Tourstruktur muss a priori nicht bekannt sein. (Def.1)







Savings
(Q: eigene Messung)

S&I: Savings mit anschließendem Einfügeverfahren
(Q: eigene Messung)

I: Einfügeverfahren (Insertion)
man seed (Manuelle Auswahl der Seed-Kunden für Insertion)
random seed (Zufallsauswahl der Seed-Kunden für Insertion)
(Q: eigene Messung)

Max Gleichzeitigkeit:
Anzahl der am Liefertag maximal gleichzeitig zu beliefernden Kunden
(Q: Analyse der Eingangsdaten)

Die oben postulierte Arbeitshypothese H1 wird somit wegen des Gegenbeispiels aus C10# abgeschwächt zu H2:

Die Kostenindikatoren m, l_{ges} und t_{ges} wachsen mit abnehmender Lieferzeitintervallbreite, falls der Transportauftragsbestand keine signifikante implizite Tourstruktur aufweist. (H2)

Ein neuer Ansatz zur Lösung des D-VRSP-TW-UC-DO

Da bisherige Untersuchungen darauf hin deuten, dass i.d.R. die Tourkosten verfahrensunabhängig stark mit der abnehmenden Lieferzeitfensterbreite steigen (s.o.), bei Vorliegen einer impliziten Tourstruktur diese Kosten jedoch konstant bleiben können (falls keine Gleichzeitigkeitsrestriktionen greifen), liegt es nahe, sich dieses Phänomen der impliziten Tourstruktur zur Lösung das D-VRSP-TW-UC-DO im Kontext des Lebensmittel-home-delivery zunutze zu machen.

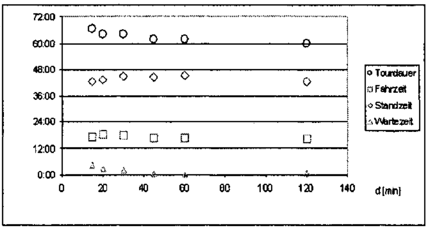
Verfahren

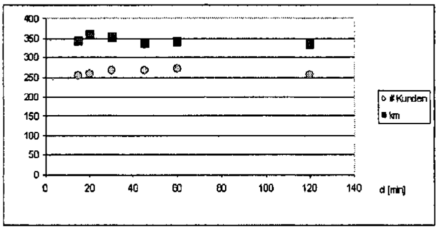
Um eine implizite Tourstruktur bei den zu bedienenden Aufträgen zu erzeugen, werden zuerst sogenannte *Aggregate* definiert. Ein Aggregat ist ein geografisches Gebiet (z.B. ein PLZ-Bereich), das als Platzhalter für die im Gebiet erwarteten Lieferaufträge fungiert. Die „Standzeit" in einem Aggregat ist die Summe der zu erwartenden Bedienzeiten plus die Summe der zu erwartenden Fahr- und Wartezeiten innerhalb des Aggregats. Die Aggregate werden zu *Tour-Templates* verbunden. Ein Tour-Template ist die raum-zeitliche Sequenz dieser Aggregate. Anhand dieser Tour-Templates kann bestimmt werden, wann voraussichtlich welches Fahrzeug welches Aggregat bedienen wird. Bei eingehenden Bestellungen für ein bisher „unbesetztes" Aggregat werden anhand der Tour-Templates Lieferzeitvorschläge gemacht, aus denen der Kunde auswählen kann. Sobald eine Lieferung bestätigt wird, wird diese physisch in eine Tour eingefügt, die ausschließlich durch die bereits zugesagten Lieferungen aufgespannt wird. Bei eingehenden Bestellungen für ein bereits „besetztes" Aggregat werden Lieferzeitvorschläge dadurch ermittelt, dass die Bestellung an bester Position in die Tour eingefügt wird. In Anlehnung an (Kunze 2003) sei dieses Verfahren hier ORA-Verfahren genannt.

Ergebnisse

Im Gegensatz zu den oben dargestellten Ergebnissen von klassischen Verfahren, die keine bekannte implizite Tourstruktur zur Tourenbildung berücksichtigen, lässt sich für das ORA-Verfahren exemplarisch zeigen, dass die Transportkostenindikatoren unabhängig von der zugesagten Service-Zeitfensterbreite sind, für den Fall, dass die Anzahl der nachgefragten Lieferungen deutlicht größer als das Transportkapazitätsangebot ist, und dass die Lieferzeitanteile deutlich größer als die durchschnittlichen Fahrzeitanteile sind.

Das folgende Beispiel zeigt den Verlauf der Performance-Indikatoren „Gesamttourdauer t [min]" (unterteilt nach Fahr-, Stand- und Wartezeit), „Anzahl bedienbarer Kunden N" und „Gesamttourkilometer x [km]". Die Ergebnisse beziehen sich auf ein Szenario mit N^{max}=500 geografisch homogen verteilten Lieferanfragen aus einem Gebiet mit 8 km Radius um den Depotstandort, einem Fuhrpark von 6 Fahrzeugen mit je 3 Touren (Vormittag, Nachmittag, Abend) und je 10 Minuten Lieferzeit/Zustellung. Entfernungen wurden über Luftlinie mit 20km/h errechnet - d.h. die Lieferzeitanteile dominieren die Fahrzeitanteile.





Als wichtigstes Ergebnis zeigt sich, dass die Anzahl der zugesagten Lieferungen *n*, die bedient werden können, **unabhängig von der Avispräzision** eine Sättigung bei ca. 262 Stops aufweist ($n^{\varnothing}$ = 262; n^{max} = 271). Die Gesamttourdauer t_{ges} ist fast konstant ($t_{ges}^{\varnothing}$=63h; t_{ges}^{max}=66h), genau wie der insgesamt zurückgelegte Weg ($l_{ges}^{\varnothing}$ = 344; l_{ges}^{max} = 361). Beide letztgenannten Werte variieren erwartungsgemäß je nach dem, welche Aufträge in welcher Reihenfolge zugesagt werden.

Das Verfahren nutzt den Einfluss der „impliziten Tourstruktur“, der auch in der Analyse der Solomoninstanzen C10# deutlich wurde, so dass sich die Kenngrößen *n*, t_{ges} und l_{ges} „unabhängig“ von der Avispräzision entwickeln können.

Kritisch ist anzumerken, dass im untersuchten Beispiel der Fahrzeitanteil relativ gering ist. Weitere Untersuchungen sind daher in Vorbereitung, um den Einfluss des Fahrzeitanteils auf die Indikatoren *n*, t_{ges} und l_{ges} näher zu analysieren.

Zusammenfassung

In der Tourenplanungspraxis hat sich das Prinzip „Avis nach Tourenplanung“ durchgesetzt, da gängige Verfahren recht gute Optimierungsergebnisse ohne enge Zeitfenster erzielen können. Im Anschluss an die Tourenoptimierung können dann die geplanten Lieferzeitpunkte sehr präzise avisiert werden. Muss eine derartige **präzise Avisierung** jedoch **vor der Tourenplanung** in Unkenntnis der anderen noch zu erwartenden Lieferaufträge erfolgen, so ist die „**Gleichzeitigkeit** von Lieferterminen“ ein wesentlicher Kostentreiber (s. Analyse von C10#), da sie im worst case die benötigte Anzahl an Fahrzeugen determinieren kann. Es ist daher darauf zu achten, dass die Anzahl gleichzeitiger Liefertermine unbedingt unter der Anzahl der verfügbaren Fahrzeuge liegt. Sind Tourenplanungsaufgaben ohne das Vorliegen einer bekannten **impliziten Tourstruktur** zu lösen, so steigen die Kostenindikatoren *d* und *t* mit der Enge der Zeitfenster. Gibt es eine „optimalitätsnahe“ implizite Tourstruktur im raum-zeitlichen Auftragsbestand, so ist nicht garantiert, dass Heuristiken diese implizite Tourstruktur finden. Das vorgestellte Verfahren nutzt die Freiheit im Dialog mit dem Kunden Lieferzeitfenster so zu definieren (**Vorschlagsverfahren**), dass Sie gut in eine implizite Tourstruktur passen. Es wurde an einem Beispiel gezeigt, dass so die **Abhängigkeit der Transportkosten von der Avispräzision durchbrochen** werden kann.

Quellen

(Bent u. van Hentenryck 2003), Bent, Russel W.; Van Hentenryck, Pascal (2003) *Scenario Based Planning for Partially Dynamic Vehicle routing with Stochastic Customers [online],* Department of Computer Science; Brown University Providence, RI 02912; www.cs.brown.edu/people/rbent/ORsr.pdf

(Bent & van Hentenryck 2001; R211) Bent, R.; Van Hentenryck, P. (2001) *A Two-Stage Hybrid Local Search for the VRP-TW,* Technical Report CS-01-06, Department of Computer Science, Brown University

(Berger et al. 1998), Berger, Jean ; Salois, Martin ; Begin, Regent (1998) *A hybrid genetic algorithm for the VRP-TW*, Proceedings of the 12th Bienneal Conference of the Canadian Society for Computational Studies of Intelligence, S.114-127, Springer Verlag, Berlin

(Bräysy 2001), Bräysy, Olli (2001), *Genetic Algorithms for the VRP-TW*, Arpakannus 1/2001 Special issue on Bioinformatics and Genetic Algorithms; S33-38

(Chao et al 1996), Chao, I-Ming, Golden, Bruce L., Wasil, Edward A. (1996) *The team orienteering problem*, European Journal of Operations Research 88, S. 464-474

(Cordeau et al 2000), Cordeau, Jean-Francois ; Laporte, Gilbert ; Mercier, Alain (2000) *Unified tabu search heuristic for VRP-TW*, Publication CRT-2003-03 ; University of Montreal, Canada

(Erera u. Daganzo 2000) Erera, Alan L.; Daganzo Carlos F. (2000) *Coordinated vehicle routing with uncertain demand*; presentation at ROUTE 2000 – International workshop on vehicle routing, Skodsborg (Denmark), August 16-19; 2000

(Gambardella et al 1999) Gambardella, Luca M.; Taillard, Eric; Agazzi, Giovanni (1999) *MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows* New Ideas in Optimization; S. 63-76, McGraw-Hill, London

(Gendreau et al 1996) Gendreau, Michel; Laporte, Gilbert; Séguin, René (1996), *A Tabu search heuristic for the vehicle routing problem with stochastic demands and customers*; Operations Research 44-3, S.469-477

(Homberger u. Gehring 1999) Homberger, J.; Gehring, H. (1999) *Two evolutionary metaheuristics for the VRP-TW*, INFOR, Vol. 37, S.297-318

(Kunze 2003) Kunze, Oliver (2003) *A new interactive approach on route planning with tight delivery time windows*, Proceedings of the 3rd International Conference on City Logistics, Madeira, Portugal (June 25th-27th, 2003)

(Potvin u. Bengio 1996) Potvin, Jean-Yves; Bangio, Samy (1996) *The VRP-TW part II: genetic search*, Journal on Computing, 8(2) S. 165-172

(Rousseau et al, forthcoming;) Rousseau, L. M. ; Gendreau, M. ; Pesant, G. (forthcoming) *Using Constraint-Based Operators to solve the VRP-TW*, Journal of Heuristics

(Solomon 1984) Solomon, M. (1984) *Vehicle Routine and Scheduling with Time Window Constraints: Models and Algorithms*, Dissertation, Department of Decision Sciences, University of Pennsylvania, Philadelphia

(Thangiah 1995) Thangiah, Sam (1995) *Vehicle routing with time windows using genetic algorithms*, Application Handbook of Genetic Algorithms: New Frontiers, Volume II, S.253-277, CRC Press, Boca Raton

Freight Flow Consolidation in Presence of Time Windows

Jörn Schönberger and Herbert Kopfer

Chair of Logistics, Department of Business Administration and Economics
University of Bremen, Wilhelm-Herbst-Straße 5, 28359 Bremen, Germany
{sberger,kopfer}@logistik.uni-bremen.de

Abstract. This contribution addresses the consideration of time windows in the optimization of multi-commodity network flows. For each node, one interval is specified in which the visitation is allowed. Applications in freight flow consolidation let this problem become interesting. An optimization model is proposed and a construction heuristic is presented. For improving the generated solutions, a genetic algorithm framework including several hill climbing procedures for local optimization, is configured.

1 Introduction

Multi-Commodity Network Flow Problems (MCBFP) are subject of a large number of scientific investigations. They are often consulted if a least cost flow of physical goods through a given transport network is searched. Typically, the costs represent the consumption of resources like time, fuel or budgets.

In logistics, the transportation of goods is only one particular step in the value creating process of a product. This step has to be synchronized with previous and subsequent processing steps. Therefore, time windows are specified for each single transport task in order to ensure a temporal coordination of the processing steps.

This article is about the optimization of time window-constrained flow of goods. In Section 2 the problem is stated in detail. Section 3 describes the used construction heuristic, Section 4 contains the description of a memetic improvement algorithm. The results of several numerical experiments are presented and discussed in Section 5.

2 Multi-Commodity Flow with Time Windows

A logistics service provider (LSP) is responsible for the reliable fulfillment of N pickup and delivery requests. Each request r_i expresses the need for the movement of a commodity i with capacity c_i. It has to be picked up at location p_i within the time window T_i^{pick} and unloaded within the time window $T_i^{delivery}$ at location q_i. Since the LSP does not own any vehicles it pays a forwarding company for the physical execution of the requests. The minimization of the execution costs for the complete portfolio is required.

Literature. MCNFPs are targeted in several contributions. Here, it is referred to the comprehensive article [1]. The most famous special case is the shortest path problem [2].

Applications of MCNFP related to the optimization of the flow of goods in a transport network are described in [3], termed Freight Optimization.

The consideration of time windows has been described only in the context of the single-commodity shortest path problem [4].

Problem Statement. The set $\mathcal{V} := \{p_1, \ldots, p_N, q_1, \ldots, q_N\}$ of all involved locations and the set $\mathcal{A} := \mathcal{V} \times \mathcal{V}$ of arcs between pairs of the involved locations lead to the graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \gamma, \tau)$ representing the network available to fulfill the transport demands. The function γ is defined on the set of arcs and assigns the travel distance γ_{ij} for processing from i to j to each arc (i, j) and the time for traversing (i, j) is τ_{ij}.

Each loading activity and each unloading activity is represented by a triple $a(o) := (o, t_o^{start}, t_o^{end})$. At location o, the corresponding loading or unloading activity takes place. It starts at time t_o^{start} and is completed at time $t_o^{end} := t_o^{start} + d_o$ where d_o refers to the dwell time associated with o. The earliest allowed starting time of the operation associated with o is denoted by t_o^{min} and the latest allowed finishing time is named by t_o^{max}.

The way of commodity i through the graph $\mathcal{G}$ originating from $o_1 := p_i$ and terminating in $o_{N_i} := q_i$ is determined in the *origin/destination-path (o/d-path)* $\mathcal{P}_i := (a(o_1), a(o_2), \ldots, a(o_{N_i}))$ *of commodity i.*

Along an o/d-path for commodity i, the operations are scheduled recursively starting from the earliest allowed execution time of the corresponding pickup-operation $a(o_1)$. The following starting times for $i = 1, \ldots, N_i - 1$ are computed by $t_{o_i+1}^{start} = \max\{t_{o_i}^{min}, t_{o_{i+1}}^{start} + \tau_{o_i, o_{i+1}}\}$ and the finishing times are calculated by $t_{o_i}^{end} := t_{o_i}^{start} + d_{o_i}$.

The *o/d-path* $\mathcal{P}_i$ *is feasible for commodity i* if it satisfies the time window conditions for its associated pickup operation $a(p_i)$ and its associated delivery operation $a(q_i)$.

A fee $F^{ij}(c)$ has to be transferred to the cooperating forward company for the movement of a commodity with a given capacity c along the arc (i, j).

Typically, F is degressive with respect to increasing capacity c, so that it is more profitable to move one large commodity with capacity αc along (i, j) than moving α commodities each with capacity c along this arc. Thus the consolidation of several commodities associated with several requests that are shipped along an arc (i, j) starting at time t into a shipment $S^{ij}(t) \subseteq \{r_1, \ldots, r_N\}$ is profitable in certain cases. It leads to a reduced amount to be paid as long as the saved amount of fees dominates the additional feeder and distribution costs.

The overall sum of fees to be paid for realizing the feasible o/d-paths $\mathcal{P}_1, \ldots, \mathcal{P}_N$ is calculated as follows. At first, all shipments $S^{ij}(t)$ occurring in $\mathcal{P}_1, \ldots, \mathcal{P}_N$ are identified. Then, the capacities $C(S^{ij}(t))$ of the shipments $S^{ij}(t)$ are computed by summing up the capacities of the included commodities. Next, the freight fees $F^{ij}(C(S^{ij}(t)))$ to be paid for the execution of $S^{ij}(t)$ are calculated. Finally, the overall sum of fees $F(\mathcal{P}_1, \ldots, \mathcal{P}_N)$ to be paid is computed by summing up the freight fees calculated for the particular shipments.

It is aimed to determine a set of o/d-paths $\mathcal{P}_1, \ldots, \mathcal{P}_N$, so that (A1): The o/d-path $\mathcal{P}_i$ of commodity i is feasible, (A2): If two or more paths are consolidated into the shipment $S^{ij}(t)$ at location i then i is associated with the pickup operation of a request r_l and the commodity l is contained in this shipment, (A3): If the shipment $S^{ij}(t)$ is resolved at location j then j is associated with the delivery operation of a request r_k and the associated commodity is contained in $S^{ij}(t)$.

The set $\mathcal{P}_1, \ldots, \mathcal{P}_N$ satisfying (A1)-(A3) is called an *o/d-path-family.* It is aimed to generate a least cost o/d-path-family.

Test Cases. The construction of artificial pickup and delivery transport requests is described in [5]. Following these proposals, instances are generated for all six proposed problem classes with tight or relaxed time windows combined with spatially scattered, semi-clustered or clustered locations.

The fee $f^{ij}(c)$ to be paid for the capacity c along the arc (i, j) is defined as $f^{ij}(c) = \gamma_{ij}$ for $c > 0$ and $f^{ij}(c) = 0$ for $c = 0$.

3 Construction Heuristic

Originally, the used construction procedure has been proposed for the construction of vehicle routes serving time window-constrained customer locations [6]. Three subsequent stages are controlled by a permutation $\sigma = (\sigma_1, \ldots, \sigma_N)$ of the commodities.

Preprocessing. The relevant part of the time axis is partitioned into m equidistant time slots $S_1, \ldots, S_m$. Each operation is sorted into the slot in which its latest allowed execution time falls. For commodity i the expression s_i^p refers to its pickup slot (slot of the pickup operation) and s_i^d to the delivery slot (slot of the delivery location).

An example with six commodities is used to support the presentation of the following construction step. The control permutation $\sigma = (2, 5, 4, 1, 6, 3)$ is applied. The customer specified time windows lead to the following slot assignments: $s_1^p = 2$, $s_1^d = 3$, $s_2^p = 3$, $s_2^d = 4$, $s_3^p = 2$, $s_3^d = 5$, $s_4^p = 2$, $s_4^d = 4$, $s_5^p = 1$, $s_5^d = 6$, $s_6^p = 3$ and $s_6^d = 6$.

Path-Construction. Initially, an exclusive path $\mathcal{P}_i = (a(p_i), a(q_i))$ is set up for each commodity. These paths are modified in the following steps with the goal of concatenating exclusive paths to more complex paths leading to larger shipments.

Two paths $\mathcal{P}_k$ and $\mathcal{P}_l$ are called *incompatible* if and only if at least one of the following two conditions is satisfied: (1) at least one operation o_k in $\mathcal{P}_k$ and one operation $o_l \neq o_k$ in $\mathcal{P}_l$ fall into the same time slot or (2) all operations in one path fall in time slots larger than the slot of the final operation of the other o/d-path.

In the case of satisfaction of (1), two different operations have to be scheduled approximately at the same time, which is assumed to lead to a time window constraint violation as soon as both are served in the same path. The validity of condition (2) implicates, that one o/d-path cannot be started after the other on is terminated, so that no physical bundling of the associated commodities is possible.

Two paths that are not incompatible are called *compatible*. Only two compatible paths can be merged with the goal to generate larger shipments.

The available paths are checked successively for pairwise compatibility with each other paths. As soon as two compatible paths are detected, they are updated dynamically by absorbing the operations from the other path.

Let $\mathcal{P}_{\sigma_k}$ be the first so far unconsidered path in the order determined by σ. It is checked successively whether $\mathcal{P}_{\sigma_j}$ is compatible with $\mathcal{P}_{\sigma_k}$. If $\mathcal{P}_{\sigma_k}$ and $\mathcal{P}_{\sigma_j}$ are compatible then $\mathcal{P}_{\sigma_k}$ is updated by inserting all operations from $\mathcal{P}_{\sigma_j}$ that fall into slots s with $s^p_{\sigma_k} < s < s^d_{\sigma_k}$ and which are not included so far in $\mathcal{P}_{\sigma_k}$. The other path $\mathcal{P}_{\sigma_l}$ is updated in the same manner.

This update-strategy is demonstrated for the example introduced above. The first path is $\mathcal{P}_2$. This path is compatible with $\mathcal{P}_5$, so that $\mathcal{P}_5$ is updated to $\mathcal{P}_5 := (a(p_5), a(p_2), a(q_2), a(q_5))$ but $\mathcal{P}_2$ remains unchanged since no operations fall in a time slot between $a(p_2)$ and $a(q_2)$. Next it is found, that $\mathcal{P}_4$, $\mathcal{P}_1$ and $\mathcal{P}_6$ are incompatible with $\mathcal{P}_2$. Finally, the compatibility of $\mathcal{P}_3$ with $\mathcal{P}_2$ is detected but again no additional operations can be inserted into $\mathcal{P}_2$. The next path to be combined with other paths is $\mathcal{P}_5$. It is incompatible with $\mathcal{P}_4$, $\mathcal{P}_1$ and $\mathcal{P}_6$ but compatible $\mathcal{P}_3$ so that $\mathcal{P}_5$ is updated to $\mathcal{P}_5 := (a(p_5), a(p_3), a(p_2), a(q_2), a(q_3), a(q_5))$ and $\mathcal{P}_3$ is updated to $\mathcal{P}_3 = (a(p_3), a(p_2), a(q_2), a(q_3))$. Next, $\mathcal{P}_4$ is compared with $\mathcal{P}_1$ (incompatible), $\mathcal{P}_6$ (compatible, updating $\mathcal{P}_4 := (a(p_4), a(p_6), a(q_4))$ and $\mathcal{P}_6 := (a(p_6), a(q_4), a(q_6)))$ and $\mathcal{P}_3$ (incompatible). The path $\mathcal{P}_1$ is incompatible with $\mathcal{P}_6$ and, finally, $\mathcal{P}_3$ and $\mathcal{P}_6$ are incompatible.

Ensuring time window feasibility. The generated o/d-paths are successively checked for time window constraint violations. In doing so, two cases are distinguished.

If path $\mathcal{P}_{\sigma_k}$ has been checked and if no violations have been detected, then the path $\mathcal{P}_{\sigma_k}$ is confirmed. This means, the calculated arrival and departure times at the associated locations visited in $\mathcal{P}_{\sigma_k}$ are propagated into the so far unconsidered paths $\mathcal{P}_{\sigma_{k+1}}, \ldots, \mathcal{P}_{\sigma_N}$ and can only be modified as long as they fall into the associated time windows.

If a time window constraint violation is detected in path $\mathcal{P}_{\sigma_k}$ at a certain location, then the associated pickup operation $a(o_1)$ and the corresponding delivery operation $a(o_2)$ of this commodity are deleted in the paths $\mathcal{P}_{\sigma_k}, \ldots, \mathcal{P}_{\sigma_N}$ and the o/d-path of this commodity is determined as the exclusive o/d-path $(a(o_1), a(o_2))$. This o/d-path is confirmed immediately.

4 Memetic Algorithm Path-Improvement

The memetic algorithm (MA) framework introduced in [7] is adapted for evolving o/d-path-families towards the least costs-goal. An initial population consisting of K different o/d-path-families is generated by calling the construction heuristic, described in Section 3, K times. After each call, the control permutation σ is re-determined randomly. Each existing population is replaced according to the $\mu + \lambda$-replacement strategy [8] by recombining new sequences of intermediate stops between the pickup and delivery location of each commodity. Mutation inserts a so far unconsidered operation into randomly selected offspring paths.

The offspring o/d-path-families generated by mutation and crossover do not comply with the conditions (A2) and (A3) in every case. Generally, it exists a commodity i, so that the generated offspring path $\mathcal{P}_i$ contains a pickup operation (delivery operation) of another commodity k and there is no shipment in with both i and k are carried away from (brought to) the associated location.

As a remedy, the o/d-paths in an o/d-path-family are modified and detected deficiencies of the kind mentioned just above are corrected in a straightforward way. Therefore, the o/d-paths in an offspring are checked successively according to the order σ. After potential violations of (A2) or (A3) are corrected, the path is confirmed and cannot be modified anymore.

Let $\mathcal{P}_k = (a(o_1^k), \ldots, a(o_{N_k}^k))$ be an o/d-path. For two included operations $a(o_i^k)$ and $a(o_j^k)$ with $(i \leq j)$ the $a(o_i^k)$-$a(o_j^k)$-*subpath* of $\mathcal{P}_k$ is defined as $(a(o_i^k), a(o_{i+1}^k), \ldots, a(o_{j-1}^k), a(o_j^k)$. An o/d-path $\mathcal{S}$ is called a subpath of another o/d-path $\mathcal{P}$ if there exists operations $a(o_1)$ and $a(o_2)$ in $\mathcal{P}$ so that $\mathcal{S}$ equals the $a(o_1)$-$a(o_2)$-subpath of $\mathcal{P}$.

Let $a(o_j^k)$ be an operation in $\mathcal{P}_k$. The $a(o_j^k)$-*tail of* $\mathcal{P}_k$ is defined as the subpath $(a(o_j^k), \ldots, a(o_{N_k}^k))$. Additionally, the $a(o_j^k)$-*beginning of* $\mathcal{P}_k$ is given by $(a(o_1^k), \ldots, a(o_j^k))$. Two paths $\mathcal{P}_j$ and P_k are called *consistent* if and only if at least one of the following conditions is satisfied: (i) $\mathcal{P}_j$ and $\mathcal{P}_k$ do not have any operations in common, (ii) $\mathcal{P}_j$ is a subpath of $\mathcal{P}_k$ or vice-versa, (iii) the $a(o_1^j)$-tail of $\mathcal{P}_k$ equals the $a(o_{N_k}^k)$-beginning of $\mathcal{P}_j$ or (iv) the $a(o_1^k)$-tail of $\mathcal{P}_j$ equals the $a(o_{N_j}^j)$-beginning of $\mathcal{P}_k$

In all other cases $\mathcal{P}_j$ and P_k are called *inconsistent*. An o/d-path-family in which all included o/d-paths are pairwise consistent satisfy the conditions (A2) and (A3).

The following scheme is used to check the pairwise consistency, to correct inconsistencies and to propagate the decisions into so far unconfirmed o/d-paths. Initially, all o/d-paths in the considered o/d-path-family are unconfirmed. Now let $\mathcal{P}_k$ be the first so far unconfirmed o/d-path according to the order induced by σ.

First phase (consistency check): It is checked whether $\mathcal{P}_k$ is pairwise consistent with all so far confirmed o/d-paths. If $\mathcal{P}_k$ and a confirmed o/d-path $\mathcal{P}_j$

are inconsistent then all error-causing operations are successively removed from $\mathcal{P}_k$ together with the associated pickup or delivery operation (if included). Finally, the modified $\mathcal{P}_k$ is confirmed. For all commodities i associated with a removed operation, the o/d-path $\mathcal{P}_i := (a(p_i), a(q_i))$ is generated and confirmed.

Second phase (subpath propagation): All subpaths of $\mathcal{P}_k$ are propagated into the paths of the commodities associated with so far unconfirmed o/d-paths. Consider successively all operations $a(o)$ in $\mathcal{P}_k$. Let $c(a(o))$ be the commodity related to operation $a(o)$. If $\mathcal{P}_{c(a(o))}$ is unconfirmed so far then one of the following three options is executed.

If $\mathcal{P}_k$ contains both operations belonging to $c(a(o))$ then $\mathcal{P}_{c(a(o))}$ is set to the $a(p_{c(a(o))})$-$a(q_{c(a(o))})$-subpath of $\mathcal{P}_k$. If $\mathcal{P}_k$ contains only the pickup-operation $a(p_{c(a(o))})$ then the associated delivery operation $a(q_{c(a(o))})$ is appended to the $a(p_{c(a(o))})$-tail of $\mathcal{P}_k$ and this extended tail becomes the new o/d-path for commodity $c(a(o))$. If $\mathcal{P}_k$ contains only the delivery-operation $a(q_{c(a(o))})$ then the associated pickup operation $a(p_{c(a(o))})$ is prefixed to the $a(q_{c(a(o))})$-beginning of $\mathcal{P}_k$ and this extended beginning becomes the new o/d-path for commodity $c(a(o))$.

The mode of operation of this procedure is demonstrated in the following example. Parameterized with the sequence $\sigma = (2, 5, 4, 1, 6, 3)$, the procedure modifies the erroneous o/d-path-family $\mathcal{P}_1 = (a(p_1), a(q_1), \mathcal{P}_2 = (a(p_2), a(q_2)), \mathcal{P}_3 = (a(p_3), a(p_2), a(q_2), a(q_3)), \mathcal{P}_4 = (a(p_4), a(p_6), a(q_4)), \mathcal{P}_5 = (a(p_5), a(p_2), a(p_3), a(q_2), a(q_4), a(q_3), a(p_1), a(q_5))$ and $\mathcal{P}_6 = (a(p_6), a(q_4), a(p_3), a(q_6))$ into the o/d-path-family $\mathcal{P}_1 = (a(p_1), a(q_5), a(q_1), \mathcal{P}_2 = (a(p_2), a(q_2)), \mathcal{P}_3 = (a(p_3), a(q_3)), \mathcal{P}_4 = (a(p_4), a(p_5), a(p_2), a(q_2), a(q_4)), \mathcal{P}_5 = (a(p_5), a(p_2), a(q_2), a(q_4), a(p_1), a(q_5))$ and $\mathcal{P}_6 = (a(p_6), a(q_4), a(q_6))$ which is free of errors (see also Fig. 1).

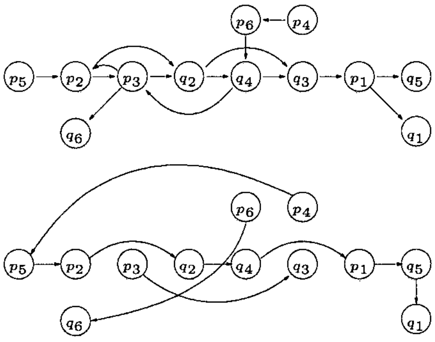


Fig. 1. Hill climbing repair of (A2) and (A3) constraint violations. In the figure above, q_2 has two predecessors and p_2 has two successors. The hill climbing procedure is applied to this family, which is shown in the lower figure. It is free of constraint violations.

5 Computational Experiments

The proposed MA has been implemented and assessed for its suitability and is applied in five independent runs to the instances of the six problem classes. **Algorithm Configuration.** A population consisting of K=200 individuals is evolved through 150 iterations so that overall 30000 o/d-path-families are tested. The crossover frequency is set to 100% and the mutation frequency is 90%. These intensities are very high but necessary, since the application of the hill climbers refuses a large part of the proposed modifications. The number of time slots is set to $m = 5N$.

Experiments. For each of the six problem classes three indicators are derived from the observed results. At first, the percentages ϵ of exclusively fulfilled requests, which are not contained in any bundle with at least one other request commodity, are calculated. Secondly, the improvement over the evolution stages is subsumed. Therefore, the improvement ι relative to the best observed fitness in the initial population is calculated. At last, the average deviation δ from the costs of the solution in which all requests are served by own vehicles (the VRPTW-case) is calculated in order to answer the question for which type of requests the freight consolidation-algorithm is best suited.

Presentation and Discussion of Results. The achieved values are presented in Tab. 1. It is observed that tight time windows (R1, RC1, C1) prevent the consolidation of a large fraction of requests. In almost half of the requests, the associated commodities cannot be consolidated with others. In case of relaxed time windows this quote ϵ varies between 12.2% and 26%.

The memetic search is able to find significantly better o/d-path-families compared to the initially generated families.

	R1	RC1	C1	R2	RC2	C2
ϵ	54.7%	58.5%	48.1%	12.2%	26.0%	26.0%
ι	30.4%	30.6%	62.0%	47.5%	46.6%	54.6%
δ	32.7%	39.6%	51.0%	-30.7%	4.8%	-60.6%

Table 1. Computational results achieved for the artificial test instances

The value of ι varies between 30.4% and 62.0% (with means the initially observed costs can be reduced by 30.4% up to 62.0%). Finally, it can be stated, that the overall algorithm performance is sufficient for the test cases with tight instances. Here significant improvements between 32.7% and 51.0% are achieved compared to the VRPTW-case. However, the performance for the problems with relaxed time windows is disappointing. In the R2 and in the C2-instances, the VRPTW-values are not reached. Since the observed improvement is sufficient (with respect to the ι-values), the families produced by the construction heuristic are of insufficient quality. It might be that the time window oriented consolidation is not effective enough. Additionally, the scope of the problem should be extended and for each arcs, several different fee functions (representing alternative LSPs) should be provided.

6 Conclusions and Outlook

The extension of MCNFP by time windows for visiting nodes in the underlying network has been addressed in this article. An adequate optimization model has been proposed. Several specialized algorithms are composed to an MA that has proven its general applicability to solve artificial test instances. However, the proposed construction heuristic seems to be well suited only for instances with tight time windows, so that other path construction approaches should be assessed.

References

1. McBride R.D. (1998) Advances in Solving the Multicommodity-Flow Problem. Interfaces **28** **(2)**, 32–41
2. Williams H.P. (1999) Model building in mathematical programming, 4th edition, Wiley
3. Kopfer H. (1992) Konzepte genetischer Algorithmen und ihre Anwendung auf das Frachtoptimierungsproblem im gewerblichen Güterfernverkehr. OR Spektrum **14**, 137–147
4. Desrochers M., Soumis F. (2001) A reoptimization algorithm for the shortest path problem with time windows. European Journal of Operational Research **35**, 242–254
5. Nanry W.B., Barnes J.W. (2000) Solving the pickup and delivery problem with time windows using reactive tabu search. Transportation Research Part B **34**, 107–121
6. Schönberger J., Kopfer H., Mattfeld, D.C. (2002) A Combined Approach to Solve the Pickup And Delivery Selection Problem in Leopold-Wildburger U., Rendl F., Wäscher G. (eds.) Operations Research Proceedings 2002
7. Schönberger J. (2004) Operational Freight Carrier Planning - Investigations on Basic Concepts, Optimization Models and Advanced Memetic Algorithms. PhD-Thesis, so far unpublished
8. Bäck T., Fogel D.B., Michalewicz Z. (2000) Evolutionary Computation 1 - Basic Algorithms and Operators. IoP Publishing

Minimizing Total Delay in Fixed-Time Controlled Traffic Networks

Ekkehard Köhler, Rolf H. Möhring, and Gregor Wünsch

Technische Universität Berlin, Institut für Mathematik, MA 6-1, Straße des 17. Juni 136, D-10623 Berlin, Germany
{ekoehler,moehring,wuensch}@math.tu-berlin.de

Abstract. We present two different approaches to minimize total delay in signalized fixed-time controlled inner city traffic networks. Firstly, we develop a time discrete model where all calculations are done pathwise and vehicles move on "time trajectories" on their routes. Secondly, an idea by GARTNER, LITTLE, and GABBAY (GLG) is extended to a continuous, linkwise operating model using "Link Performance Functions" to determine delays. Both models are formulated as mixed-integer linear programs and are compared and evaluated by PTV AG's simulation tool VISSIM 3.70.

1 Introduction

Controlling the inner city traffic by a "good" setting of relevant traffic light parameters is an appropriate way of reducing congestions in networks and delays in general, respectively. We consider a scenario of dense but almost steady traffic, which appears for example in periods of morning peaks of rush hour traffic. In such cases often fixed-time controlling is used since vehicle actuated or adaptive controlling cannot accentuate their advantages, which lie in the ability of adjusting to different traffic situations. Before we start describing the two optimization models, we review some notation used in traffic engineering.

First of all, we consider networks where there is a *light-signal system* at each intersection, including various single *traffic lights* some of which are combined into so-called *signal groups*. Each of these signal groups controls traffic throughput of a different direction, for example inbound, outbound, or turning traffic. For each signal group there is a *signal timing plan* determining the beginning and ending of the green- and red phase, the so called *red-green split*. After a predetermined amount of time, the *cycle time*, patterns of red and green recur. The most important parameter is the so called *offset*, which describes how light-signal systems of different intersections, or their signal timing plans respectively, are set relative to a given zero-point (Sec. 2) or are set relative to each other (Sec. 3). For both approaches cycle time, red-green split, and the vehicles' *travel time* on the links are fixed and the offset acts as decision variable.

2 A discrete path-based approach

Since in this first approach all delay analysis is done pathwise, we refer to the model as *path-based model* and restrict ourselves in the following to a single path. The model's main characteristic is its discrete structure, see Fig. 1. The cycle time is divided into T trajectories. For example, let $T = 40$; then at the beginning of each route each trajectory carries 0.555 cars per unit of time in case of an assumed traffic volume of 1000 vehicles per hour and a cycle time of 80 seconds. Below, we identify the cycle time with the parameter T and establish a partition into one trajectory per second of cycle time, which will be $T = 80$ for all quoted examples.

For a further development of the model it is necessary to introduce the following parameters: the set of all paths in the network is denoted by $\mathcal{P}$ and the edge set[1] of a path $P \in \mathcal{P}$ is given by $E(P) = \{e_1^P, \cdots, e_{\alpha_P}^P\}$ where α_P is the number of edges of P. The set of intersections, i.e. the set of different traffic-signal systems, is denoted by $\mathcal{K}$, whereas $\mathcal{R}(K)$ is the set of the signal groups corresponding to intersection K. Canonically, the parameter $\tau(e)$ stands for the integral travel time needed to traverse edge $e \in E(P)$ of path $P \in \mathcal{P}$. We will formulate the problem as a MIP and use the following variables: first, we have binary offset variables h, which are defined for each intersection and each point in time $\{1, \ldots, T\}$. For a $K \in \mathcal{K}$ only one variable $h[K, t]$ equals 1. The t_0 with $h[K, t_0] = 1$ corresponds to the offset value at intersection K. A second group of binary variables indicates whether a certain point of time $t \in \{1, \ldots, T\}$ at intersection $K \in \mathcal{K}$ and signal group $R \in \mathcal{R}(K)$ belongs to a green phase or not. These variables are denoted by

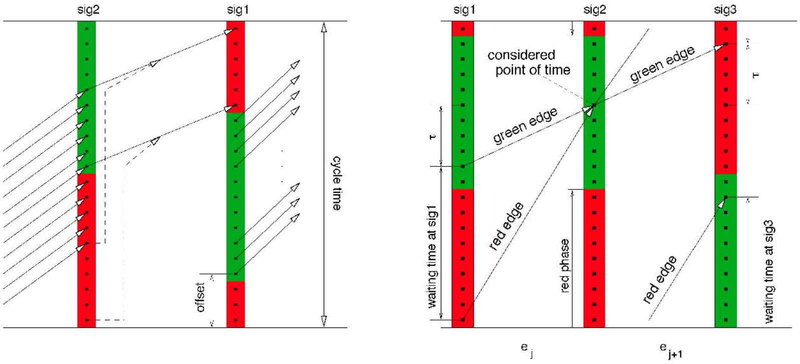


Fig. 1. Consecutive traffic signals of a particular path. At the beginning of each path each of the trajectories leads the same amount of vehicles.

[1] We refer to links as edges although we do not actually use an underlying graph structure.

z. So, $z[K, R, t] = 1$ indicates a red phase and $z[K, R, t] = 0$ a green phase, respectively. The variables h and z are connected via

$$\sum_{i \in l[K,R,t]} h[K, i] = z[K, R, t] \qquad \forall K \in \mathcal{K}, \forall R \in \mathcal{R}(K), \forall t \in T, \qquad (1)$$

where $l[K, R, t]$ is a preprocessed list of numbers in $\{1, \ldots, T\}$ such that t belongs to a red phase at intersection K and signal group R iff the offset at K is an element of $l[K, R, t]$.

As mentioned above, vehicles move on trajectories on their routes. Each trajectory leads to a certain point of time at an intersection and depending on the traffic light cars on that trajectories can pass the intersection or have to wait because of a red signal. For both cases (a red or a green signal) outgoing trajectories are provided, always one of which is blocked by a constraint and vehicles have to follow "their right" trajectory, as a result. See Fig. 1. The variables x_r and x_g measure the amount of vehicles following those *red edges* and *green edges*. Such blocking is realized by

$$x_g[P, e, t] \leq M \cdot (1 - z[K_{(P,e)}, R_{(P,e)}, t]) \qquad \forall P \in \mathcal{P}, \forall e \in E(P) \backslash \{e_1^P\}, \forall t \in T \qquad (2)$$

and

$$x_r[P, e, t] \leq M \cdot z[K_{(P,e)}, R_{(P,e)}, t] \qquad \forall P \in \mathcal{P}, \forall e \in E(P) \backslash \{e_1^P\}, \forall t \in T, \ (3)$$

where M denotes a big constant and $K_{(P,e)}, R_{(P,e)}$ the obvious dependencies. Of course, the approach of traffic flow on trajectories only works, if the cars are "forced" to stay on the trajectories and do not get lost. This is ensured by the following flow conservation equation

$$\begin{aligned} x_g[P, e+1, t] + x_r[P, e+1, t] &= x_g[P, e, t - \tau(e)] \qquad (4) \\ &+ x_r[P, e, t - \tau(e) - \mathbf{WT(P, e)}] \quad \forall P, t, \forall e \in E(P) \backslash \{e_1^P, e_{\alpha P}^P\}. \end{aligned}$$

The waiting time, which has to be constant and integral, is denoted by the parameter WT and calculated with WEBSTER's classical formula [3].

2.1 The MIP formulation

Together with an equation ensuring that only one offset is adjusted and an equation corresponding to an initial flow conservation, constraints (1)-(4) now form our mixed-integer linear program. This MIP models our optimization problem well, however, it has to be mentioned that the model's size, namely the large amount of integral variables, leads to large computation times. For example, a computation time of more than 9 hours[2] is needed for a 9-intersection test network and a discretization of the cycle time into 80 units.

[2] on a 1.7GHz Linux computer with 512MB memory

Note that for the objective, only vehicles on red edges must be taken into account.

$$minimize \sum_{P\in\mathcal{P}} \sum_{j=2}^{\alpha_P} \sum_{t\in T} \quad \mathbf{WZ}[K_{(P,e_{j-1}^P)}, R_{(P,e_{j-1}^P)}] \cdot x_r[P, e_j^P, t]$$

$$\sum_{t\in T} h[K,t] = 1 \qquad \forall K \in \mathcal{K}, \tag{5}$$

$$\sum_{i\in l[K,R,t]} h[K,i] = z[K,R,t] \tag{6}$$

$$\forall K \in \mathcal{K}, \forall R \in \mathcal{R}(K), \forall t \in T,$$

$$x_g[P,e,t] \leq M \cdot (1 - z[K_{(P,e)}, R_{(P,e)}, t]) \tag{7}$$

$$\forall P \in \mathcal{P}, \forall e \in E(P)\backslash\{e_1^P\}, \forall t \in T,$$

$$x_r[P,e,t] \leq M \cdot z[K_{(P,e)}, R_{(P,e)}, t] \tag{8}$$

$$\forall P \in \mathcal{P}, \forall e \in E(P)\backslash\{e_1^P\}, \forall t \in T,$$

$$x_g[P, e_2^P, t] + x_r[P, e_2^P, t] = flowrate[P] \tag{9}$$

$$\forall P \in \mathcal{P}, \forall t \in T,$$

$$x_g[P, e_{j+1}^P, t] + x_r[P, e_{j+1}^P, t] =$$

$$x_g[P, e_j^P, t - \tau(e_j^P)] + x_r[P, e_j^P, t - \tau(e_j^P) - \mathbf{WT}(P, e_j^P)] \tag{10}$$

$$\forall P \in \mathcal{P}, \forall t \in T, j = 2 \cdots \alpha_P,$$

$$h, z \quad \in \{0,1\},$$

$$x_g, x_r \quad \geq 0.$$

3 A continuous model

This second model is based on an approach by GARTNER, LITTLE, and GABBAY (GLG)[1]. In 1975 they formulated a "Network Coordination Problem" (NCP) where cycle time and red-green split are fixed and total waiting time is to be minimized by finding optimal offsets. Later, they included cycle time and red-green split to the set of variables and formulated a "Network Synchronization Problem" [2]. We extend their NCP model in a canonical way and introduce some adjustments in the objective function.

Again, some parameters have to be introduced first: The cycle time is again denoted by T (in our examples it is set to 80 seconds). The traffic network is represented by a directed multi-graph $G = (V, A)$, where the nodes $v \in V$ are the intersections and the edges $e \in E$ stand for the traffic carrying streets/links between intersections. Thus one can see that the graph has to be directed, since traffic may move in both directions of a link. Moreover there maybe more than one copy of an arc because we have to distinguish between different vehicle flows of a link according to which signal group they come from and go to. The set of all cycles in G will be denoted by $\mathcal{C}$ and for a single $\ell \in \mathcal{C}$ the set of forward edges is $F(l)$ and the set of reverse edges is $R(l)$, whereas a cycle is traversed clockwise.

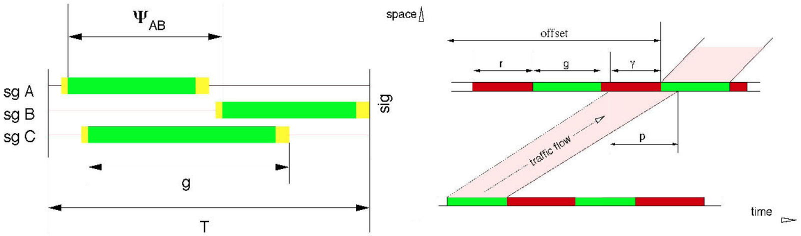


Fig. 2. The left side shows a sketch of a signal timing plan with three signal groups. The parameter Ψ describes the adjustment of the beginning of the green phases of the different signal groups. Of course, it holds that $\Psi_{AA} = 0$ and $\Psi_{AB} = 1 - \Psi_{BA}$. The right side illustrates the traffic flow in a platoon at one link. Because this structure allows linkwise definitions of e.g. the offset, all calculations and waiting time analysis is done linkwise. Note that the platoon length p does not necessarily equal the length of its green phase.

As already mentioned, there are two main characteristics of the model. Firstly, it is assumed that vehicles move in platoons with a constant rate of flow within. Secondly, this approach does not use any discretization but the parameters and variables can take continuous values. The arrival time of the platoon is denoted by γ and has to be within the interval $[-r, g]$. So, the offset ϕ is defined as the distance (in time) between the platoon's starting and the beginning of the green phase that defines the relevant interval for the platoon's γ. For example, in Fig. 2, γ is negative. Because of the linear dependency $\phi = \tau - \gamma$ one can use the arrival time γ as well as ϕ for calculating the delays.

Now, all necessary parameters and variables have been introduced. The big advantage of this "link-based" approach is its compactness. Only one group of constraints has to be formulated, namely the so called cycle-ėquations

$$\sum_{e \in F(\ell)} \phi_e - \sum_{e \in R(\ell)} \phi_e + \sum_{r=1}^{k_\ell} \Psi^{v_r^\ell}_{P[r,e^\ell_{r-1}],P[r,e^\ell_r]} = n_\ell T \qquad \forall \ell \in \mathcal{C}.$$

The parameter $P[\cdot,\cdot]$ indicates the particular signal groups. These equations are essential, since all calculations shall be done modulo the cycle time, because all cycles shall be regarded as equal[3]. Therefore a particular point of time during one circulation of the cycle has to correspond to the *same* point of time in *all* cycles. For this reason, the sum of all offsets around a cycle[4] in the graph must be an integral multiple of T.

[3] Therefore traffic load values are fixed over all cycles.

[4] A cycle may also consist of only a pair of edges!

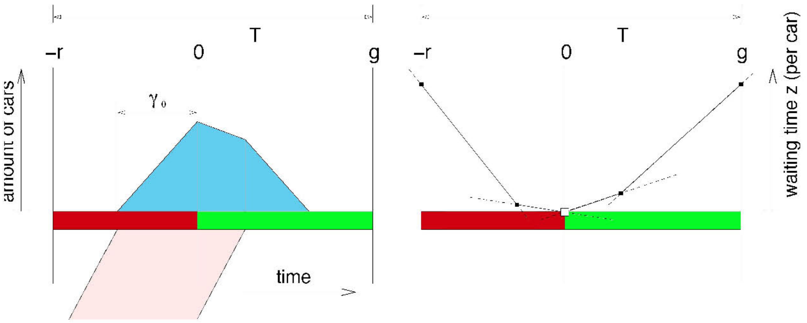


Fig. 3. A sketch of the arising waiting queues of a platoon arriving at approx. $\gamma = -r/2$. Note the break in the curve due to different service rates, first $s - q$ and after the platoon's tail s. The curve on the right represents the Link Performance Function of platoon's arrival time γ.

Now, the only open question is how to calculate delays. This is done using the arrival time γ. This is, however, quadratically related to the waiting time. To obtain a linear model each link's delay function is evaluated at some intermediate points and then approximated by a piecewise linear function, see Fig. 3. The choice of such intermediate points determines the shape of the link's delay function. At this point, we significantly simplified the GARTNER, LITTLE, and GABBAY method by ensuring the convexity of the curve. This is important for the following reason. Suppose the curve is not convex, the inequalities $z \geq g_{1,2,\ldots}(\gamma)$ where g denotes the approximation lines, do not force waiting time z to be equal to an actual waiting time. But instead the variable z probably takes too high values.

3.1 The MIP formulation

Also this model is represented as a mixed-integer linear program.

$$minimize \sum_{(i,j)\in A} f_{ij} z_{ij}$$

$$\begin{aligned}
\sum_{e\in F(\ell)} \phi_e - \sum_{e\in R(\ell)} \phi_e + \sum_{r=1}^{k_\ell} \Psi^{v_r^\ell}_{P[r,e^\ell_{r-1}],P[r,e^\ell_r]} &= n_\ell T && \forall \ell \in \mathcal{C}, \\
z_{ij} &\geq g^e_r(\gamma_{ij}) && \forall e = (i,j) \in A, \\
& && r = 1 \ldots k_e, \\
\underline{n_\ell} \leq n_\ell &\leq \overline{n_\ell} && \forall \ell \in \mathcal{C}, \\
n_\ell &\in && \forall \ell \in \mathcal{C}, \\
\phi_{ij} &\in [\tau_{ij} - g_{ij}, \tau_{ij} + r_{ij}], && \\
& && \forall (i,j) \in A.
\end{aligned}$$

One can see immediately the compact formulation compared to the first model. This also leads to faster computation times. E.g. a test network of 25 intersections could be optimally solved in about 5 seconds.

In addition to improvements within the links' delay functions we also extended the GLG approach such that we distinguish traffic flow on one edge not only according to its direction but also according to its origin's and destination's signal group. To this end, we simply allocate multiple link copies and analyze them separately. This only needs a few more cycle equalities and therefore some more integral variables, but it admits to consider a more complex traffic flow. Note that adding good lower and upper bounds on the integral variable n may lead to better computation times.

4 Simulation using VISSIM 3.70

Both optimization models have been tested and validated with PTV's simulation tool VISSIM 3.70. It provides microscopic traffic flow simulation and various analysis tools. One of them, the so called *lost time* corresponds best to our criteria *waiting time.*

In addition to test runs on simple instances, such as arterials or 2 × 2 grid networks, we constructed a 3 × 3 as well as a 4 × 4 grid network and defined 14 respectively 19 routes on which motorized individual traffic moves. In detail we have 12(16) routes of traffic following the grid on straight paths without any turns, divided into 6(8) "horizontal paths" and 6(8) vertical traffic streams. In addition, we added 2 non-conventional paths to the 3 × 3 grid that include right-turn traffic and 3 paths to the 4 × 4 grid containing left-turn traffic.

Results of simulation runs for both of these instances are listed in the tables below. They are compared to the results of a heuristic that sets a progressive signal system to some of the paths. In this way, lost time is minimized for 43% of all cars in the 3 × 3 grid and for 46% of the vehicles in the larger grid, respectively.

Table 1. Comparison between lost time measurements for the 3 × 3 grid's traffic with VISSIM 3.70, using the offsets of both approaches and a heuristic. The discrete model is denoted by A, whereas B stands for the continuous approach.

Model	Solution	Lost time in sec/veh	Rate of improvement in %
A	non opt.	21.2	29.3%
B	opt.	23.4	22%
heuristic		30.0	-
	random	32.5	-8.3%

As it can be seen in the tables only suboptimal offsets could be used for evaluating the path-based model, since its size did not permit to calculate an optimal solution within a reasonable amount of time, which was a few hours.

Table 2. Same measurements as listed in Table 1 but for the 4 × 4 grid instance.

Model	Solution	Lost time in sec/veh	Rate of improvement in %
A	non opt.	42.6	1.8%
B	opt.	39.7	8.5%
B^{*}[a]	opt.	38.8	10.6%
heuristic		43.4	-
	random	55.9	-28.8%

[a] Model with small changes at the LPFs, namely different interpolation points.

Table 2 also contains average simulation results for 11 randomly chosen offset sets. Although a more realistic comparison to results of, say, actual offsets of an actual traffic network are missing, it can be seen that both models calculate better offsets than the heuristic. We consider this to be a promising start and we will extend and advance both models as well as design further non-grid real world instances and compare results of our models with results of the actual adjusted offset set.

Acknowledgement

We would like to thank PTV AG, Karlsruhe, and Klaus Nökel for providing the simulation tool VISSIM 3.70 and for inspiring discussions.

References

1. Gartner, N. H., Little, J. D. C., Gabbay, H. (1975) Optimization of Traffic Signal Settings by Mixed-Integer Linear Programming, Part I: The Network Coordination Problem. Transportation Science **9**, 321–343
2. Gartner, N. H., Little, J. D. C., Gabbay, H. (1975) Optimization of Traffic Signal Settings by Mixed-Integer Linear Programming, Part II: The Network Synchronization Problem. Transportation Science **9**, 344–363
3. Webster, F. V. (1958) Traffic Signal Settings. Road Research Technical Paper **39**

On Asymptotic Optimality of Permutation Schedules in Stochastic Flow Shops and Assembly Lines

Roman Koryakin

Sovolev Institute of Mathematics of SB RAS, Acad. Koptyug pr-t 4,
630090 Novosibirsk, Russian Federation
romank@mail.nsk.ru

Abstract. For the stochastic flow shop problem with m machines, n jobs and operation processing times being independent identically distributed random variables, we consider permutation schedules — those in which each machine processes the jobs in the same order. For fixed m and increasing n, the asymptotic behavior of an arbitrary permutation schedule length is researched. Let T be the makespan of that schedule. Considering the maximum machine load L as a lower bound of the makespan, we show that for a wide class of operation processing time distributions the average-case ratio T/L converges to 1.

Thus, an arbitrary permutation flow shop schedule is asymptotically optimal with $n \to \infty$, and one can construct such schedule in a linear time. The same result is derived for the stochastic assembly line problem with m machines and n jobs.

1 Introduction and Definitions

Flow shop and assembly line are classical machine scheduling problems. In general case, both problems are $\mathcal{NP}$-hard, however several polynomial time algorithms are constructed in [3] (assembly lines) and [4], [5] (flow shops) producing solutions close to the optimum with absolute guarantee. It is interesting that in all algorithms the solutions were chosen within the class of permutation schedules. Thus, the idea to consider an arbitrary permutation schedule arose.

In the paper, we consider stochastic assembly line and flow shop problems and show that for an arbitrary permutation schedule the average case ratio converges to 1 asymptotically.

In general, each of the two scheduling problems considered below can be formulated as follows. Jobs $J_1, \ldots, J_n$ are to be processed on machines $M_1, \ldots, M_n$. Each job J_j consists of m operations $o_{1j}, \ldots, o_{mj}$. Operation o_{ij} is processed on machine M_i and requires time $p_{ij} \geq 0$ for its processing. Operation processing must satisfy the following conditions:

(B_1) *no simultaneous processing of any two operations on one machine is allowed;*

(B_2) *no operation preemption is allowed.*

Let s_{ij} denote the starting time of operation o_{ij} in a given schedule. The problem is to derive a schedule $S = \{s_{ij} \,|\, j = 1, \ldots, n; \quad i = 1, \ldots, m\}$ satisfying $(B_1) - (B_2)$ (and some additional requirements specific for each problem) and minimizing the maximum operation completion time (the makespan):

$$T(S) = \max_{j,i}(s_{ij} + p_{ij}). \tag{1}$$

Let p be the maximum operation length, L_i be the load of machine M_i and L be the maximum machine load:

$$p = \max_{i,j} p_{ij}, \quad L_i = \sum_{j=1}^{n} p_{ij}; \quad L = \max_{i} L_i. \tag{2}$$

Obviously, L is the lower bound on the length of any schedule S satisfying $(B_1) - (B_2)$. Precise formulations of the considered machine scheduling problems are given below.

In the Flow Shop Problem, there are two additional conditions on operation processing. Firstly,

(B_3) *no simultaneous processing of any two operations of one job is allowed*;

and secondly, each job must be processed on machines $M_1, \ldots, M_m$ consequently:

(B_4) $s_{ij} \geq s_{i-1,j} + p_{i-1,j} \quad (i = 2, \ldots, m; \; j = 1, \ldots, n)$.

Flow Shop Problem. *Derive a schedule satisfying* $(B_1) - (B_4)$ *and minimizing functional (1).*

In the assembly line problem, the last operation of each job must be processed on machine M_m:

(B_5) $s_{mj} \geq s_{ij} + p_{ij} \quad (i = 1, \ldots, m - 1; \; j = 1, \ldots, n)$.

Assembly Line Problem. *Derive a schedule satisfying* $(B_1), (B_2), (B_5)$ *and minimizing functional (1).*

2 Stochastic Machine Scheduling Problems

Formulating a stochastic machine scheduling problem, we assume all operation processing times p_{ij} being random variables with non-negative real values.

To solve a stochastic machine scheduling problem means to construct an algorithm producing a schedule S, the length $T(S)$ of which almost always (with increasing number of jobs n) satisfies the following relation for any $\varepsilon > 0$:

$$\lim_{n \to \infty} \mathbb{P}\left(\frac{T(S) - OPT}{OPT} < \varepsilon\right) = 1. \tag{3}$$

Schedule S is *asymptotically optimal* if it satisfies (3).

3 Results

Let $\pi = \{\pi_1, \ldots, \pi_n\}$ be a permutation of indices $\{1, \ldots, n\}$. A schedule S_π is called a *permutation schedule* if the jobs are scheduled on each machine in the same order π.

Let operation processing time distributions satisfy the following condition.

(P) *Random variables* $\{p_{ij} \geq 0 \,|\, i = 1, ..., m; j = 1, \ldots, n\}$ *are nondegenerate (i.e.,* $\mathbb{D}p_{11} \doteq \sigma^2 > 0$*) independent identically distributed and*

$$\sigma^2 < \infty.$$

Now we formulate the main results of the paper.

Theorem 1. *For the stochastic flow shop problem with* m *machines and* n *jobs satisfying* (P)*, an arbitrary permutation schedule* S_π *is asymptotically optimal, the makespan* $T(S_\pi)$ *being estimated as follows:*

$$T(S_\pi) < U \doteq (m-1)p + \left(1 + \frac{(m-1)\ln n}{\sqrt{n}}\right) L \tag{4}$$

almost always with $n \to \infty$.

Theorem 2. *For the stochastic assembly line problem with* m *machines and* n *jobs satisfying* (P)*, an arbitrary permutation schedule* S_π *is asymptotically optimal, the makespan* $T(S_\pi)$ *being estimated as follows:*

$$T(S_\pi) < U \doteq p + \left(1 + \frac{\ln n}{\sqrt{n}}\right) L \tag{5}$$

almost always with $n \to \infty$.

4 Proofs

Proof of theorem 2. The length of the permutation schedule S_π for the assembly line problem can be written as follows ([3]):

$$\begin{aligned} T(S_\pi) &= \max_{i=1,\ldots,m-1} \max_{k=1,\ldots,n} \left(\sum_{j=1}^{k} p_{i\pi_j} + \sum_{j=k}^{n} p_{m\pi_j} \right) \\ &\leq \max_{i=1,\ldots,m-1} \max_{k=1,\ldots,n} \left(\sum_{j=1}^{k-1} p_{i\pi_j} - \sum_{j=1}^{k-1} p_{m\pi_j} \right) + p + L. \end{aligned} \tag{6}$$

Introducing the $(m-1)$-dimensional vectors $\boldsymbol{e}_{\pi_j} = (p_{1\pi_j} - p_{m\pi_j}, \ldots, p_{m-1,\pi_j} - p_{m\pi_j}) \in \mathbb{R}^{m-1}$ $(j = 1, \ldots, n)$ and the norm $\|\cdot\|$ defined for $\boldsymbol{x} = (x_1, \ldots, x_{m-1})$ by

$$\|\boldsymbol{x}\| = \max_{i=1,\ldots,m-1} |x_i|, \tag{7}$$

one can represent the upper bound (6) in the following compact form:

$$T(S_\pi) \leq \max_{k=1,\dots,n-1} \|S_k\| + p + L, \tag{8}$$

where $\boldsymbol{S_k} = \boldsymbol{e}_{\pi_1} + \cdots + \boldsymbol{e}_{\pi_k}$. Thus, to prove (5), it suffices to show that

$$\lim_{n\to\infty} \mathbb{P}\left(\max_{k=1,\dots,n-1} \|\boldsymbol{S_k}\| < \frac{\ln n}{\sqrt{n}} L \right) = 1. \tag{9}$$

It is easy to see that vectors $\{e_{\pi_j}\}_{j=1}^{n-1}$ and the norm defined in (7) satisfy the lemma 1 conditions (see Appendix). Therefore, for some $x \geq 0$

$$\mathbb{P}\left(\max_{k=1,\dots,n-1} \|\boldsymbol{S_k}\| \geq x \right) \leq 2\mathbb{P}\left(\|\boldsymbol{S_{n-1}}\| \geq x \right). \tag{10}$$

It follows from theorem 3 (see Appendix) applied to vectors $\{\boldsymbol{e}_{\pi_j}\}_{j=1}^{n-1}$ that $\mathbb{P}(\|\boldsymbol{S_{n-1}}\| \geq \sqrt{n} \ln^{1/2} n) \to 0$ with $n \to \infty$, which immediately implies

$$\lim_{n\to\infty} \mathbb{P}\left(\max_{k=1,\dots,n-1} \|\boldsymbol{S_k}\| < \sqrt{n} \ln^{1/2} n \right) = 1. \tag{11}$$

Due to (9) and (11), we need to show that the following relation holds:

$$\lim_{n\to\infty} \mathbb{P}\left(L > n \ln^{-1/2} n \right) = 1. \tag{12}$$

Note that (12) holds as soon as the relation

$$\mathbb{P}(L_i > n \ln^{-1/2} n) \to \infty \tag{13}$$

holds for each $i = 1, \dots, m$ with $n \to \infty$. Remind that by (2) L_i represents the sum of independent identically distributed random variables p_{ij} with positive mean. Then, theorem 3 can be applied to the centered random variables $\{p_{ij} - \mathbb{E}p_{ij}\}_{j=1}^{n-1}$, which implies (13) and (12). Relation (9) is implied by (11) and (12).

So, we showed the asymptotic estimate (5). Now to prove the asymptotic optimality of the schedule S_π, we need to show that the estimate U satisfies the following relation for any $\varepsilon > 0$:

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{U - L}{L} < \varepsilon \right) = 1. \tag{14}$$

It is easy to see that (14) implies (3), since the maximum machine load L is the optimum lower bound.

Relation (14) holds, whenever $\mathbb{P}(p/L < \varepsilon) \to 1$ for any fixed $\varepsilon > 0$ with $n \to \infty$. In [2] under even weaker conditions on the p_{ij} distribution (than the condition (P) above), it is shown that

$$\lim_{n\to\infty} \mathbb{P}(Cp < L) = 1$$

for every fixed constant $C > 0$. Thus, the asymptotic optimality of schedule S_π is proved. □

Proof of theorem 1. The proof of theorem 1 is almost identical to the one of theorem 2 with minimal changes. The length of the permutation schedule S_π for the flow shop problem can be written as follows (see [4], for example):

$$\begin{aligned}
C_{\max}(S_\pi) &= \max_{1\le k_1\le\ldots\le k_{m-1}\le n}\left(\sum_{j=1}^{k_1} p_{1\pi_j} + \sum_{k_1}^{k_2} p_{2\pi_j} + \ldots + \sum_{j=k_{m-1}}^{n} p_{m\pi_j}\right)\\
&\le \max_{1\le k_1\le\ldots\le k_{m-1}\le n}\left[\left(\sum_{j=1}^{k_1} p_{1\pi_j} - \sum_{j=1}^{k_1-1} p_{2\pi_j}\right) + \cdots +\right.\\
&\left. + \left(\sum_{j=1}^{k_{m-1}} p_{m-1,\pi_j} - \sum_{j=1}^{k_{m-1}-1} p_{m\pi_j}\right)\right] + L\\
&\le \sum_{i=1}^{m-1} \max_k \left(\sum_{j=1}^{k} p_{i\pi_j} - \sum_{1}^{k-1} p_{i+1,\pi_j}\right) + L. \qquad (15)
\end{aligned}$$

Similar to the proof of theorem 2, we introduce the $(m-1)$-dimensional vectors $\boldsymbol{e}_{\pi_j} = (p_{1\pi_j} - p_{m\pi_j}, \ldots, p_{m-1,\pi_j} - p_{m\pi_j}) \in \mathbb{R}^{m-1}$ $(j = 1, \ldots, n)$ and the norm $\|\cdot\|$ defined for $\boldsymbol{x} = (x_1, \ldots, x_{m-1})$ by

$$\|\boldsymbol{x}\| = \max\left\{\max_i |x(i)|,\ \max_{i,j} |x(i) - x(j)|\right\}, \qquad (16)$$

one can represent the upper bound (15) in the following compact form:

$$T(S_\pi) \le \max_{k=1,\ldots,n-1} \|\boldsymbol{S_k}\| + (m-1)p + L, \qquad (17)$$

where $\boldsymbol{S_k} = \boldsymbol{e}_{\pi_1} + \cdots + \boldsymbol{e}_{\pi_k}$. Like the norm (7), the norm (16) satisfies lemma 1 conditions. Thus, to accomplish the theorem 1 proof, one should repeat the technique used in the proof of theorem 2 (see (6)—(14)). □

5 Remarks and Conclusions

In the paper, there is a simple example of the stochastic approach to machine scheduling problems. The proved theorems give us an efficient tool to solve large scale flow shop and assembly line problems, since an arbitrary permutation schedule can be easily constructed in linear time. Although for the considered problems, there are exist effective approximation algorithms with complexity $O(m^2n^2)$, for other machine scheduling (and discrete optimization) problems the situation may not be that successful. So far, it seems very likely that similar results can be derived for other shop scheduling problems with the minimum makespan objective.

It would be interesting to extend the above results to the case, when the operation processing times are differently distributed random variables. Deriving of the rate function for the probability in (3) remains very interesting open question as well.

6 Appendix

I. The Lévi Inequality. A random vector ξ is *symmetrical* if the distributions of ξ and $-\xi$ coincide. Let $\xi_1, \ldots, \xi_n$ be independent symmetrical random vectors with values in some linear measurable space X. The function $|x|$ with values in in the set $[0; +\infty]$ is a measurable seminorm on X: it is measurable and satisfies the conditions

$$|x+y| \leq |x| + |y|, \; |\alpha x| = |\alpha||x|, \; \alpha \in \mathbb{R}$$

on the set $\{x : |x| < \infty\}$. Consider the random variables

$$S_k = \sum_{j=1}^{k} \xi_j \; (k = 1, \ldots, n), \quad S_n^* = \max_{j=1,\ldots,n} |S_j|.$$

Lemma 1. *For all $x \geq 0$ holds*

$$\mathbb{P}(S_n^* \geq x) \leq \mathbb{P}(|S_n| \geq x).$$

Lemma 1 is formulated and proved in [6], p. 86.

II. Central Limit Theorem. Let $\xi_1, \xi_2, \ldots$ be a sequence of independent identically distributed vectors, $\mathbb{E}\xi_k = 0$, $\sigma^2 = \mathbb{E}\xi_k^T \xi_k$, $S_n = \sum_{k=1}^{n} \xi_k$. Let random vector ζ have Gaussian distribution with parameters $(0, \sigma^2)$.

Theorem 3. *With $n \to \infty$*

$$\frac{S_n}{\sqrt{n}} \Rightarrow \zeta,$$

where symbol $\Rightarrow$ denotes the weak convergence of random vectors.

More details can be found in [1], p. 153.

Acknowledgements

The author expresses his sincere thanks to Sergey Sevastianov for very useful conversations and concerning to above problems and his contribution to improving the style of the paper.

References

1. Borovkov A.A. (1998) Probabilities Theory. Amsterdam: Gordon and Beach.
2. Koryakin R.A. (2003) On the Stochastic Open Shop Problem. A.Albrecht, K.Steinhöfel(Eds.) Stochastic Algorithms: Foundations and Applications, Springer Verlag, 117–124
3. Potts C.N. et al. (1995) The Two Stage Assembly Scheduling Problem: Complexity and Approximation. Oper. Res. **43**, 346–355
4. Sevastianov S.V. (1995) Vector Summation in Banach Space and Polynomial Time Algorithms for Flow Shops and Open Shops. Math. of Oper. Res. **20**, 90–103
5. Sevastianov S.V. (1998) Nonstrict Vector Summation in Multi-Operation Scheduling. Annals of Oper. Res. **83**, 179–211
6. Yurinsky V. (1995) Sums and Gaussian Vectors. Lecture Notes in Math. **1617**, Springer Verlag

An Exact Branch–and–Price Algorithm for Workforce Scheduling

Christoph Stark and Jürgen Zimmermann

Clausthal University of Technology, Department for Operations Research,
Julius-Albert-Str. 2, D–38678 Clausthal-Zellerfeld

Abstract. We consider a generic workforce scheduling problem, where employees are characterized by qualifications. Given a set of shifts for each day, we have to determine for each employee his working days as well as a specific shift for each working day. The overall objective is to find a set of feasible schedules with respect to hard and soft restrictions. We propose an integer multi–commodity network flow formulation for the problem under consideration and show how branch–and–price can be used to solve this problem.

1 Introduction

Workforce scheduling is the process of generating work schedules for a number of employees such that an organization can satisfy the demand for its goods or services. We refer to [3] for a recent survey on workforce scheduling. In this paper, we consider a problem which decidedly extends the so-called days-off scheduling problem, where the number of employees to be assigned to different working patterns has to be determined. Given a set of predetermined shifts for each day of the planning horizon, we have to find a schedule for each employee that specifies his days off, his working days as well as a specific shift for each working day. In particular, we consider employees with different qualifications and individual preferences. Moreover, we distinguish between hard restrictions that must be met and soft restrictions which should be obeyed.

Hard restrictions:

1. No employee may be assigned to a shift he is not qualified for.
2. In between two shifts, there must be at least 11 hours of idle time.
3. No employee may work longer than 10 days in a row.
4. Each employee must receive two days off for every 5 days on duty.

Soft restrictions:

1. At least two days off should be scheduled in a row.
2. A day off should be preceded by another day off or an early shift and should be succeeded by another day off or a late shift.
3. A vacation should be preceded by an early shift and succeeded by a late shift.

Our objective is to find a solution which is feasible with respect to the hard constraints and minimizes violations of the soft constraints. Moreover, working at night or on weekends is awarded with significant extra pay. Thus, a long–term objective is to balance gross wages among employees.

2 Problem Formulation

We consider a planning horizon of $t = 1, \ldots, 14$ days. Let J_t be the set of all shifts that take place on day t and define $J := \bigcup_{t=1}^{14} J_t$. J_t contains exactly one dummy shift which is used to admit assignments of superfluous employees to shifts on day t. With each shift $i \in J$, we associate two events s_i and e_i representing the start and end of shift i, respectively. Let set $J^d \subset J$ contain all dummy shifts. A transition between two shifts $i \in J_t$ and $j \in J_{t'}$, $t < t'$, is associated with the events e_i and s_j. For $t' - t > 1$, a transition represents one or more days off. Transitions violating hard restriction 2 are not considered.

2.1 Multi–Commodity Flow Network

For each employee $k \in K$, we consider a network $G^k = (N^k, A^k)$ (cf. Fig. 1). The set of nodes N^k consists of two nodes s_i and e_i for each $i \in J$ that employee k is suitably skilled for as well as a source α and a sink ω. All employees are sufficiently skilled for the dummy shifts. Nodes s_i and e_i associated to already granted holidays or unacceptable shifts for employee k are omitted in N^k. For two nodes $s_i, e_i \in N^k$, A^k contains a directed arc (s_i, e_i). Moreover, A^k contains the arcs (α, s_i) for all $s_i \in N^k$ and (e_i, ω) for all $e_i \in N^k$. Finally, there is an arc (e_i, s_j) for each transition. For each arc $(i, j) \in A^k$, we introduce an arc weight (cost) c_{ij}^k, which serves to penalize violations of the soft restrictions and takes into consideration employee k's preference and eligibility for individual shifts. Observe that a path p_k from α to ω corresponds to a feasible work sequence with respect to hard restrictions 1 and 2. Overlapping networks G^k, $k \in K$, results in a multi–commodity flow network. While the proposed network is structurally similar to the network formulation in [4], we

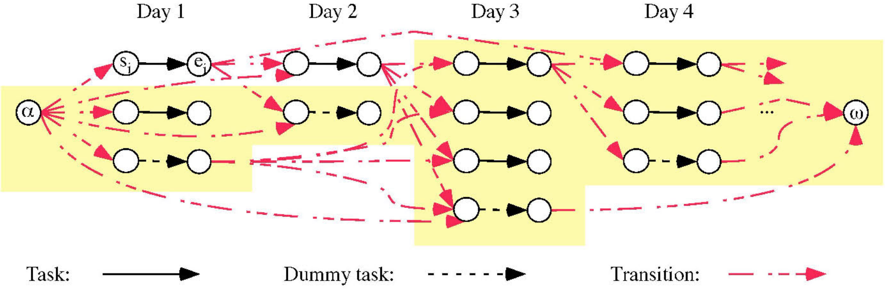


Fig. 1. Sketch of the underlying multi–commodity flow network

consider individual shifts instead of predefined sequences of shifts and days off are accounted for implicitly.

2.2 Resource Constraints

In order to observe hard restrictions 3 and 4, we introduce two resources γ_1 and γ_2. With each arc $(i,j) \in A^k$, we associate a resource consumption r^{γ}_{ij}, $\gamma \in \Gamma := \{\gamma_1, \gamma_2\}$. $\rho^{\gamma k}_i$ is the total resource consumption of resource γ on a path from node α to node i. In case a path from α to i is extended by arc (i,j), $\rho^{\gamma k}_j$ is given by $f^{\gamma}(\rho^{\gamma k}_i, r^{\gamma}_{ij})$. For some nodes $i \in N^k$ we introduce a lower and upper bound for $\rho^{\gamma k}_i$. Notice that without loss of generality r^{γ}_{ij}, a^{γ}_i, b^{γ}_i are the same for each G^k, $k \in K$, while $\rho^{\gamma k}$ depends on k.

Resource γ_1: Maximum number of consecutive work days. No employee may work more than 10 days in a row (hard restriction 3). Thus, with each arc $(s_i, e_i) \in A^k$ we associate a resource consumption $r^{\gamma_1}_{s_i,e_i} := 1$, for each transition (e_i, s_j) that corresponds to at least one day off. We set $r^{\gamma_1}_{e_i,s_j} := -14$, and all remaining $r^{\gamma_1}_{ij} := 0$. We define $f^{\gamma_1}(\rho^{\gamma_1 k}_i, r^{\gamma_1}_{ij}) := \max\{0, \rho^{\gamma_1 k}_i + r^{\gamma_1}_{ij}\}$, and we demand $0 \leq \rho^{\gamma_1 k}_{s_i} \leq 9$ for all $s_i \in N^k$.

Resource γ_2: Number of days off. Hard restriction 4 implies that for every 5 days on duty an employee is entitled to 2 days off. For each arc $(s_i, e_i) \in A^k$, we set $r^{\gamma_2}_{s_i,e_i} := \frac{2}{5}$, for each transition (e_i, s_j) corresponding to at least $d > 0$ days off, we require $r^{\gamma_2}_{e_i,s_j} := -d$, and all remaining $r^{\gamma_2}_{ij} := 0$. We set $f^{\gamma_2}(\rho^{\gamma_2 k}_i, r^{\gamma_2}_{ij}) := \rho^{\gamma_2 k}_i + r^{\gamma_2}_{ij}$, and we require $-2.4 \leq \rho^{\gamma_2 k}_{s_i} \leq 2$ for all $s_i \in N^k$.

2.3 Optimization Model

Let P_k be the set of all paths from α to ω in G^k and x^{p_k} be the flow on path $p_k \in P_k$, $k \in K$. Moreover, let $\delta^{p_k}_{ij} := 1$ if $(i,j) \in p_k$ and $\delta^{p_k}_{ij} := 0$ otherwise. The cost per unit flow on path $p_k \in P_k$ is $c_k(p_k) := \sum_{(i,j)\in p_k} c^k_{ij}$ for all $p_k \in P_k$, $k \in K$. We introduce slack variables d_i (dummy employees) for each shift $i \in J \setminus J^d$, which serve to absorb shifts that are not covered by any $x^{p_k} > 0$. The utilization of dummy employees is penalized with parameter λ. We now propose a set partitioning formulation for the underlying problem:

$$\text{Min.} \quad \sum_{k\in K} \sum_{p_k\in P_k} c_k(p_k) x^{p_k} + \sum_{i\in J\setminus J^d} \lambda d_i \tag{1}$$

$$\text{s.t.} \quad \sum_{p_k\in P_k} x^{p_k} = 1 \qquad k \in K \tag{2}$$

$$\sum_{k\in K} \sum_{p_k\in P_k} \left(\delta^{p_k}_{s_i e_i} x^{p_k}\right) + d_i = 1 \qquad i \in J \setminus J^d \tag{3}$$

$$\sum_{p_k\in P_k} \left(\delta^{p_k}_{ij} x^{p_k}\right) \left(\rho^{\gamma k}_i + r^{\gamma}_{ij} - \rho^{\gamma k}_j\right) \begin{pmatrix} \leq \\ = \end{pmatrix} 0 \;\; (i,j) \in A^k, k \in K, \gamma \in \Gamma \tag{4}$$

$$a_i^\gamma \leq \rho_i^{\gamma k} \leq b_i^\gamma \qquad i \in N^k, k \in K, \gamma \in \Gamma \qquad (5)$$

$$d_i \geq 0 \qquad i \in J \setminus J^d \qquad (6)$$

$$x^{p_k} \in \{0,1\} \qquad p_k \in P_k, k \in K \qquad (7)$$

Observe that restrictions (4) depend on the resource under consideration, i.e., (4) must hold as an inequality for resource γ_1 and (4) is an equation for γ_2. Problem (1) – (7) is difficult to solve due to nonlinear constraints (4) and the huge number of variables x^{p_k}.

3 Branch–and–Price Solution Procedure

Branch–and–price (cf. e.g. [1]) is a powerful method for solving integer programs with large numbers of variables. Its basic idea is to combine column generation, which serves to solve the linear relaxation of the underlying problem, with branch–and–bound.

3.1 Column generation

The linear relaxation of problem (1) – (7) *(master problem)* $\boldsymbol{M}$ is obtained by replacing restrictions (7) by $x^{p_k} \geq 0$ for all $p_k \in P_k, k \in K$. An optimal solution to $\boldsymbol{M}$ provides a lower bound for (1) – (7). The basic concept of column generation is as follows. Instead of $\boldsymbol{M}$, a *restricted master problem* $\boldsymbol{M}'$, which contains only a small subset of decision variables x^{p_k}, is solved. Further columns for $\boldsymbol{M}'$ with reduced objective function coefficients $\tilde{c}_k(p_k) < 0$ are determined by solving a so–called *subproblem.* The process of solving $\boldsymbol{M}'$ and determining and appending favorable columns to $\boldsymbol{M}'$ is repeated until no further columns with $\tilde{c}_k(p_k) < 0$ exist.

The restricted master problem $\boldsymbol{M}'$ is obtained by considering an appropriate subset $P_k' \subset P_k$ of variables x^{p_k}. A proper subset P_k' is given by any basic feasible solution to $\boldsymbol{M}$ or can be generated by constructing a set of distinct paths from α to ω in G^k for each k. We require each path in P_k' to fulfill (4) and (5) and are therefore able to drop these restrictions from $\boldsymbol{M}'$.

An optimal solution to $\boldsymbol{M}'$ provides the shadow prices τ_k for employee k and π_{s_i,e_i} for each arc (s_i, e_i). The reduced objective function coefficients of any resource-feasible path p_k can thus be expressed as

$$\tilde{c}_k(p_k) := \sum_{(i,j)\in p_k} \underbrace{(c_{ij}^k - \pi_{ij})}_{\text{modified cost } \bar{c}_{ij}^k} - \tau_k \quad .$$

An optimal solution of $\boldsymbol{M}$ must satisfy two optimality conditions: (a) $\tilde{c}_k(p_k) \geq 0$ for all feasible $p_k \in P_k, k \in K$, and (b) $\tilde{c}_k(p_k) = 0$ for all p_k with $x^{p_k} > 0$. These conditions imply that τ_k is the length of a shortest path from α to ω in G^k with respect to the modified cost $\bar{c}_{ij}^k$. A path p_k with minimum $\tilde{c}_k(p_k)$ is obtained by determining a resource–feasible shortest path from α to ω in G^k for each k. An appropriate algorithm for the resource–constrained shortest path problem is given in [2]. Its fundamental idea is to use multidimensional labels with label components for the total resource consumption $\rho_i^{\gamma k}$ at a given node i as well as the corresponding length of the path from α to i.

3.2 Branch–and–bound

We use variable dichotomy for branching. The two subproblems $\boldsymbol{M}_1$ and $\boldsymbol{M}_0$ for problem $\boldsymbol{M}$ are obtained as follows.

$\boldsymbol{M}_1$: $x^{p_k} = 1$, *employee k must work schedule p_k.* We delete all variables (columns) $x^{p'_k}$ with $p'_k \in P_k$, $p_k \neq p'_k$ and all variables $x^{p'_k}$ ($p'_k \in P_{k'}$, $k' \in K$, $k' \neq k$), for which there is an arc $(i,j) \in p_k \cap \{J \setminus J^d\}$ with $\delta_{ij}^{p_k} = \delta_{ij}^{p'_k}$. No further columns for employee k are generated in subsequent subproblems.

$\boldsymbol{M}_0$: $x^{p_k} = 0$, *employee k must not work schedule p_k.* We delete variable x^{p_k} from $\boldsymbol{M}$ and need to ensure that path p_k is not generated for employee k in $\boldsymbol{M}_0$ or any subsequent subproblem. Thus, we store a list of forbidden paths for each employee k and each node of our search tree. In case a forbidden path is obtained, we compute the next–shortest path. We refer for instance to [5] for a k–shortest path algorithm, which can be easily adapted towards the case with resource constraints.

Using the proposed branching rule in combination with a FIFO search strategy permits us to quickly obtain good integral solutions, because subproblems towards the bottom of the search tree become a lot easier to solve.

3.3 Preliminary computational results

In order to gain some preliminary computational results we constructed a random testset consisting of problems with 10, 25, 50, 75, and 100 employees and a total of 100, 250, 500, 750, and 1000 shifts throughout the planning horizon, respectively. Table 1 summarizes some of our empirical results, specifically the time needed to solve the initial restricted master problem $\boldsymbol{M}'$ (lower bound) and the time needed to obtain an integer solution. On average, the integer solutions' objective function values deviated between 0.5% and 10% from the lower bound. For the initial subset P'_k of decision variables we simply presumed that all shifts are covered by dummy employees. Note that

number of employees	10	25	50	75	100
Lower bound	2	40	425	1522	4184
Integer solution	3	88	1088	4121	11538

Table 1. Preliminary computational results (average values in seconds)

the integral solution is the first solution obtained in the course of the branch-and-bound algorithm and is thus usually suboptimal. Moreover, it should be noted that approximately 50% of the elapsed time was spent by the LP-solver (CPLEX 9.0) and its subroutines. Generating optimal solutions has proven to be extremely time-consuming due to the trailing-off effect of branch-and-price. However, our results encourage the application of an ϵ-approximative branch-and-bound framework, e.g. with an epsilon of 10%.

4 Conclusion

In this paper we present a multi–commodity flow network formulation for a general workforce scheduling problem. Moreover, we propose an appropriate branch–and–price solution procedure which is able to generate solutions that are on average within 10% of the lower bound in less than 20 minutes for problem instances with 50 employees.

There are several further promising fields of research. A proper continuous planning environment permits us to pursue the objective of gross wage balancing. This can be done by evaluating an employee's deviation from the average wage level in a preprocessing phase and subsequently modifying the arc weights accordingly. Moreover, alternate constraint-based branching schemes may considerably shorten the time required to obtain exact solutions. Further on, it seems worthwhile to investigate the possible application of Lagrange pricing techniques instead of solving a linear relaxation of the underlying problem. Finally, we plan on developing appropriate neighborhood operators for local search. This permits us to hybridize branch–and–price and local search, for instance by using local search to quickly find local optima whenever an integral solution has been found.

References

1. Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H. (1998) Branch–and–Price: Column Generation for Solving Huge Integer Programs. Operations Research **46**, 316–329
2. Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F. (1995) Time Constrained Routing and Scheduling. In: Ball, M.O., et al. Handbooks in OR & MS, Vol. 8., pp. 35–139, Elsevier, Amsterdam
3. Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D. (2004) Staff Scheduling and Rostering: A Review of Applications, Methods and Models. European Journal of Operational Research **153**, 3–27
4. Millar, H.H., Kiragu, M. (1998) Cyclic and Non-Cyclic Scheduling of 12h Shift Nurses by Network Programming. European Journal of Operational Research **104**, 582–592
5. Yen, J.Y. (1971) Finding the k Shortest Loopless Paths in a Network. Management Science **17**, 712–716

A Single Processor Scheduling Problem with a Common Due Window Assignment

Adam Janiak and Marcin Winczaszek

Institute of Engineering Cybernetics, Wrocław University of Technology,
Janiszewskiego 11/17, 50-371 Wrocław, Poland
e-mails: adam.janiak@pwr.wroc.pl, marcin.winczaszek@pwr.wroc.pl

Abstract. We consider a single processor scheduling problem with a common due window assignment. Jobs completed within the due window incur no penalty, while other jobs incur either earliness or tardiness penalties. Boundaries of the due window are decision variables. The objective is to minimize the sum of the total weighted earliness, the total weighted tardiness and due window width penalty. This problem is an extension of the classical Weighted Earliness and Tardiness problem (**WET**). We proved that our problem is NP-hard and presented some properties of an optimal solution. To solve the problem we constructed a dynamic programming algorithm and a fully polynomial time approximation scheme. We also presented a polynomial time algorithm for the case with unit job processing times.

1 Introduction

In this paper we consider a single processor scheduling problem with a common due window assignment. Jobs completed within the due window incur no penalty, while other jobs incur either earliness or tardiness penalties. This problem is an extension of the well known Weighted Earliness and Tardiness problem (**WET**) [5] [3].

Problems with a common due window assignment were studied in [4], [6] and [8]. For comprehensive review of scheduling problems with due date and due window assignment see Gordon *et al* [2].

The considered problem can be stated as follows. Consider a set of n jobs $\mathbf{J} = \{1, \dots, n\}$ to be processed on a single processor, which can handle only one job at a time. Processor idle times are forbidden. Each job j is characterized by processing time p_j and weights α_j, β_j. Let C_j denote the completion time of job j. The objective is to find a sequence of jobs and a left and a right edge of the due window (denoted by q_1 and q_2, respectively) such that the criterion

$$\sum \alpha_j E_j + \sum \beta_j T_j + \gamma(q_2 - q_1)$$

is minimized, where $E_j = \max\{q_1 - C_j, 0\}$ is earliness of job j, $T_j = \max\{C_j - q_2, 0\}$ is tardiness of job j and γ is a due window width penalty. The formulated problem will be referred to as **P**.

We shall also consider two special cases of this problem, namely:
P1 - the case with unit processing times ($p_j = 1$ for each $j \in \mathbf{J}$),
P2 - the case with agreeable weights ($p_i/\alpha_i \leq p_j/\alpha_j \Leftrightarrow p_i/\beta_i \leq p_j/\beta_j$ for each $i, j \in \mathbf{J}$).

The problems are applicable in many manufacturing systems, where the negotiation between the producer and the customer occurs. The negotiation concerns the delivery of the final products. The producer should deliver the products before some established moment and, on the other hand the customer will not receive the products before the other established moment. This results in due window which corresponds to the time frame during which the customer is most willing to take delivery of the products. Products manufactured too early has to be held in inventory until the customer is ready to receive them. This results in such costs as capital, insurance and deterioration costs. On the other hand products manufactured too late incur costs connected with late charges, express delivery, or loss sales.

The remaining part of this paper is organized as follows. In Section 2 we prove that **P2** and **P** are NP-hard and present some properties of an optimal solution of **P**. We show, in Section 3, that **P1** can be solved in polynomial time. In Section 4 we present dynamic programming algorithm solving optimally **P2**. A fully polynomial time approximation scheme for **P2** is described in Section 5. Finally, we summarize the obtained results in Section 6.

2 Preliminary Analysis

In this section we present some properties of an optimal solution of **P** and prove that **P2** and **P** are NP-hard.

Before we start our consideration, let us first introduce some useful notation:
$\mathbf{T} \overset{\Delta}{=} \{j \in \mathbf{J} : C_j > q_2\}$ - a set of tardy jobs,
$\mathbf{E} \overset{\Delta}{=} \{j \in \mathbf{J} : C_j \leq q_1\}$ - a set of early jobs,
$\mathbf{W} \overset{\Delta}{=} \mathbf{J} \setminus (\mathbf{E} \cup \mathbf{T})$ - a set of due window jobs.

Property 1. There exists an optimal solution to **P** where some job completes at q_1.

Proof. Assume that we have an optimal solution where no job finishes at q_1. Let u be a job such that $S_u < q_1 < C_u$. A cost of decreasing q_1 by $\varepsilon \leq q_1 - S_u$ is equal to:

$$\begin{aligned} \Delta_1 = \textstyle\sum_{j \in \mathbf{E}} \alpha_j (q_1 - C_j) + \gamma(q_2 - q_1) - \sum_{j \in \mathbf{E}} \alpha_j ((q_1 - \varepsilon) - C_j) \\ -\gamma(q_2 - (q_1 - \varepsilon)) = \varepsilon(\textstyle\sum_{j \in \mathbf{E}} \alpha_j - \gamma). \end{aligned}$$

Similarly, a cost of increasing q_1 by $\varepsilon \leq C_u - q_1$ is equal to:

$$\Delta_2 = \varepsilon(\gamma - \sum_{j \in \mathbf{E}} \alpha_j) = -\Delta_1.$$

Clearly, $\Delta_1 \leq 0$ or $\Delta_2 < 0$. Increase q_1 if $\Delta_2 < 0$ or decrease q_1 if $\Delta_1 \leq 0$ until $q_1 = C_j$ for some $j \in \mathbf{J}$. Since total penalty does not increase, this schedule is also optimal. □

Property 2. There exists an optimal solution to **P** where some job completes at q_2.

Proof of this property is similar to the proof of Property 1.

Property 3. In an optimal solution of **P**, jobs in set **E** are sequenced in nonincreasing order of p_j/α_j and jobs in set **T** are sequenced in nondecreasing order of p_j/β_j.

This proof may be easily obtained by a simple adjacent jobs interchanging.

Property 4. There exists an optimal solution to **P** where the jobs from **W** are sequenced arbitrarily.

Proof. It follows directly from the fact that the value of objective function is independent of the sequence of jobs processed within the due window. □

Properties 1-4 hold also for the problems **P1** and **P2**, because these problems are special cases of the problem **P**.

Before we prove NP-hardness of problem **P2**, let us introduce the following lemma.

Lemma 1. *If $\gamma > \sum_{j \in \mathbf{J}} \alpha_j$, then $q_1 = q_2$ in an optimal solution of **P2**.*

Proof. Assume that we have an optimal solution σ where $q_2 = q_1 + \varepsilon$. Increase q_1 by ε. Cost of this modification does not exceed $\varepsilon(\sum_{j \in \mathbf{J}} \alpha_j - \gamma) < 0$. Hence, σ is not an optimal solution. □

Now, we will shortly present definition of problem **WET** [3]. Consider a set of n jobs to be processed on a single processor, which can handle only one job at a time. Each job j is characterized by processing time p_j and weight w_j. The objective is to find a sequence of jobs and a common due date (denoted by d) such that the criterion

$$\sum w_j E_j + \sum w_j T_j$$

is minimized, where $E_j = \max\{d - C_j, 0\}$ is earliness of job j, $T_j = \max\{C_j - d, 0\}$ is tardiness of job j.

Theorem 1. *Problem **P2** is NP-hard.*

Proof. Let **P2'** denote a special case of the problem **P2** where $\alpha_j = \beta_j$ for $j = 1, \ldots, n$ and $\gamma > \sum_{j=1}^{n} \alpha_j$. From Lemma 1 follows that $q_1 = q_2$ in an optimal solution of **P2'**. Hence **P2'** is equivalent to **WET**.

We showed that NP-hard **WET** problem is a special case of **P2** problem. Hence, **P2** is NP-hard. □

Since **P2** is a special case of **P**, it follows from Theorem 1 that:

Corollary 1. *Problem **P** is NP-hard.*

3 Unit Processing Times

In this section we show how to solve problem **P1** in polynomial time by transforming it to the assignment problem.

The assignment problem [7] is defined as follows: Assign m jobs to m positions with the least total cost, if job i is assigned to position j with a non-negative cost c_{ij}. Define $x_{ij} = 1$ if job i is assigned to position j and $x_{ij} = 0$ otherwise.

Since $\forall_{j \in \mathbf{J}} p_j = 1$, from Properties 1 and 2 follows that $q_1, q_2 \in \{1, 2, \ldots, n\}$. Consequently, the objective function of the problem **P1** is equal to:

$$\sum_{j=1}^{q_1} (q_1 - j)\alpha_{[j]} + \sum_{j=q_2+1}^{n} (j - q_2)\beta_{[j]} + \gamma(q_2 - q_1).$$

In order to solve **P1** we have to solve a corresponding assignment problem defined as follows:

$$\begin{aligned} &\text{Minimize } \textstyle\sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} x_{ij} \\ &\text{subject to } \textstyle\sum_{i=1}^{m} x_{ij} = 1 \ (j = 1, \ldots, m), \\ &\textstyle\sum_{j=1}^{m} x_{ij} = 1 \ (i = 1, \ldots, m), \\ &x_{ij} = 0 \text{ or } 1 \ (i, j = 1, \ldots, m), \end{aligned}$$

$$\text{where: } m = 3n,$$

$$c_{ij} = \begin{cases} (j-1)\alpha_i & \text{for } i \in [1, n], j \in [1, n], \\ (j-n)\beta_i & \text{for } i \in [1, n], j \in [n+1, 2n], \\ \gamma & \text{for } i \in [1, n], j \in [2n+1, 3n], \\ 0 & \text{for } i \in [n+1, 3n]. \end{cases}$$

The obtained result we interpret as follows. If $x_{ij} = 1$ for $i \in [1, n]$, $j \in [1, n]$, then assign job i to position $q_1 - j + 1$. If $x_{ij} = 1$ for $i \in [1, n]$, $j \in [n+1, 2n]$, then assign job i to position $q_2 + j - n$. Assign the remaining jobs arbitrarily to positions $q_1 + 1, \ldots, q_2$. The values q_1 and q_2 are calculated by $q_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}$ and $q_2 = n - \sum_{i=1}^{n} \sum_{j=n+1}^{2n} x_{ij}$, respectively.

Theorem 2. *Problem P1 can be optimally solved in $O(n^3)$ time.*

Proof. Hungarian method [7] solves optimally the assignment problem in $O(n^3)$ time. The time complexity of the transformation given above is $O(n^2)$. Consequently, **P1** can be solved in $O(n^3)$ time. □

4 A Dynamic Programming Algorithm

In order to solve **P2**, we construct a pseudopolynomial algorithm based on dynamic programming method. In this algorithm, the state variables are: the sum of processing times of the jobs in $\mathbf{E}$ and the sum of processing times of the jobs in $\mathbf{T}$ (denoted by e and t, respectively). The objective function is the minimum sum of penalties delivered by jobs $1, \dots, j$ (we shall refer to it as $F_j(e,t)$).

Reindex the jobs so that $p_1/\alpha_1 \leq \dots \leq p_n/\alpha_n$. For each job j there are three possible choices of scheduling:

- Job j is scheduled so that $C_j \leq q_1$. (From Property 3 follows that in optimal solution job j is processed as the first of the jobs $1, \dots, j$). In this case, value of the objective function increases by $\alpha_j(e-p_j)$ and value of state variable e increases by p_j.
- Job j is scheduled so that $C_j > q_2$. (From Property 3 follows that in optimal solution job j is processed as the last of the jobs $1, \dots, j$). In this case, value of the objective function increases by $\beta_j t$ and value of state variable t increases by p_j.
- Job j is scheduled so that $q_1 < C_j \leq q_2$. In this case, value of the objective function increases by γp_j.

Formal description of the algorithm is given below:

Algorithm D

Step 1. (Initialization). Renumber the jobs so that $p_1/\alpha_1 \leq \dots \leq p_n/\alpha_n$. Set

$$F_0(e,t) := \begin{cases} 0 & \text{if } e = 0 \text{ and } t = 0, \\ +\infty & \text{otherwise.} \end{cases}$$

Then set $j := 1$.

Step 2. (Recurrence). For each $0 \leq e \leq \sum p_j$ and $0 \leq t \leq \sum p_j$ calculate:

$$F_j(e,t) := \min \begin{cases} F_{j-1}(e-p_j,t) + \alpha_j(e-p_j), \\ F_{j-1}(e,t-p_j) + \beta_j t. \\ F_{j-1}(e,t) + \gamma p_j, \end{cases}$$

If $j = n$, then go to Step 3, otherwise set $j := j+1$ and repeat Step 2.

Step 3. (Optimal solution). Calculate value of an optimal solution: $F^* := \min_{e,t} F_n(e,t)$ and find the corresponding optimal solution by using backtracking method.

Theorem 3. *Algorithm D solves optimally problem P2 in $O(n(\sum p_j)^2)$ time.*

Proof. Since we have $\sum_{j=1}^{n} p_j$ different values of parameter e and parameter t, the time complexity of Step 2 is $O((\sum p_j)^2)$. Step 2 is repeated n times. Step 1 and Step 3 need $O(n)$ operations. Consequently, the time complexity of the algorithm is $O(n(\sum p_j)^2)$. Correctness of Algorithm D results from correctness of general technique of dynamic programming [1]. □

5 A Fully Polynomial Time Approximation Scheme

For **P2** we constructed also a fully polynomial time approximation scheme based on an approximation scheme for **WET** introduced by *Kovalyov and Kubiak* [5]. The scheme will be presented shortly. Details may be found in [5].

Reindex the jobs so that $p_1/\alpha_1 \leq \ldots \leq p_n/\alpha_n$. Let $h_l(x)$ denote a function that returns 1 if $l = x$, otherwise returns 0. For each schedule that meets Properties 1-3 we define $x_j = 0$ if $C_j > q_2$, $x_j = 1$ if $C_j \leq q_1$ and $x_j = 2$ if $q_1 < C_j \leq q_2$. Let X be the set of all vectors $x = (x_1, \ldots, x_n)$, where $x_j \in \{0, 1, 2\}$. Define
$X_j = \{x \in X | x_i = 0, i = j + 1, \ldots, n\}$,
$W_0(x) = P_0^E(x) = P_0^T(x) = 0$,
$P_j^E(x) = P_{j-1}^E(x) + p_j h_1(x_j)$,
$P_j^T(x)j = P_{j-1}^T(x) + p_j h_0(x_j)$,
$W_j(x) = W_{j-1}(x) + \alpha_j h_1(x_j) P_{j-1}^E(x) + \beta_j h_0(x_j) P_j^T(x) + \gamma p_j h_2(x_j)$,
$x \in X$, $j = 1, \ldots, n$.

The first j components of each $x \in X_j$ correspond to a schedule of jobs $1, \ldots, j$, $W_j(x)$ is the total penalty of this schedule, $P_j^E(x)$ and $P_j^T(x)$ are the sums of processing times of jobs in set **E** and **T**, respectively. In this way **P2** reduces to the problem of minimizing $W_n(x)$. We shall denote $x \in X$ that minimizes $W_n(x)$ by $x^* = (x_1^*, \ldots, x_n^*)$.

In any iteration j the algorithm partitions set of partial solutions (schedule of jobs $1, \ldots, j$) into disjoint subsets by procedure *Partition*(A, G, δ) presented in paper [5], where $A \subseteq X$, G is a nonnegative integer function on X_j and $0 < \delta \leq 1$. This procedure partitions A into k disjoint subsets $A_1^G, A_2^G, \ldots, A_k^G$ so that $|G(x) - G(x')| \leq \delta \min\{G(x), G(x')\}$ for each $x, x' \in A_i^G$, $i = 1, \ldots, k$. Moreover, in the mentioned paper, it was proved that for $0 < \delta \leq 1$

$$k \leq \log G_{max}/\delta + 2, \tag{1}$$

where $G_{max} = \max_{x \in A} G(x)$.

The algorithm proceeds by iteratively computing a sequence of sets $Y_1, \ldots, Y_n$ where $Y_j \subseteq X_j$, $j = 1, \ldots, n$. Vector x^0 (solution delivered by the algorithm) is then selected from Y_n. We start with set $Y_0 = \{(0, \ldots, 0)\}$. In each iteration j, $j = 1, \ldots, n$, a set is constructed by generating three vectors for each vector $x \in Y_{j-1}$ by adding 0, 1 or 2 in position j of x. The set is then partitioned, using procedure *Partition*, into disjoint subsets in such a way that for every x and x' from the same subset we have: $|W_j(x) - W_j(x')| \leq \varepsilon \min\{W_j(x), W_j(x')\}$, $|P_j^E(x) - P_j^E(x')| \leq (\varepsilon/2n) \min\{P_j^E(x), P_j^E(x')\}$, $|P_j^T(x) - P_j^T(x')| \leq (\varepsilon/2n) \min\{P_j^T(x), P_j^T(x')\}$ and $|h_2(x) - h_2(x')| = 0$. Two solutions are then chosen from each subset to be included in set Y_j; one with minimal $W_j(x)$ value and the other with maximal $W_j(x)$ value. All the remaining solutions in subset are discarded.

Formal description of the algorithm is given below:

Algorithm H_ε

Step 1. (Initialization). Reindex the jobs so that $p_1/\alpha_1 \leq \ldots \leq p_n/\alpha_n$. Set $Y_0 := \{(0,\ldots,0)\}$ and $j := 1$.

Step 2. (Generation of $Y_1,\ldots,Y_n$). For every $x \in Y_{j-1}$ generate three vectors by adding 0, 1 or 2 in position j of vector x. Let us denote the generated set of vectors by Y_j'. For each $x \in Y_j'$ compute $P_j^E(x)$, $P_j^T(x)$ and $W_j(x)$.
If $j = n$, then set $Y_n := Y_n'$ and go to Step 3.
Otherwise perform the following computations:
(Partition Y_j' with respect to W_j) Call *Partition*(Y_j', W_j, ε) to partition Y_j' into disjoint subsets $Y_1^W,\ldots,Y_{k_W}^W$.
(Partition Y_j' with respect to P_j^T) Call *Partition*$(Y_j', P_j^T, \varepsilon/(2n))$ to partition Y_j' into disjoint subsets $Y_1^T,\ldots,Y_{k_T}^T$.
(Partition Y_j' with respect to P_j^E) Call *Partition*$(Y_j', P_j^E, \varepsilon/(2n))$ to partition Y_j' into disjoint subsets $Y_1^E,\ldots,Y_{k_E}^E$.
Portion Y_j' into disjoint subsets Y_1^D, Y_2^D so that $\forall_{x\in Y_1^D} x_j \in \{0,1\}$ and $\forall_{x'\in Y_2^D} x_j' = 2$.
For all quadruples (l_1,l_2,l_3,l_4) such that $Y_{l_1}^W \cap Y_{l_2}^E \cap Y_{l_3}^T \cap Y_{l_4}^D \neq \emptyset$, $1 \leq l_1 \leq k_W$, $1 \leq l_2 \leq k_E$, $1 \leq l_3 \leq k_T$ and $1 \leq l_4 \leq 2$, set $Z_{(l_1,l_2,l_3,l_4),j} := Y_{l_1}^W \cap Y_{l_2}^E \cap Y_{l_3}^T \cap Y_{l_4}^D$.
In each subset $Z_{q,j}$ choose vector $x^{(q,min)}$ and $x^{(q,max)}$ such that
$W_j(x^{(q,min)}) = \min\{W_j(x) | x \in Z_{q,j}\}$,
$W_j(x^{(q,max)}) = \max\{W_j(x) | x \in Z_{q,j}\}$.
Set $Y_j := \{x^{(q,min)}, x^{(q,max)} | q$ is quadruple (l_1,l_2,l_3,l_4) such that $Y_{l_1}^W \cap Y_{l_2}^E \cap Y_{l_3}^T \cap Y_{l_4}^D \neq \emptyset$, $1 \leq l_1 \leq k_W$, $1 \leq l_2 \leq k_E$, $1 \leq l_3 \leq k_T$ and $1 \leq l_4 \leq 2\}$ and $j := j+1$. Repeat Step 2.

Step 3. (Solution). Select vector $x^0 \in Y_n$ such that $W_n(x^0) = \min\{W_n(x)| x \in Y_n\}$.

Lemma 2. *The number of different quadruples* (l_1,l_2,l_3,l_4) *such that* $Y_{l_1}^W \cap Y_{l_2}^E \cap Y_{l_3}^T \cap Y_{l_4}^D \neq \emptyset$, $1 \leq l_1 \leq k_W$, $1 \leq l_2 \leq k_E$, $1 \leq l_3 \leq k_T$ *and* $1 \leq l_4 \leq 2$ *does not exceed* $2k_W(k_E + k_T)$.

The proof is omitted since it is almost identical to the proof of analogical lemma presented in [5].

Theorem 4. *Algorithm* H_ε *finds* $x^0 \in X$ *such that* $W_n(x^0) \leq (1+\varepsilon)W_n(x^*)$.

We may easily obtain the proof of this theorem from the proof of analogical theorem presented in [5]. It is enough to notice that from the way of partitioning Y_j' into subsets Y_1^D, Y_2^D follows that

$$\forall_{x,x'\in Z_{q,j}} |h_2(x) - h_2(x')| = 0$$

and to make some minor changes in few expressions.

Theorem 5. *Algorithm H_ε runs in $O(n^2 \log^3(\max_j\{n, 1/\varepsilon, p_j, \alpha_j, \beta_j, \gamma\})/\varepsilon^2)$ time.*

Proof. From [5] we know that the time complexity of iteration j of Step 2 is equal to $O(f_{j-1}\log(f_{j-1}))$, where f_j is the number of different quadruples (l_1, l_2, l_3, l_4) such that $Y^W_{l_1} \cap Y^E_{l_2} \cap Y^T_{l_3} \cap Y^D_{l_4} \neq \emptyset$ in iteration j. By Lemma 2, f_j does not exceed $2k_W(k_E + k_T)$.

Finally, from equation (1), $k_E, k_T \leq 2n\log(\max\{P^E_{j-1}(x), P^T_{j-1}|Y'_{j-1}\})/\varepsilon + 2 \leq 2n\log(\max_j\{n \cdot p_j\})/\varepsilon + 2 \leq 4n\log(\max_j\{n, p_j\})/\varepsilon + 2$ and $k_W \leq \log(\max\{W_{j-1}(x)|Y'_{j-1}\})/\varepsilon + 2 \leq \log(\max_j\{n^2 \cdot p_j \cdot \max\{\alpha_j, \beta_j, \gamma\}\})/\varepsilon + 2 \leq 4\log(\max_j\{n, p_j, \alpha_j, \beta_j, \gamma\})/\varepsilon + 2$ for each $j = 1, \ldots, n$. Thus, we have $f_{j-1} \leq O(n\log^2(\max\{n, p_j, \alpha_j, \beta_j, \gamma\})/\varepsilon^2)$. Consequently, iteration j of Step 2 requires $O(n\log^3(\max\{n, 1/\varepsilon, p_j, \alpha_j, \beta_j, \gamma\})/\varepsilon^2)$ time and the overall time complexity of H_ε is $O(n^2\log^3(\max\{n, 1/\varepsilon, p_j, \alpha_j, \beta_j, \gamma\})/\varepsilon^2)$. □

6 Conclusions

In this paper we considered a single processor scheduling problem with a common due window assignment. We described some properties of an optimal solution and proved that the problem is NP-hard. We also presented a pseudopolynomial time algorithm based on dynamic programming method and a fully polynomial time approximation scheme for the case with agreeable weights. Moreover, we proved that the case with unit processing times can be solved in polynomial time by transformation to the assignment problem.

References

1. Bellman, R. *Dynamic Programming* Princeton University Press, Princeton, NJ, 1957.
2. Gordon, V., Proth, J.-M., Chu, C. (2002) A survey of the state-of-the-art of common due date assignment and scheduling research. *European Journal of Operational Research*, **139**, 1–25.
3. Hall, N. G., Posner, M. E. (1991) Earliness-tardiness scheduling problems, i: weighted deviation of completion times about a common due date. *Operations Research*, **39**, 836–846.
4. Janiak, A., Marek, M. (2001) Multi-machine scheduling problem with optimal due interval assignment subject to generalized sum type criterion. Operations Research Procedings 2001. Springer, 207–212.
5. Kovalyov, M.Y., Kubiak, W. (1999) A fully polynomial approximation scheme for the weighted ealiness-tardiness problem. *Operations Research*, **47**, 757–761.
6. Kramer, F.-J., Lee, C.-Y. (1994) Due window scheduling for parallel machines. Mathematical Computer Modelling. **20**, 69–89.
7. Kuhn, H. W. (1995) The hungarian method for the assignment problem. *Naval Research Logistics*, **2**, 83–97.
8. Liman, S.D., Panwalkar, S.S., Thongmee, S. (1998) Common due window size and location determination in a single machine scheduling problem. Journal of the Operational Research Society. **49**, 1007–1010.

Modeling SMEs' Choice of Foreign Market Entry: Joint Venture vs. Wholly Owned Venture

Xuemin Zhao and Reinhold Decker

Department of Economics and Business Administration, University of Bielefeld.

Abstract

Starting with a comprehensive analysis of empirical and theoretical papers on market entry mode choice, as well as by referring to the characteristics of small and medium sized enterprises (SMEs), we develop a new quantitative model that represents SMEs' choice between joint venture (JV) and wholly owned foreign venture (WOFV). With the help of this model, we deduce some useful propositions for decision makers, both in companies concerned and in economic policy.

Keywords: Foreign market entry mode choice; Quantitative model; Small and medium sized enterprises.

1. Literature review

Originated from the study of multinational enterprises market entry mode choice studies, theoretical or empirical, are very popular in marketing literature. Theoretical studies aim to explain how multinationals should choose their entry mode or to identify the determinants of entry mode choice. Empirical studies aim to justify the existing theories through data analysis.

The available theoretical models take two forms in general, i.e. qualitative and quantitative. The former are conceptual and abundant. The latter are mainly game theoretical and rare. However, currently there is no established entry mode theory. Zhao and Decker (2004) have indicated this by a comprehensive analysis on the strengths and weaknesses of existing models. Furthermore, the existing models are designed mainly for multinationals but not for SMEs (Kumar and Subramaniam 1997; Nakos and Brouthers 2002).

The well-known qualitative models are the stage of development (SD) model, the transaction cost analysis (TCA), the ownership, location and internalization (OLI) model and the organization capacity (OC) model (Zhao and Decker 2004). However, these models have some common weaknesses. They largely ignore the decision making process. Moreover, they also largely ignore the role of the decision maker in the process of entry mode choice. A process-oriented model worth to be mentioned here is the so-called hierarchical (H) model (Kumar and Subramaniam 1997). However, the model itself does not explain how entry mode choice is actually made in reality.

Among the very few quantitative models entry mode choice was usually analyzed in a two or three-firm economy, which enlightens the effects of strategic interactions (Görg 1998; Müller 2001). The optimal entry mode is determined by comparing the expected profits of each alternative in equilibrium. However, their shared shortcoming results from the fact that firms do not really play in such an abstract economy. Furthermore, very few of these papers have investigated the choice between JV and WOFV.

Empirical studies agree with each other neither on what factors are determinative nor on what influences some factors might exert on entry mode choice. Zhao and Decker (2004) present a deeper discussion on the classification of factors investigated, the possible relations identified as well as the conflicts between theory and empirics observed.

In summary, an explanatory framework tailored for SMEs is still indispensable. The new model we aim to develop in this paper explores the factors that influence entry mode choice from different perspectives, in particular the decision maker, the organization as well as the environment. Moreover, we do not ignore the decision making process. In this respect, our approach is both process- and content-oriented.

2. A new market entry mode choice model

SMEs are small; therefore, they operate in a competitive market. Comparatively, they are simple in their organizational structures and organizational goals. Thus, managers have no managerial discretion in their entry mode choice, i.e. they represent the SME completely. Therefore, they adopt an entry mode that maximizes their utilities and the firm's profit as well. However, the mangers face time, information and resource constraints for their decision-making (Kumar and Subramaniam 1997). In essence, they are boundedly rational and they have the difficulty of evaluating all the alternatives at a time. Therefore, they limit their sights to the most promising alternatives in each decision. Against this background, the present paper focuses on the choice between JV and WOFV by assuming that equity entry was selected on the first decision level.

The difference between JV and WOFV is, in essence, the difference in ownership ratios (Anderson and Gatignon 1986). Therefore, the choice between JV and WOFV is actually the choice of optimal ownership ratio. To construct a simple but meaningful model we make the following assumptions:

1. The SME's ownership ratio of operation in the host country is denoted by θ, with $\theta \in (0,1]$, i.e. there is no limitation on ownership ratio θ in the host

country. If $\theta \in (0,0.95)$, the SME enters as a JV, otherwise it enters as a WOFV.

2. The decision maker has a constant absolute risk averse (CARA) utility, i.e. $U(R) = -e^{-\gamma R}$, where R is the aggregate profit of the SME's operations home and abroad. γ is the parameter of risk aversion, and it is positive.
3. The company possesses a simple production technology producing one output $q(x_1, x_2)$ with capital input x_1 and monetary normalization of all non-capital inputs x_2. These two inputs are perfect substitutes. The output increases with each input respectively and a decreasing marginal productivity is assumed. In the following context superscripts f and h denote the parameters of foreign and home country technologies respectively. The capital input of host country production ($x_1^f > 0$) might result from two sources, namely the investment of the SME (θx_1^f) and that of its partner in the host country ($(1-\theta)x_1^f$). Further, r^f, r^h, w^f, w^h , F^f, and F^h denote inputs cost rates and fixed costs of production in these two countries respectively.
4. The output price p^f is assumed to be a random variable with normal distribution, i.e. p^f □ (μ, σ^2) and $p^f > 0$. Other variables are constant and given. The risk of operating abroad is represented by the variance (σ^2) of the output price.
5. The allocation of profits between the SME and its partner in the host country depends on the ratio of ownership (θ). The SME's profits made abroad and at home are taxed separately without any overlapping, with $t \in [0,1)$. Finally, θR^f and R^h are the SME's profits from investments in foreign and home country respectively.

With the above assumptions, the decision-making problem can be formulated as:

$$Max\ E\ U(\theta R^f + R^h) \tag{2.1}$$

$$\text{s.t.}\quad 0 < \theta \leq 1, \tag{2.2}$$

$$\theta * x_1^f + x_1^h = X_1\ , \tag{2.3}$$

where X_1 is the available capital of the SME for allocation between the home and the host country. The non-capital inputs x_2 are not restricted. Then we can specify θR^f and R^h as follows:

$$\theta R^f = \theta\left[(p^f * q^f(x_1^f, x_2^f) - r^f * x_1^f - w^f * x_2^f - F^f) * (1 - t^f)\right], \tag{2.4}$$

$$R^h = (p^h * q^h(x_1^h, x_2^h) - r^h * x_1^h - w^h * x_2^h - F^h) * (1 - t^h) . \tag{2.5}$$

With the assumption that p^f is the only normally distributed random variable in Eq. (2.4) we can conclude that θR^f and $R = \theta R^f + R^h$ have normal distributions as well. So the mean and the variance of R are:

$$\overline{R} = \theta\left[\mu * q^f(x_1^f, x_2^f) - r^f * x_1^f - w^f * x_2^f - F^f\right] * (1 - t^f) + R^h , \tag{2.6}$$

$$Var(R) = \theta^2 * (1 - t^f)^2 * (q^f)^2 * \sigma^2 . \tag{2.7}$$

The assumption of CARA utility together with the normal distribution of the aggregate profit gives rise to a mean-variance utility function where the company's expected utility is a linear function of the mean and the variance of the aggregate profit (cf. Sargent 1987). Therefore, we have:

$$Max\ E\ U(\theta R^f + R^h) \Leftrightarrow Max \left[\overline{R} - \gamma * Var(\bar{R})/2\right] \tag{2.8}$$

under constraints (2.2) and (2.3).

By using the Kuhn-Tucker methodology, we can solve this problem and get:

$$\theta^* = \frac{(1-t^f)*(\mu * q^f - r^f * x_1^f - w^f * x_2^f - F^f) - (1-t^h) * x_1^f * (p^h * q^h_{x_1^h} - r^h) - \lambda_1 + \lambda_2}{\gamma * (1-t^f)^2 * (q^f)^2 * \sigma^2} , \tag{2.9}$$

$$\theta^* = 1, \tag{2.10}$$

$$\theta^* = 0. \tag{2.11}$$

Eq. (2.10) is explicitly the binding condition where the entry mode is defined as WOFV, and Eq. (2.11) is inconsistent with our assumption, therefore it is not a solution. Eq. (2.9) is the optimal ownership ratio that defines the corresponding entry mode, because the negative second order derivative ensures its sufficiency:

$$\partial^2 L(\theta, \lambda_1, \lambda_2) / \partial\theta^2 = (x_1^f)^2 (1 - t^h) p^h q^h_{x_1^h x_1^h} - \gamma (1 - t^f)^2 (q^f)^2 \sigma^2 < 0. \tag{2.12}$$

The Kuhn-Tucker theorem implies that $\lambda_2 = 0$, if $\theta > 0$. Then θ^* can be re-expressed as:

$$\theta^* = \frac{(1-t^f)*(\mu * q^f - r^f * x_1^f - w^f * x_2^f - F^f) - (1-t^h) * (p^h * q^h_{x_1^h} - r^h) * x_1^f - \lambda_1}{\gamma * (1-t^f)^2 * (q^f)^2 * \sigma^2} . \tag{2.13}$$

With the assumption that θ is positive, we can also explicitly conclude that θ^* is also positive.

3 Implications and propositions

To identify the possible influences of some observable parameters (such as t and r) or some non-observable parameters (such as γ and σ) on decision variable θ^*, we need to do some comparative statics analysis.

The existing empirical results exemplify that entry mode choice is dependent on risk aversion (Osland et al. 2001; Bhaumik and Gelb 2003). To go into this matter we differentiate θ^* with respect to the risk adverse parameter γ and get:

$$\theta^*_{\gamma} = -\theta^* / \gamma < 0. \tag{3.1}$$

This negative relation is widely accepted in prior academic research and in practice as well. However, how sensitive is θ^* with respect to γ ? The elasticity of θ^* with respect to γ gives us the answer:

$$El_{\theta^*\gamma} = (\partial\theta^* / \partial\gamma) * (\gamma / \theta^*) = -1. \tag{3.2}$$

Together with the qualitative cognitions from the relevant literature, we can make the first proposition.

> ***Proposition 1:*** *Given a sufficient incentive to invest in the host country (θ^* >0), then the more risk averse the decision maker is, the less likely he adopts a high equity entry mode, such as WOFV. A reduction of the existing risk aversion, e.g. due to an increasing experience of the decision maker or due to the substitution of the decision maker by a less risk averse one, leads to a proportional increase of the optimal ownership ratio θ^*.*

By following the same procedures, we deduce proposition 2 describing the relation between the optimal ownership ratio θ^* and the estimated risk σ.

> ***Proposition 2:*** *Given a sufficient incentive to invest in the host country (θ^* >0), and the higher the estimated risk of operation in the host country is, the less likely the decision maker will adopt a high equity entry mode, such as WOFV. A reduction of the estimated risk of operation in the host country, e.g., due to less uncertainty about the host country market as a result of learning effects or due to the maturity of the host country market, leads to an over-proportional increase of optimal ownership ratio θ^*.*

In our context, the expected profit of foreign operation is deducted by the potential profit of investing the same amount of capital in the home country. This difference is called the 'opportunity cost' adjusted expected profit and it equals the following expression:

$$R_{adj} = (1-t^f)*(\mu * q^f - r^f * x_1^f - w^f * x_2^f - F^f) - (1-t^h)*(p^h * q^h_{x_1^h} - r^h) * x_1^f. \tag{3.3}$$

Existing papers (e.g. Müller 2001; Eicher and Kang 2002) postulate that the expected profit determines the entry mode choice. To see the truth, we do the same procedures and come to the following implication:

> ***Proposition 3:*** *The optimal entry mode choice is positively affected by the 'opportunity cost' adjusted expected profit of operation in the host country market. The more profit a SME can earn by investing in the host country compared with what it could earn by investing in the home country, the more likely a high equity entry mode is adopted. A change of the adjusted expected profit leads to an over-proportional change of θ^* in the same direction.*

Examining Eq. (3.3) more closely, we can easily find how the 'opportunity cost' adjusted expected profit is influenced systematically by the parameters of the home country (e.g. p^h, $q^h_{x_1^h}$, t^h, and r^h) and by those of the host country (e.g. μ, q^f, r^f, w^f, F^f, and t^f) as well. Explicitly those parameters are meaningful and powerful measures of market attractiveness. Together with proposition 3 we can make the following proposition:

> ***Proposition 4(a):*** *The more attractive the host country market is, the more likely a SME adopts a high equity entry mode; and the other way around, the more attractive the home country market is in comparison with the host country market, the less likely a high equity entry mode is adopted.*

However, to know that θ^* is negatively related with some observable parameters, such as r^f and t^f, is just half of the story. The crucial question is that how sensitive the optimal ownership ratio is with respect to r^f and t^f.

By reformulating Eq. (2.13), we can show that θ^* is a strictly downward-sloping linear function of r^f, i.e. $\theta^* = -a * r^f + b$, with $a > 0,\ b > 0$. Therefore, even though the slope of the linear function θ^* is constant, the elasticity varies along the respective curve (Perloff 2001). The elasticity of θ^* with respect to r^f is the more negative, the higher r^f is, and vice versa. Thus, through some computations we receive proposition 4(b) describing this variation.

> ***Proposition 4(b):*** *Within the interval (0, b/a) capital cost rate r^f induces a varying sensitivity of the optimal ownership ratio θ^*. Meeting $r^f \in [0, b/2a)$ entitles the SME decision makers to deal with the choice of the entry mode more liberally due to the inferior elasticity. However, if $r^f \in (b/2a, b/a)$ the decision makers in SMEs as well as in economic policy of the host country are well advised to pay special attention to this parameter due to its over-proportional negative effect on optimal owner-*

ship ratio θ^*, and thus the investment behavior of foreign companies overall.

In an analogous manner, we can deduce the formal relationships that indicate us how sensitive the optimal ownership ratio is with respect to variations of t^f. For simplicity, we denote the expected profit of foreign operation with A, and the sum of the potential profit of investing the same amount of capital in the home country and the Lagrange multiplier λ_1 with B. Then the final proposition can be written as:

Proposition 4(c): *With other variables being constant a change of t^f from 0 to 1 induces a varying sensitivity of optimal ownership ratio θ^* with respect to t^f. Particularly, if t^f is lower than a 'critical' value $(A-B)/(A+B)$, the optimal entry mode is less dependent on t^f but more dependent on other parameters, to which the SME decision makers should pay attention. On the other hand if t^f exceeds this threshold one should be aware of the possible effects of t^f on the optimal entry mode. The same applies for economic policy in the host country regarding the implications for the foreign investment climate. Finally, when t^f takes a value close to 1 a non-equity entry mode should be taken into consideration except that there are some relevant non-profit objectives.*

4. Conclusions

Through analyzing existing theories as well as empirical studies on market entry mode choice, we detected an explicit need of models that are suited to support the entry mode choice of SMEs. Starting from the relevant characteristics of SMEs we developed a simple mathematical decision-making model which indicates how the choice between JV and WOFV could actually be made. Special attention was devoted to the investigation of qualitative as well as quantitative relationships between the optimal ownership ratio and some important parameters. These parameters have been explored from different aspects, in particular the decision maker, the organization, and the economical environment in which the former two are embedded. The resulting propositions offer the SMEs managers and economic policy makers in the host countries as well, some useful implications on their decision-making. Among other things we could show the necessity of considering the so-called 'opportunity costs' of an investment in the host country during the process of entry mode choice.

Nevertheless, the explanatory power of the model strongly depends on the underlying assumptions. In this sense, it is more normative than descriptive. Future research could examine the effect of relaxing some of these assumptions. For example, the influence of managerial discretion on entry mode choice can be studied by relaxing the goal alignment. Organizational forms can mediate the transaction costs (Williamson 1975) that are assumed to be dominant determinants of the entry mode choice (Anderson and Gatignon 1986). Therefore, relaxing the simple organizational structure might induce some new implications on entry mode choice. Finally, some new insights might come out, when taking the host country's policy restriction on entry mode choice into account.

References

[1]Anderson E, Gatignon H (1986) Modes of Foreign Entry: A Transaction Cost Analysis and Propositions. JIBS 17:1 - 26

[2]Bhaumik SK, Gelb S (2003) Determinants of MNC's Mode of Entry into an Emerging Market: Some Evidence from Egypt and South Africa. DRC Working Papers No.13 Foreign Direct Investment in Emerging Markets. Center for New and Emerging Markets, London Business School

[3]Eicher T, Kang JW (2002) Trade, Foreign Direct Investment or Acquisition: Optimal Entry Modes for Multinationals. University of Washington

[4]Görg H (1998) Analyzing Foreign Market Entry: The Choice between Greenfield Investment and Acquisition, Technical Paper No.98/1, Tritiny Economic Papers Series

[5]Kumar V, Subramaniam V (1997) A Contingency Framework for the Mode of Entry Decision. JWB 32 (1): 53 - 72

[6]Müller T (2001) Analyzing Modes of Foreign Market Entry: Greenfield Investment versus Acquisition. Discussion Paper in Economics, 2001-01, University of Munich

[7]Nakos G, Brouthers KD (2002) Entry mode choice of SMEs in Central and Eastern Europe. Enterpreneurship Theory and Practice 27(1): 47-64

[8]Osland GE, Taylor CR, Zou SM (2001) Selecting International Modes of Entry and Expansion. Marketing Intelligence and Planning 19 (3):153 - 161

[9]Perloff JM (2001) Microeconomics. Addison Wesley, Boston

[10]Sargent TJ (1987) Macroeconomic Theory – 2 edn. Academic Press, Boston

[11]Williamson O (1975) Markets and Hierarchies. Free Press, New York

[12]Zhao XM and Decker R (2004) Choice of Foreign Market Entry Mode: Cognitions from Empirical and Theoretical Studies. Discussion Paper No. 512, University of Bielefeld

Pattern Detection with Growing Neural Networks – An Application to Marketing and Library Data

Reinhold Decker and Antonia Hermelbracht

Department of Economics and Business Administration
University of Bielefeld, 33615 Bielefeld, Germany

Abstract. This paper introduces a new growing neural network for pattern detection which bears certain resemblances to the growing neural gas network suggested by Fritzke (1995) [2]. However, the algorithm at hand is more parsimonious with respect to the number of parameters to be specified a priori. Thus it is largely autonomous regarding the data-driven construction of the final network topology which unburdens the user significantly. To demonstrate its performance and adaptability the new algorithm is applied to real classification tasks in lifestyle analysis and media usage analysis.

1 Introduction

In the last two decades self-organizing neural networks have become an important part of the data analytical instruments in natural and social sciences. Recent improvements aim both at the elimination of existing methodological problems and the alignment of available algorithms to specific areas of application. Even though diverse methodological difficulties have been solved by algorithms such as the neural gas network (Martinetz, Schulten 1991) [1], the growing neural gas network (Fritzke 1995) [2], the growing hierarchical self-organizing map (Dittenbach et al. 2002) [3], as well as the grow when required neural network (Marsland et al. 2002) [4], the efficient determination of adequate parameter settings still continues to be a crucial problem.

Against this background the paper at hand presents a contribution to the solution of this problem. Specifically, we are going to introduce a self-organizing neural network where the number of parameters to be preset by the user at the beginning of the adaptation process is limited to two. The so-called Growing Neural Network with Autonomous Parameter Selection (GNNAPS) determines all controlling parameters apart from the data compression level and the number of iterations required autonomously.

The practical benefits of this approach are demonstrated by means of two real data sets covering different areas of application. Firstly, we are going to detect consumer types in the framework of lifestyle analysis. The reliable identification of consumption patterns requires special attention in marketing due to the interrelations between purchase intention and purchase behaviour. Secondly, the detection of media usage patterns in academic libraries will be shown. Shrinking budgets and increasing costs cause the library managements more and more to adopt marketing research methods to attain deeper understanding of book usage behaviour. The data-driven determination of meaningful book usage patterns may be helpful with regard to the optimization of future media procurement and the adaptation of current services to changing user needs in academic as well as public libraries.

The remainder of this paper is structured as follows. In the next section we give a brief description of the main components of the new algorithm, followed by sketchings of selected findings of both empirical applications. The paper concludes with a brief review of the achieved cognitions.

2 A growing neural network with autonomous parameter selection

The methodology of data compression and feature extraction underlying the new algorithm is vector quantization, the basic idea of which is to represent J input vectors $\boldsymbol{t}_j = (t_{j1}, \ldots, t_{jk}, \ldots, t_{jK})$ (with $j \in \{1, \ldots, J\}$) of a data set Ω by an as small as possible number of weight vectors $\boldsymbol{\eta}_h = (\eta_{h1}, \ldots, \eta_{hk}, \ldots, \eta_{hK})$ (with $h \in \{1, \ldots, H\}$ and $H < J$). To this end, Ω is divided into H classes Ω_h, where each class is represented by a weight vector $\boldsymbol{\eta}_h$. Each input vector $\boldsymbol{t}_j$ is classified into one class Ω_h, or rather assigned to one weight vector $\boldsymbol{\eta}_h$, such that the distance between the input vector and the respective weight vector is minimal (Kohonen 2001) [5]. The only decisions the user has to make in advance when applying the algorithm, apart from indicating the dimensionality K of the data, concern the required data compression level CL $(0 < CL < 1)$ and the maximum number of iterations L (with index $l \in \{1, \ldots, L\}$) considered to be necessary.

At the beginning of the adaptation process, i.e. in iteration $l = 1$, the neural network, or rather the associated set of units $\mathcal{U}$, contains two non-connected units u_1 and u_2, i.e. $\mathcal{U} = \{u_1, u_2\}$, with associated weight vectors $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$, both initialized with positive random values. The firing counters and the training requirements of these units are initialized according to $y_1 = y_2 = 0$ and $w_1 = w_2 = 1$. The set $\mathcal{C}$ of connections between units is empty. Both sets together determine the topological structure of the neural network.

Furthermore, let S_{Max} be the maximum Euclidean distance between two input vectors and S_{Min} the corresponding minimum. The decision as to whether or not a new unit should be added to the network during an iteration is assumed to depend on the activity of the best matching ('winning') unit, i.e. the one with the smallest distance from the current input vector (Marsland et al. 2002) [4]. The smaller this distance, the higher the activity is. If the activity of the winning unit falls below the threshold $v_{Thres} = \exp\big(-\big(\frac{S_{Max}}{2} - \big(\frac{S_{Max}}{2} - S_{Min}\big)\cdot(1-CL)^{\frac{1}{4}}\big)\big)$, we take this as a hint at an insufficient fit between the winning unit and the current input vector. Finally, we define learning rates $\epsilon_{Best} = 0.1 + \frac{1}{2}\big(1 - \exp\big(-\frac{CL\cdot S_{Max}}{\sqrt{L}}\big)\big)$, with an anchor point at 0.1, and $\epsilon_{Sec} = \frac{\epsilon_{Best}}{\sqrt{L}}$ for the best matching unit ('first winner') and the second best matching unit ('second winner') regarding the current input. With these presets one iteration of the algorithm includes the following steps:

1. Randomly select an input vector $\boldsymbol{t}_j = (t_{j1}, \dots, t_{jK})$, with $j \in \{1, \dots, J\}$, from data set Ω and calculate the Euclidean distance $dist(\boldsymbol{t}_j, \boldsymbol{\eta}_h) = \sqrt{\sum_{k=1}^{K}(t_{jk} - \eta_{hk})^2}$ to all units $u_h \in \mathcal{U}$ of the network. The first and second smallest distance determine the best matching unit $u_{h_{Best}}$ and the second best matching $u_{h_{Sec}}$, with $h_{Best} = \arg\min_{h|u_h \in \mathcal{U}} dist(\boldsymbol{t}_j, \boldsymbol{\eta}_h)$ and $h_{Sec} = \arg\min_{h|u_h \in \mathcal{U} \setminus u_{h_{Best}}} dist(\boldsymbol{t}_j, \boldsymbol{\eta}_h)$.
2. Calculate the activity of the best matching unit regarding the current input vector according to $v_{h_{Best}} = \exp\big(-dist(\boldsymbol{t}_j, \boldsymbol{\eta}_{h_{Best}})\big)$.
3. Calculate the threshold for the training requirement according to
   $w_{Thres} = \epsilon_{Best} \cdot \big(1 - \frac{1}{\sqrt{|\mathcal{U}|}}\big)$.
4. Two non-connected winning units (first and second winner) will be connected, either if the distance between the respective weight vectors is so small that no further unit can be inserted there, or if at least one of both is not yet adapted sufficiently:
   If $\exp\left(-\frac{dist(\boldsymbol{\eta}_{h_{Best}}, \boldsymbol{\eta}_{h_{Sec}})}{2}\right) \geq v_{Thres}$
   or $w_{h_{Best}} \geq w_{Thres}$ or $w_{h_{Sec}} \geq w_{Thres}$ then do
      If $(h_{Best}, h_{Sec}) \notin \mathcal{C}$ then do $\mathcal{C}^{New} = \mathcal{C}^{Old} \cup \{(h_{Best}, h_{Sec})\}$ end.
      $a_{h_{Best}h_{Sec}} = 0$ (initialization of the age of the connection)
   end.
5. If both the activity and the training requirement of the first winner fall below their thresholds and if, at the same time, the current number of units is smaller than the overall number of input vectors to be represented, then a new unit (weight vector) is added to the network, i.e.:
   If $v_{h_{Best}} < v_{Thres}$ and $w_{h_{Best}} < w_{Thres}$ and $|\mathcal{U}| < J$ then do
      $\mathcal{U}^{New} = \mathcal{U}^{Old} \cup \{u_{h_{New}}\}$, $\boldsymbol{\eta}_{h_{New}} = \frac{1}{2}(\boldsymbol{\eta}_{h_{Best}} + \boldsymbol{t}_j)$,
      $\mathcal{C}^{New} = \mathcal{C}^{Old} \setminus (h_{Best}, h_{Sec})$,
      $\mathcal{C}^{New} = \mathcal{C}^{Old} \cup \{(h_{New}, h_{Best}), (h_{New}, h_{Sec})\}$,
      $y_{h_{New}} = 0$, $w_{h_{New}} = 1$, $y_{h_{Best}} = 0$, $w_{h_{Best}} = 1$
   end.
   (If so, the firing counter $y_{h_{New}}$ and the training requirement $w_{h_{New}}$ of the new unit will be initialized and the firing counter $y_{h_{Best}}$ as well as the training requirement $w_{h_{Best}}$ of the first winner unit will be reset.)
6. If no new unit is added in step 5, the weight vector of the first winner as well as those of its topological neighbors are adapted:
   $\boldsymbol{\eta}_{h_{Best}}^{New} = \boldsymbol{\eta}_{h_{Best}}^{Old} + \Delta\boldsymbol{\eta}_{h_{Best}}$, $\Delta\boldsymbol{\eta}_{h_{Best}} = \epsilon_{Best} \cdot w_{h_{Best}}^{\frac{\ln(l+\exp(1))}{\ln(L+\exp(1))}} \cdot (\boldsymbol{t}_j - \boldsymbol{\eta}_{h_{Best}})$,
   $\boldsymbol{\eta}_{h_i}^{New} = \boldsymbol{\eta}_{h_i}^{Old} + \Delta\boldsymbol{\eta}_{h_i}$, $\Delta\boldsymbol{\eta}_{h_i} = \epsilon_{Sec} \cdot w_{h_i} \cdot (\boldsymbol{t}_j - \boldsymbol{\eta}_{h_i}) \; \forall \; i, \; (h_{Best}, h_i) \in \mathcal{C}$.
7. Update the firing counter and the training requirement of the first winner unit according to $y_{h_{Best}}^{New} = y_{h_{Best}}^{Old} + 1$ and $w_{h_{Best}} = \frac{1}{y_{Best}+1}$.

8. Age all connections between the first winner and its topological neighbors according to $a^{New}_{h_{Best}h_i} = a^{Old}_{h_{Best}h_i} + 1 \; \forall \; i$ with $(h_{Best}, h_i) \in \mathcal{C}$ and delete those connections that exceed the maximum age, i.e. set $\mathcal{C}^{New} = \mathcal{C}^{Old} \setminus (h_i, h_{i'}) \; \forall \; h_i, h_{i'}$ with $(h_i, h_{i'}) \in \mathcal{C}$ and $a_{h_i h_{i'}} > a_{Max} = |\mathcal{U}| - 1$.
9. Remove those units which do not have a connection to any other unit and only contribute negligibly to the goodness of data representation: $\mathcal{U}^{New} = \mathcal{U}^{Old} \setminus u_{h_i} \; \forall \; h_i$ with $(h_i, h_{i'}) \notin \mathcal{C}$, $h_i \neq h_{i'}$, and $y_{h_i} < |\mathcal{U}|$.
10. After having updated the iteration counter (by $l^{New} = l^{Old} + 1$), check whether the maximum number of iterations L has been exceeded, i.e. if $l^{New} > L$ holds. If so, the algorithm terminates and the connectivity matrix $C = (c_{h_1 h_2})_{h_1, h_2 = 1, \ldots, H = |\mathcal{U}|}$ is generated with $c_{h_1 h_2} = 1$, if $(h_1, h_2) \in \mathcal{C}$, and $c_{h_1 h_2} = 0$, otherwise.
    Otherwise the algorithm continues with step 1.

3 Selected experimental results

3.1 Application to lifestyle analysis

The reliable extraction of meaningful lifestyle patterns from survey data can contribute to a deeper understanding of existing product-customer relationships. Therefore, different approaches, mostly applying traditional cluster analysis, have been discussed in the past to address this topic. The current application is the first one using a growing self-organizing neural network and should provide an impression of the expressiveness of the results being attainable in this way.

The data set to be analyzed was made accessible by the German ZUMA institute and is part of a sub-sample of the 1995 GfK ConsumerScan Household Panel Data (Papastefanou et al. 1999) [6]. The data contains information about socio-economic and demographic characteristics of the considered households (e.g. sociability, openness towards innovations, and enterprizing spirit), attitudes towards nutrition (e.g. German products, ecological food, and plain fare), aspects of daily life (e.g. traditional living and mistrust towards new products), as well as shopping (e.g. price consciousness and information demand). A considerable number of the respective items are more or less concerned with individual nutrition behaviour. The importance of convenience food-related lifestyle determinants when considering food purchasing patterns is emphasized by several market researchers (e.g. Grunert et al. 1997) [7]. To identify existent lifestyle patterns we considered the attitudes of $J = 4\,266$ households measured by means of $K = 81$ Likert-scaled items (with 1 = 'I definitely disagree.' to 5 = 'I definitely agree.'). The scale was assumed to be equidistant, so we could treat the data as metric.

Applying the GNNAPS algorithm to this data with compression level $CL = 0.98$ and a maximum number of iterations $L = 5\ 000\ 000$ leads to a network with $H = 6$ units (prototypes), a quantization error $QE = 37\ 655$, and a value of compactness $Com = 59.72$. For a deeper discussion of both performance measures see e.g. Kohonen 2001 [5] and Marsland et al. 2002 [4]. Larger networks with $H = 16$ (and $QE = 36\ 203$, $Com = 241.97$) or $H = 22$ (and $QE = 35\ 709$, $Com = 489.69$) prototypes correspond to a similar performance regarding the goodness of pattern representation (see QE), but are also characterized by their lesser parsimony/compactness (see Com) regarding the topological structure.

The number of households being represented by the weight vectors $\eta_1 = (\eta_{1,1}, \ldots, \eta_{1,81})$, $\ldots$, $\eta_6 = (\eta_{6,1}, \ldots, \eta_{6,81})$ range between 569 and 907. Each of these 'lifestyle prototypes' can be described by considering its specific attitude profile. Some of the respective attitudes differ more strongly than others between the individual prototypes. From a marketing point of view especially the clearly discriminating characteristics are interesting. Though comprehensive comparisons of the prototypes on the basis of all 81 items are possible, we exemplarily restrict our interpretations for the sake of simplicity to the two largest types of households represented by weight vectors η_2 and η_4. Considering those items which differ mostly across these prototypes leads to the results depicted in Table 1, where low values refer to the disagreement and high value to the agreement with the respective statement. Those item weights which equal the maximum or minimum across all prototypes are marked with the signs + and −, respectively.

Table 1. Comparison of selected 'lifestyle prototypes'

Items (short forms)	η_2 (825)	η_4 (907)
don't like changes	3.83	2.47^-
cooking of known/secure dishes	3.36	1.94^-
traditional orientation	4.11	2.85^-
enthusiasm about foreign delicacies	2.48^-	4.15^+
preferring plain fare	4.27	3.03^-
cooking new, unusual dishes	2.09	3.60^+
too much fuss about nutrition	3.46	2.81
like trying foreign dishes	2.02^-	4.09^+
cooking just well-tried recipes	4.14	2.63^-
getting pleasure from discovering new things	2.17^-	3.99^+

(+/− maximum/minimum item weight across all prototypes)

Prototype or rather weight vector η_2 represents 825 households with a more or less conservative, traditional orientation. Households of this type don't like the new and foreign and they try to avoid changes. The second prototype (η_4) represents more cosmopolitan people, who like to try new and unknown things and who enjoy discovering the world and its eating variety. The weight value 3.83 in the first row of data in the table, for example, indicates the rather affirmative attitude of the respective households towards the item 'I don't like changes in my life, I prefer to keep my habits.'.

Other important items which discriminate between the individual prototypes are, e.g., the relevance of quality in product purchasing decisions, the preference towards brand name products, the time invested in cooking, the attention paid to mild/non-irritant food, the predisposition regarding the purchase of products without additives, the attitude towards environmental pollution, and the preference towards German goods. Consumer profiles of this kind can serve as the basis of target group-oriented marketing activities in food industry and they can support the design of customer designed new products, for instance. Many of our findings are similar to results published by others (e.g. Grunert et al. 1997) [7], which speaks for the adequacy of the new approach for the present area of application.

3.2 Application to media usage analysis

We also applied the GNNAPS algorithm to a very recent and absolutely different data set, which describes the media usage in academic libraries. The data was made accessible by the Bielefeld University Library and was collected between October 2003 and May 2004. It contains information about several book specifications (e.g. year of publication, number of pages and editions), library specific itemizations (e.g. media location and number of copies), lending conditions (e.g. lending periods, prolongations, reservations, and interlending), as well as user characteristics (e.g. gender and user group). To identify interesting patterns of media usage we considered the individual profiles of 3 773 books measured by means of 34 items, where non-metric items, like gender and location, were taken into account by appropriate frequency data.

Applying the GNNAPS algorithm to this data with $CL = 0.93$ and $L = 10\ 000\ 000$ provides $H = 11$ prototypes and performance values $QE = 12\ 804$ and $Com = 95.51$. But similar neural networks with more, e.g. 14 ($QE = 12\ 306$ and $Com = 106.18$), or less prototypes, e.g. 7 ($QE = 13\ 368$ and $Com = 25.93$), uncover interesting patterns as well. In this sense, the neural network which we chose for the following interpretations constitutes a compromise between representation accuracy and parsimony.

The number of books being represented by the weight vectors $\eta_1 = (\eta_{1,1}, \ldots, \eta_{1,34}), \ldots, \eta_{11} = (\eta_{11,1}, \ldots, \eta_{11,34})$ ranges between 2 and 2 655. Three of these prototypes, namely η_7, η_9, and η_{10}, just represent a few books with three, three, and two elements respectively. These books strongly differ from the others, e.g. due to their far above average number of pages and copies and, at least partially, due to their very frequent usage, which altogether points at encyclopedia or dictionaries. More than half of the books considered, namely 2 655, are represented by prototype η_1. This prototype or rather 'book type' is interesting with regard to the aims mentioned above, because it represents a considerable share of the library's book stock. A second group of books, which is represented by weight vector η_2, contains 609 different media, which mainly differ from the biggest group by a four times more frequent lending. Finally, there are three further prototypes representing groups of about 125 to 170 books. Once again we can compare the available prototypes to discuss interesting relations or differences. In Table 2 we exemplarily depicted selected weights of prototype η_1 and η_5. To facilitate interpretations we indicated rounded real values instead of the standardized ones, which have been used for the network's adaptation.

Table 2. Comparison of selected 'book prototypes'

Items (short forms)	η_1 (2 655)	η_5 (126)
number of copies	1.20^-	2.10
number of pages	250^-	375
last edition	1.12	2.25
year of publication	1991	1994^+
number of lendings	0.89	5.25
number of reservations	0.04	0.66
lending period (no. of copies with 14-day lending)	0.13	5.19
number of copies lent during the session	0.57	3.70

(+/− maximum/minimum item weight across all prototypes)

The first prototype obviously represents 2 655 'smaller' and older books, which are mostly available only in the first edition and have been lent less than one time in the relevant period (including all copies) on average. Most of these books are represented by one single copy and are characterized by 30-day rather than 14-day lending periods. Such books are often dissertations or research reports and are typical for the media assortment of an academic library. In contrast to this prototype η_5 represents more recent and 'voluminous' books (regarding the average number of pages) with more than one copy. Most of them are available in a second or higher edition. Such books are lent more than five times during the relevant period of time on average. Their lending period is almost always 14 days and users tend to make reservations for these books and often renew them. Lastly, these books are more frequently used while the university is in session compared to the books represented by prototype η_1. Such media are often textbooks.

For all prototypes the usual time the book lending occurs is between 12 am and 4 pm. The most remarkable differences result from the second and third position shared by the morning time (from 7 am to 12 am) and the evening time (from 4 pm to 8 pm). Obviously, the lending of books in the late afternoon and early evening is unusual, though possible. Altogether, only very few of the books considered are the subject of interlending processes. With respect to the amount of missing/untraceable books no exciting differences could be detected. The same applies to the proportions of the different user groups. To sum up, the available profiles may be used to support future media acquisition decisions, and they enable a more accurate adjustment of library services to the users' needs. In the present case, it seems to be advisable to procure more copies of those books that are represented by prototypes with high numbers of reservations, for instance.

4 Conclusions

The new algorithm proved to be able to adequately represent heterogeneous data patterns with a comparatively small number of parameters to be controlled by the user. A clear advantage results from the fact that no a priori knowledge regarding the presumptive number of prototypes is required. The large autonomy of the algorithm facilitates its practical application to various classification tasks in marketing and elsewhere, e.g. in librarianship, especially by those, who are less familiar with growing self-organizing neural networks. Another benefit of GNNAPS results from the intuitive interpretability of the prototypes as centroids of the represented subsets of data. Unfortunately, the autonomy of GNNAPS burdens its implementation. Further efforts will therefore be devoted to the improvement of the inherent balance between parsimony and autonomy.

Acknowledgment: We would like to express our special thanks to our colleagues at the Bielefeld University Library for providing the media usage data and the Deutsche Forschungsgemeinschaft for supporting parts of the present research within the ProSeBiCA project.

References

1. Martinetz, T., Schulten, K. (1991): A 'Neural Gas' Network Learns Topologies. In: Kohonen, T., Maekisara, K., Simula, O., Kangas, J. (Eds.): Artificial Neural Networks. North Holland, Amsterdam, 397 – 402
2. Fritzke, B. (1995): A Growing Neural Gas Network Learns Topologies. In: Tesauro, G., Touretzky, D. S., Leen T. K. (Eds.): Advances in Neural Information Processing Systems 7. MIT Press, Cambridge, 625 – 632
3. Dittenbach, M., Rauber, A., Merkl, D. (2002): Uncovering Hierarchical Structure in Data Using the Growing Hierarchical Self-Organizing Map. Neurocomputing 48, 199 – 216
4. Marsland, S., Shapiro, J., Nehmzow, U. (2002): A Self-Organising Network that Grows when Required. Neural Networks 15, 1041 – 1058
5. Kohonen, T. (2001): Self-Organizing Maps, 3rd. Edition. Springer, Berlin
6. Papastefanou, G., Schmidt, P., Börsch-Supan, A., Lüdtke, H., Oltersdorf, U. (Eds.) (1999): Social and Economic Research with Consumer Panel Data. Proceedings of the First ZUMA Symposium on Consumer Panel Data, Mannheim
7. Grunert, K. G., Brunsø, K., Bisp, S. (1997): Food-Related Lifestyle: Development of a Cross-Culturally Valid Instrument for Market Surveillance. In: Kahle, L., Chiagouris, L. (Eds.): Values, Lifestyles and Psychographics. Erlbaum, Mahwah, 337 – 354

The Quality of Prior Information Structure in Business Planning
- An Experiment in Environmental Scanning -

Sören W. Scholz, Ralf Wagner

Business Administration and Marketing, University of Bielefeld,
Universitätstr. 25, 33615 Bielefeld, Germany
email: sscholz@wiwi.uni-bielefeld.de, rwagner@wiwi.uni-bielefeld.de

Abstract. Increasing attention has been devoted in recent years to the firm's ability to adapt its marketing strategies to a rapidly changing environment. Given that the abundance of news, reports, and announcements found in new electronic environments such as the WWW hampers an extensive manual search, computer-based systems have become important supportive tools for business planning purposes. Several studies investigate the impact of managerial traits on this question, however the potential influence of an inadequate information structure in automatic information-seeking tools is rarely addressed.

In this paper, we examine the effect of the quality of the information structure in automated information-seeking tasks. We use a prototypic system that aims to detect and to evaluate relevant information about financial markets, and systematically contaminate the information structure by index terms referring to an adjacent but different task. Empirical evidence from an experimental evaluation of documents from the Reuters text collection substantiates the relevance of the prior information structure to the automated information search.

1 Introduction

Environmental scanning (ES) refers to the processes of scanning the business environment in order to detect changes crucial to future business success. Walters et al. (2003) argue that ES is the inevitable first step in business planning. One major challenge in business planning stems from the rather unstructured and complex business environment, consisting of a wide range of areas that may affect the business' success (Xu et al. 2003). In addition to the demanding task of identifying relevant issues from the steady stream of information with which the decision maker is faced, he must also be capable of comprehending and interpreting the meaning of this information. Regerer and Palmer (1996) as well as more recently Rouse (2002) show that naturally this process is strongly based on prior knowledge and experience. The prior domain knowledge in information retrieval – or in other words the *information structure* – has a crucial impact on information-seeking processes in general, and particularly, on the identification of relevant information (Baeza-Yates and Ribeiro-Neto 1999). While several previous studies refer to the *quality* and *efficiency* of algorithms (see e.g., Yang and Liu (1999) for a comparison of topic detection and tracking algorithms), which provide a rating or classification of documents, a systematic consideration of the impact of the a priori knowledge used in the context of information-seeking tasks has rarely been a matter of subject. This issue seems even more significant when considering that more than 75 % of organizations use information technology to aid their gathering, analyzing, and reporting efforts in ES (CIO-Insight 2003). This study aims to scrutinize the impact of the prior information structure

- on the choice of individual information sources,
- on the evaluation of information sources, and
- on the number of chosen information sources.

The selection process of our prototypic system for information selection is built upon information foraging theory (IFT), which is a promising approach in modeling human information search and selection processes in ES.

The remainder of this paper is structured as follows: First, we introduce a model for automated information selection on the WWW that roots in the *heuristics* of IFT. Subsequently, we outline an experiment for the analysis of the impact of information structure on information-seeking tasks through a performance test in the context of a typical ES task. The paper concludes with a summary of our empirical investigations and their implications.

2 The Model

2.1 Basic Principles of Information Foraging Theory

IFT is a cognitive driven approach that models information-seeking behavior of human individuals. Central to its thesis is the idea that the search for information is an expatriation of food foraging mechanisms. Pirolli and Card (1999) adapted the theory of optimal foraging, which seeks to explain the adaptations of organism structure and behavior to the environmental constraints of foraging for food. The theory is based on the hypothesis that, when possible, natural information systems evolve toward stable states that maximize the gain of valuable information per unit cost. IFT, like optimal foraging theory, is made up of three components (Stephens and Krebs 1986):

(a) *Decision component:* How to find an optimal choice for the forager's problem?
(b) *Currency component:* How are various choices to be evaluated?
(c) *Constraint component:* What limits the decision forager's feasible choices, and what limits the pay-off (currency) that may be obtained?

Taking into account these components our model is based on the subsequent assumptions:

(i) *Exclusivity of search and exploitation:* The forager can only do one action at a time. He must choose between searching for and consuming of sources.

(ii) *Sequential Poisson encounters:* The sources appear randomly one at a time. The encounter of sources can be modeled using a Poisson distribution, thus the probability for the encounter of sources is constant.

(iii) *Incomplete information:* The forager is only able to determine selected aspects of the information environment (e.g., the density of sources).

IFT is best described with the help of the term "lost opportunity" based on assumption (i). Due to the generally limited information-processing capacities of human beings, the forager has to cope with two basic decision problems. First, he cannot access all information sources, but rather has to select the most valuable sources of information he is able to handle in a given time. Analogous to optimal foraging theory, Pirolli and Card (1999) call this selection the *information diet.* Choosing this diet results in an optimization problem. Second, the forager has to assess which information sources he wants to access. In contrast to conventional foraging theory, we use imperfect information (assumption (iii)) to decide which information sources should be selected. The foragers rely on proximal cues to assess profitability and the prevalence of information sources and their representations. In line with Pirolli and Card (1999), we refer to this imperfect perception of the value, cost or access path of information as *information scent.*

We adapt IFT for the task of ES on the WWW (Decker et al. 2004). Because each new web page is just a "mouse click" away, we adjust our model foregoing the prey model of IFT (Stephens and Krebs 1986). We consider each web page as a unique information source for two reasons: On the one hand, the information forager is looking for basically 'weak' information predicting future changes in the business environment. This information is provided from various sources presenting information that do not necessarily have much in common. On the other hand, search engines offer the possibility to cluster sites dynamically so that there is no static or objective structure on which the forager can rely.

2.2 Construction of an Optimal Diet

The information forager aims to maximize of the rate of gain of valuable information per unit cost (Pirolli and Card 1999):

$$\max R = \frac{G}{T_B + T_W} \tag{1}$$

with:

G ...total net amount of valuable information gained

T_B ...total time spent searching information sources

T_W...total time spent extracting and handling information sources

Given a set of $\mathcal{D}$ information sources d_i (hereafter, documents) in the information environment the average rate of gain of information per time unit cost is:

$$R(\mathcal{D}) = \frac{\sum_{i=1}^{|\mathcal{D}|} \lambda \cdot g_i(t_{Wi})}{1 + \sum_{i=1}^{|\mathcal{D}|} \lambda \cdot t_{Wi}} \tag{2}$$

In equation 2 the term $g_i(t_{Wi})$ denotes the rate of gain of document d_i extracted in time t_{Wi} while λ is the reciprocal value of the time t_B needed to encounter a document in the given information environment (with $\lambda = 1/t_B$). The profitability π_i of each document d_i takes the limited resources of the forager into account:

$$\pi_i = \frac{g_i(t_{Wi})}{t_{Wi}} \qquad \forall \quad i \tag{3}$$

The rate-maximizing subset of information sources is selected in two steps:

1. Rank the documents by their information profitability. For simplicity of presentation, let the index be ordered such that: $\pi_i > \pi_{i+1} \quad \forall i$.
2. Add documents to the diet until the rate of gain of information for a diet of the top l (defining the set $\mathcal{L} = \{d_i | i \in \mathcal{D} \wedge \pi_i > \pi_{l+1}\}$) documents is greater than the profitability of the $l+1$ document,

$$R(\mathcal{L}) = \frac{\sum_{i=1}^{|\mathcal{L}|} \lambda \cdot g_i(t_{Wi})}{1 + \sum_{i=1}^{|\mathcal{L}|} \lambda \cdot (t_{Wi})} > \pi_{l+1}. \tag{4}$$

2.3 Assessment of Information Gain

Referring to assumption (iii), the value of information is assessed with the help of proximal cues instead of reading and comprehending the given document. Thus, the meaning potential of a language expression resembles the *information scent* of the document. Each document d_i is represented by a N-dimensional vector, where each dimension captures a single concept of the information structure:

$$\boldsymbol{w}_i = (w_{i1}, w_{i2}, \ldots, w_{iN}) \,\forall\, d_i \in \mathcal{D} \tag{5}$$

The frequency w_{ik} of usage of a language expression v_n $(n = 1, \ldots, N)$ in document d_i is adjusted by the classical term frequency-inverse document frequency (TF-IDF) approach (Baeza-Yates and Ribeiro-Neto 1999). The dimensionality N of the vector is given by vocabulary $\mathcal{V}$, with $|\mathcal{V}| = N$, which represents the mental model of the forager. Each term $v_n \in \mathcal{V}$ describes a concept (e.g. words, formulas, or symbols) conceivable by the forager.

3 The Experiment

Agents (both human managers and automata) already familiar with a given domain are more cable of electing the important concepts in that domain. However, in ES for business planning, the results from several domains (e.g., new products in a market, new production technology, changes in the legal environment or consumer preferences) need to be combined and interrelations blur the borders of adjacent domains. Therefore, the accurate discrimination of areas relevant to certain up-and-coming domains has become a cutting the edge frontier in business planning and particularly, in marketing planning

(Nitze et al. 2003). The information environment in the experiment is made up of 50 documents from the Reuters categorization test collection (Lewis 1997). Some of the documents contain information describing developments in financial markets relevant for business planning. Moreover, the domains of advertising of trade fairs, stock exchanges and activities of different national banks, currency exchange rates, corporate lawsuits etc. are also considered in the text basis. In an earlier paper (Decker et al. 2004), we demonstrated the ability to identify the most relevant documents describing general developments in financial markets (hereafter, domain $\mathcal{A}$) using an adequate prior information structure. This is given by a vocabulary $\overline{\mathcal{V}}$ of $\overline{N} = 129$ concepts v_n (e.g., bull market, credit quality, and rating agencies) invariably relevant to this domain $\mathcal{A}$.

In this extension of the previous study, we have added the second vocabulary $\overline{\overline{\mathcal{V}}}$ to the information structure. The concepts (e.g., conversion price, tender offer, or bond issue) of vocabulary $\overline{\overline{\mathcal{V}}}$ refer to the identification of singular investment options (e.g. new tender offers on stock markets). It contains $\overline{\overline{N}} = 86$ language expressions v_n that we elicited in exactly the same manner and represents the information structure of domain $\mathcal{B}$. In order to verify the implicit underlying assumption that concepts of domain $\mathcal{B}$ interfere with the assessment process of the system, we compared an expert appraisal of documents relevant in domain $\mathcal{A}$ with documents considered to be valuable in domain $\mathcal{B}$. The correlation is negative ($r = -0.43$). With respect to this result, we systematically contaminated the given vocabulary of domain $\mathcal{A}$ with varying cardinality of $\overline{\overline{\mathcal{V}}}$.

To analyze the influence of impure information structures and their impact on the assessment and selection processes modeled by IFT, we use the experimental design briefly outlined in Figure 1. As obvious from the figure,

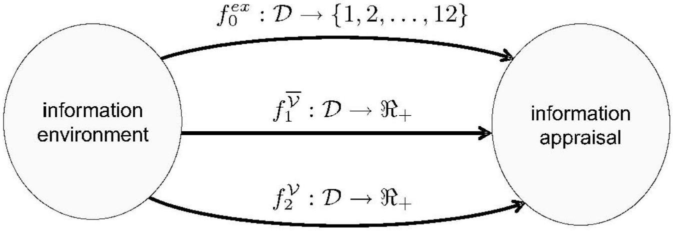

$f_0^{ex} : \mathcal{D} \rightarrow \{1, 2, \ldots, 12\}$
information environment
$f_1^{\overline{\mathcal{V}}} : \mathcal{D} \rightarrow \Re_+$
information appraisal
$f_2^{\mathcal{V}} : \mathcal{D} \rightarrow \Re_+$

Fig. 1. Experimental information assessment mapping functions

we use three different mapping functions to assess the efficiency (given by the ratio of valuable information per time length) of the documents d_i available in the information environment $\mathcal{D}$. The mapping function f_0^{ex} is a qualitative appraisal of the efficiency of documents by three experts who read each document in its entirety, and assessed the rentability of each document with regard to the given task on a 12-point scale employing the Delphi-method. The mapping function $f_1^{\overline{\mathcal{V}}}$ indicates the quantitative IFT-based selection process, which is outlined in equation 3 employing the 'pure' vocabulary $\overline{\mathcal{V}}$. The mapping function of the contaminated system is given by $f_2^{\mathcal{V}}$, where $\mathcal{V}$ consists of information structure $\overline{\mathcal{V}}$ of domain $\mathcal{A}$ and a subset of information structure $\overline{\overline{\mathcal{V}}}$ of domain $\mathcal{B}$. Using this simple and intuitive design, we want to explore the impact on the three components of our model.

- How is the *assessment* of documents influenced by the contamination of the information structure (currency component)?
- How is the quality (which documents are chosen) of the *selection process* influenced by the pollution of the information structure (decision component)?
- How many documents are actually selected due to the contamination of the information structure (constraint component)?

For the evaluation of the overall assessment quality of our two-stage selection algorithm, we compared the experts' efficiency assessment given by the mapping f_0^{ex} to the systems appraisal computed by the functions $f_1^{\overline{\mathcal{V}}}$ and $f_2^{\mathcal{V}}$ with regard to different degrees of contamination. The results, illustrated in figure 2(a), show that the degree of impurity has a linear impact on the accuracy of the model-based appraisal of the efficiency.

This measure also takes into account a differentiation of irrelevant documents that is not essential for the quality of the detection and selection processes modeled by IFT. One of the most characteristic features of the information diet calculation in IFT is the fact that the selection is largely based on the prevalence of relevant documents. How many and which documents are actually selected depends mainly on the distribution of the assessment of the top-ranked documents, as well as on the distribution of relevant documents in the information environment (given by λ) and the handling time t_{Wi} (see equation 4). The stop criterion of the associated algorithm includes only the top-ranked documents. This leads to two conclusions: First, the optimal number of selected documents cannot be computed a priori, without knowing the information environment, and second, the results illustrated in figure 2(a) do not necessarily have to influence the quality of the selection process.

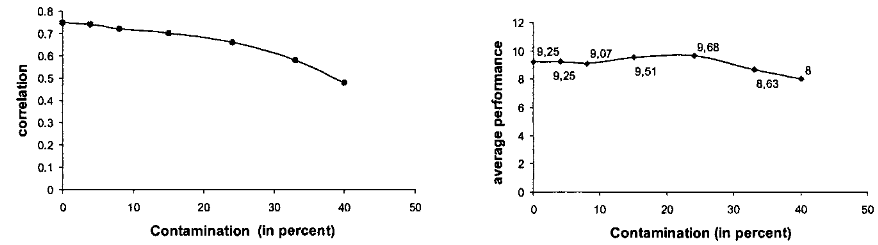
We use the following formula to measure the performance P of the IFT-based selection algorithm with regard to imbalances due to unequal number of documents in the selections.

$$P = \frac{1}{|\mathcal{S}|} \sum_{d_i \in \mathcal{S}} \tilde{\pi}_i \tag{6}$$

Where $\mathcal{S}$ is the set of selected documents and $\tilde{\pi}_i$ denotes the documents' *efficiency* as evaluated by the experts. Using the 'pure' information structure

$\overline{V}$, the scanning system selects 4 documents. Its performance equals 9.25 while the optimal selection of 4 documents based on the distribution of the efficiency of all 50 documents equals 10.75 points.

Figure 2(b) shows the influence of different states of the information structure's impurity on the selection. The abscissa indexes the degree of impurity in percentages, while the ordinate displays the performance of the selection given by this state of contamination.



(a) Appraisal of documents

(b) Selection process

Fig. 2. Influence of the information structure's impurity on quality of the appraisal and selection process

These results imply that the effect of impurity is rather small on the accuracy of the appraisal of the documents' relevance. Moreover, the number of selected documents is rather stable for varying contamination degrees and leads only to a difference of ±1 document.

4 Summary and Conclusions

An ubiquitous problem in optimizing managerial activities in business planning is the limited time resources that do not allow for an exhaustive evaluation of the business environment. We challenge this problem with a model of automated ES that is based on three components. In order to optimize the search processes in ES (decision component) the cognitive as well as time restrictions of the manager are taken into account (constraint component). This leads to the use of opportunity costs in terms of a trade-off between searching vs. exploiting of information sources (currency component). We outlined a two-stage heuristic to reveal an optimal information diet provided by automata to the human managers. The heuristic optimization considers the efforts to search for further information as well as the time needed for reading and comprehending information sources.

The fundamental result of the experimental study is the robustness of the IFT-based selection algorithm against a contamination of the information structure. All three components vary only to a very reasonable degree due to the contamination. Even when one-third of the information structure consists of language terms from an adjacent but different domain, the system obtains a high performance that is only one point less than its performance using a pure information structure. This result turns out to be even more meaningful considering the fact that search formulation is referred to as the most complex phase in information seeking (Large et al. 2001).

We show that the use of IFT-based selection processes leads to expedient, relevant selections of documents - regardless of the existence of an information structure that is, to some degree, imperfect or impure. Since managers do not necessarily possess expertise in all relevant areas of the business environment, the use of the model presented herein is a promising approach in the refinement of the managerial search strategies crucial in business planning.

References

1. Baeza-Yates, R., Ribeiro-Neto B. (1999) Modern Information Retrieval. Addison Wesley, Boston
2. CIO-Insight (2003) Research: Business Intelligence 2003 - Are Your BI Systems Making You Smarter.
   URL=http://www.cioinsight.com/article2/0,3959,1098923,00.asp
3. Decker R., Wagner R., Scholz, S. (2004) Environmental Scanning in Marketing Planning - An Internet-Based Approach. Discussion Paper No. 516, University of Bielefeld, March 2004
4. Large A., Tedd L.A., Hartley R.J. (2001) Information Seeking in the Online Age: Principles and Practice. Saur, München
5. Lewis D.D. (1997) The Reuters-21578 Text Categorization Test Collection. URL=https://www.research.att.com/ lewis/reuters21578.html
6. Nitze P.S., Parker K.R., Dishman P.L. (2003) Multi-class Interest Profiles: Applications in the Intelligence Process. Marketing Intelligence & Planning, 21, 263–271
7. Pirolli P., Card S. (1999) Information Foraging. Psychological Review, 106, 643–675
8. Reger R., Palmer T.B. (1996) Managerial Categorization of Competitors: Using Old Maps to Navigate New Environments. Organizational Science, 7, 22–39
9. Rouse, W.B. (2002) Need to Know - Information, Knowledge, and Decision Making. IEEE Transactions on Systems, Man, and Cybernetics, 32 , 282–292
10. Stephens D.W., Krebs J.R. (1986) Foraging Theory. Princeton University Press, Princeton
11. Walters B.A., Jiang J.J., Klein G. (2003) Strategic Information and Strategic Decision Making: the EIS/CEO Interface in Smaller Manufacturing Companies. Information & Management, 40, 487–495
12. Xu X.M., Kaye G.R., Duan Y. (2003) UK Executives' Vision on Business Environment for Information Scanning – a Cross Industry Study. Information & Management, 40, 381–389
13. Yang Y., Liu X. (1999) A Re-examination of Text Categorization Methods. Proceedings of 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, Berkley, 42–49

Product Line Optimization as a Two Stage Problem

Bernd Stauß and Wolfgang Gaul

Institut für Entscheidungstheorie und Unternehmensforschung
Universität Karlsruhe, P.O. Box 6980, 76128 Karlsruhe, Germany
{bernd.stauss, wolfgang.gaul}@wiwi.uni-karlsruhe.de

Abstract. Optimizing product lines is the task of planning the joint offer of multiple substitutes concurrently. Actually, this resembles the design of a choice menu where consumers are supposed to choose at most one item from a set of alternatives.

In this paper we model the product line optimization problem by decomposing it into two stages. Product decision is done in the first stage by anticipating possible outcomes of the subsequent (price) stage. This so-called hierarchical decision situation is considered due to uncertainty about consumers' tastes. Results suggest that substantial increase of expected profit can be drawn from the hierarchical approach.

1 Basics in Product Line Optimization

1.1 Background

Making decisions in an uncertain environment can be managed by two basically different strategies depending on the decider's option for possibly postponing parts of the respective decision. The first strategy consists in just waiting and deciding after realizations of the decision environment become known, i.e., *wait-and-see.* The second one is characterized by the fact that there is no time left for delaying decision making, i.e. decisions have to be taken *here-and-now* (e.g. [9]). Usually, decisions have strategical relevance, constitute some kind of commitment and, thus, cannot be changed for the short term. However, minor adjustments - frequently referred to as recourse actions - may be allowed after realizations of the decision environment are revealed.

Situations, similar to the latter, are typically found in new product development (NPD) processes. Cooper and Kleinschmidt (e.g. [6]) identified that an early and sharp definition of product concepts is one of the most important success factors in the NPD process. An early definition disciplines the NPD process by fostering the timely start of subsequent phases in product development as well as by making the whole process more stable, i.e., insusceptible to changing input information [3]. However, it should be noticed that the design and selection of concepts constitutes a very critical part of NPD processes, since they are extremely vulnerable to any miss-specification in this early phase ([15], pp. 40). This implies that there might occur two major difficulties that complicate the planning process. First, in dynamic markets consumer preference data used in the product definition and concept selection phase may become obsolete when market introduction follows with considerable delay of time ([5]). Second, consumers may find it difficult to state their preferences for widely unknown products ([16]). So, firms are expectetd to determine their spectrum of products to be offered by accepting shortcomings of information about consumer tastes and potential trends that may evolve within the time to market. Milliken [12] calls this "uncertainty concerning external factors". However, uncertainty about consumers' tastes diminishes while approaching the point where production starts. Since product policy cannot be changed in short-term due to commitments in early phases in the NPD process there just remains adaption by means of tactical marketing methods, such as, e.g., pricing. The structure of the described problem is thus similar to those of hierarchical decision situations ([10]) that may be found in various situations some of which are electrical power management (e.g., [8]), scheduling (e.g. [4]), (vehicle) routing ([11]), or revenue management ([2]).

1.2 Notation and Constraints

We first introduce some basic notation in product line optimization on which the hierarchical model in the next section is based. Suppose that there is a set of consumer segments $\mathcal{I}$ which are of size N_i ($i \in \mathcal{I}$) and which are interested in a firm's new product line described by a set of potential products $\mathcal{L}$. Usually, some kind of utility measure r_{il} is considered to quantify how some segment $i \in \mathcal{I}$ values some product $l \in \mathcal{L}$. We suppose that the outcomes of r_{il} are measured on a monetary scale. In accordance with literature, we refer to this measure as reservation price.

Furthermore, to state the product line optimization problem more formally the following binary decision variables are needed, where

$$\theta_{il} = \begin{cases} 1, & \text{if segment } i \text{ selects product } l, \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

and

$$y_l = \begin{cases} 1, & \text{if product concept } l \text{ is selected by the firm,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Despite this, we explicitly allow the setting of prices p_l for the products. Additionally, potential fixed costs F_l, that may occur in connection with the development of new products $l \in \mathcal{L}$, are considered, and - of course - variable costs c_l arise if a product l has to be produced. Last of all, we have to specify consumer choice behavior that constitutes a crucial part in choice-based product line designs. A quite often used mechanism is the first-choice rule where a segment i chooses the product $l^*(i)$ in the line that consumers of the segment like best. If we follow the general modeling the most favoured product maximizes consumer's surplus, i.e., the difference $r_{il} - p_l$. Thus, consumer choice can be defined by the constraints

$$r_{il^*(i)} - p_{l^*(i)} \geq r_{il} - p_l \qquad i \in \mathcal{I},\ l \in \mathcal{L} \tag{3}$$

$$r_{il^*(i)} - p_{l^*(i)} \geq 0 \qquad i \in \mathcal{I} \tag{4}$$

These constraints are frequently referred to as incentive constraints (3) and participation constraints (4), respectively.

2 Pricing of Existing Product Lines

Although a hierarchical problem structure is suggested where concept selection and market introduction (pricing) has to be done at different points of time while anticipating some uncertainty, we will first focus on the pricing problem where product lines have already been determined and which can be stated as follows:

$$\mathcal{Q}(y): \quad \sum_{i\in\mathcal{I}} \sum_{l\in\mathcal{L}} \theta_{il} N_i \left(p_l - c_l\right) \to \max \tag{5}$$

w.r.t

$$\sum_{\tilde{l}\in\mathcal{L}} \theta_{i\tilde{l}} \left(r_{i\tilde{l}} - p_{\tilde{l}}\right) \geq r_{il} - p_l \qquad i \in \mathcal{I},\ l \in \mathcal{L} \tag{6}$$

$$\sum_{\tilde{l}\in\mathcal{L}} \theta_{i\tilde{l}} \left(r_{i\tilde{l}} - p_{\tilde{l}}\right) \geq 0 \qquad i \in \mathcal{I} \tag{7}$$

$$\sum_{i\in\mathcal{I}} \theta_{il} \leq M y_l \qquad l \in \mathcal{L} \tag{8}$$

$$\sum_{l\in\mathcal{L}} \theta_{il} \leq 1 \qquad i \in \mathcal{I} \tag{9}$$

$$p_l \geq 0,\ \theta_{il} \in \{0,1\} \qquad i \in \mathcal{I},\ l \in \mathcal{L} \tag{10}$$

Restrictions (6) are the the already mentioned incentive constraints that force the assignment of each segment to its most favored product in terms of consumer surplus that can be achieved. However, consumers will only select products iff they get a non-negative surplus, which is the interpretation of the participation constraints (7).[1] Furthermore, products can only be chosen if the respective concept was selected in the prior decision phase due to restriction (8). As shown by Spence ([14]) and subsequently taken up by Dobson, Kalish ([7]) the problem allows for a simple decomposition which

[1] Even in the case of ambiguity it is assumed that the respective consumers select at most one product. A common practice in quantitative product line optimization is to suppose that if the outcome of the choice rule lacks uniqueness, consumers choose that product which leads to maximal profit for the firm.

is sketched in the following. Suppose, that a feasible assigment subject to (8), (9), and $\theta_{il} \in \{0,1\}$ ($i \in \mathcal{I}$, $l \in \mathcal{L}$) is given by $(\bar{\theta}_{il})_{i\in\mathcal{I},l\in\mathcal{L}}$. Let again $l^*(i)$ be the index of the product that is assigned to segment i, i.e., for which $\bar{\theta}_{il^*(i)} = 1$ holds. For ease of notation we introduce the dummy choice option $l = 0$, where $r_{i0} \equiv p_0 \equiv c_0 \equiv 0$, and $\bar{\theta}_{i0} = 1 - \sum_{l\in\mathcal{L}} \bar{\theta}_{il}$, to allow explicitly for choosing none of the products in the line. So $\mathcal{Q}(y)$ of equation (5) changes to

$$\tilde{\mathcal{Q}}(y,\bar{\theta}): \quad \sum_{i\in\mathcal{I}}\sum_{l\in\mathcal{L}} \bar{\theta}_{il} N_i (p_l - c_l) \rightarrow \max \tag{11}$$

w.r.t

$$r_{il^*(i)} - p_{l^*(i)} \geq r_{il} - p_l \qquad i \in \mathcal{I},\ l \in \mathcal{L} \cup \{0\} \tag{12}$$

$$p_l \geq 0 \qquad l \in \mathcal{L} \tag{13}$$

Restrictions (12) can be reformulated as $p_{l^*(i)} - p_l \leq r_{il^*(i)} - r_{il}$ for $i \in \mathcal{I}$, $l \in \mathcal{L} \cup \{0\}$. Note that (12) combines both types of incentive and participation constraints, since we denoted $l = 0$ as no-purchase option. Let $\mathcal{A}$ be the set of products that were chosen at least once ($\mathcal{A} = \{l \in \mathcal{L} | \sum_{i\in\mathcal{I}} \bar{\theta}_{il} \geq 1\}$). Since for any $k \in \mathcal{A}$, at most one of the restrictions (12) for $l^*(i) = k$ is binding, the following modification of (12) is possible:

$$p_k - p_l \leq \min_{i\in\mathcal{I},k=l^*(i)} \left\{ r_{il^*(i)} - r_{il} \right\} = a_{lk} \quad k \in \mathcal{A},\ l \in \mathcal{L} \cup \{0\} \tag{14}$$

If we relax the non-negativity constraints (13) the dual of (11) w.r.t. (14) is given by

$$\sum_{l\in\mathcal{L}\cup\{0\}} \sum_{k\in\mathcal{A}} a_{lk} x_{lk} \rightarrow \min \tag{15}$$

w.r.t.

$$\sum_{l\in\mathcal{L}\cup\{0\},\ l\neq k} x_{lk} - \sum_{l\in\mathcal{A},\ l\neq k} x_{kl} = \sum_{i\in\mathcal{I}} \bar{\theta}_{ik} N_i \qquad k \in \mathcal{A} \tag{16}$$

$$-\sum_{k\in\mathcal{A}} x_{lk} = 0 \qquad l \in \mathcal{L} \setminus \mathcal{A} \tag{17}$$

$$x_{lk} \geq 0 \quad l \in \mathcal{L} \cup \{0\},\ k \in \mathcal{A} \tag{18}$$

This problem can easily be transformed into a minimal cost flow problem for which a pool of efficient algorithms is available.

3 Two Stage Product Line Optimization

As indicated in the beginning, the hierarchical planning situation can be seen as a process consisting of two stages where in the first stage a firm has to check concepts and decide for every product $l \in \mathcal{L}$ whether or not to take it up into the line (i.e., $y_l = 1$ or $y_l = 0$) while facing uncertainty with regard to consumers' reservation prices. Uncertainty is operationalized by a matrix $(r_{il}(\omega))_{i\in\mathcal{I},l\in\mathcal{L}}$, where ω is a realization drawn from the set Ω of realizations under consideration. Departing from the definitions and assumptions of the preceding section a formal notation of the decision problem can be given by means of a so-called deterministic equivalent problem. Let $\mathcal{Q}(\omega, y)$ define - by analogy to (5)-(10) - the second stage profit, that can be earned by a product line determined by y and the realization ω of the reservation price matrix. Then the overall problem - referred to as master problem in the remainder of this paper - can be modeled as one of finding a product line y that maximizes expected profit $E_\omega[\mathcal{Q}(\omega, y)]$ minus investment from the first stage:[2]

$$\mathcal{M}(y) = -\sum_{l\in\mathcal{L}} F_l y_l + E_\omega\left[\mathcal{Q}(\omega, y)\right] \to \max, \quad y_l \in \{0, 1\} \tag{19}$$

It should be noted that due to the prevalent binaries the relaxed expected recourse function $E_\omega[\mathcal{Q}(\omega, y)]$ will be discontinuous and in general not concave in y ([13]), so classical decomposition by stages known from two stage stochastic linear programming is hampered. Thus, we propose a heuristic decomposition by stages approach which is explained below.

We first sketch the idea of the optimization algorithm for the pricing problem by giving a feasible assignment for this problem. In this feasible consumer-product assignment each segment $i \in \mathcal{I}$ is assigned to the product $l^*(i) = \text{argmax}_{l\in\mathcal{L}\cup\{0\}}\{r_{il}(\omega) - c_l\}$ and $r_{i0}, c_0 \equiv 0$ $(i \in \mathcal{I})$ because a lower bound for prices should be given by marginal costs, thus, firms may accept prices $p_l := c_l$ as a lower bound. Such an initial solution seems to be a quite good starting point since prices are at lower bounds and may be increased successively. This is done by determining minimal cost flows and reassigning consumers and products iteratively. I.e., a price increase is done in the following way: segments which "block" a price increment for the assigned product are temporarily reassigned to their predecessors along the minimal cost flow path. If profit decreases, the reassignment is reversed.

In order to generate promising candidates for the new product line that have to be considered as given in the successive stage, the pricing problem can be embedded in an overall Simulated Annealing procedure. This kind of search procedure tries to improve the objective value by passing the solution space in a random manner, thereby accepting a worsening of the objective value with a declining probability. The latter modification of local search minimizes the threat of being caught in local optima. The generation of candidate solutions (the so-called neighborhood solutions) constitutes a critical part of the algorithm. We implemented this in a straightforward manner by selecting randomly one element l^* of y and inverting its value (i.e., in each

[2] There are, of course, alternative criteria that are sometimes used instead of the mean-value approach (s., e.g., Schultz [13]).

	Scenario 1: $r_{il}(1)$							Scenario 2: $r_{il}(2)$							Scenario 3: $r_{il}(3)$						
segment i	products																				
	$l=1$	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
1	5	3	4	1	1	2	4	4	6	10	4	3	4	5	2	5	7	3	6	2	5
2	12	3	3	9	6	18	17	2	14	2	14	16	28	10	2	13	9	4	6	8	10
3	3	12	6	9	12	8	15	3	7	7	6	9	12	15	3	6	9	12	15	3	10
4	6	12	6	3	15	23	5	20	7	8	9	12	13	23	6	4	19	9	15	13	9
F_l	10	22	13	50	10	23	40	10	22	13	50	10	23	40	10	22	13	50	10	23	40

Table 1. Data set for computational testing

iteration ν we set $y_{l*}^{\nu} := 1 - y_{l*}^{\nu-1}$). Though this kind of generating neighborhood solutions is quite simple, it guarantees that each feasible solution is selected with a positive probability. Furthermore, we are in a very pleasant situation, since there were no constraints in the master problem with respect to y_l, so that every neighborhood solution that has randomly been generated as described is feasible (which otherwise would have to be checked). In each iteration step one has to decide whether or not the neighborhood solution y^ν is accepted. If $\mathcal{M}(y^\nu)$ has increased, the new solution is accepted. Otherwise acceptance occurs with decreasing probability (i.e., the probability of a continuous random variable $\tilde{\alpha}$ with support in $[0,1]$ that has to be less than or equal to $\exp\left\{\frac{\mathcal{M}(y^\nu)-\mathcal{M}(y^{\nu-1})}{T}\right\}$ where T is a parameter that is referred to as temperature and successively decreased by a factor $\gamma \in (0,1]$;[3] the procedure stops when temperature falls below a threshold value).

4 Value of the Hierarchical Approach

In order to report on some computational experience with the model, the algorithm, and the value of the hierarchical approach as well, a number of runs with simulated data were performed. Because of page restrictions for this paper, we just consider three scenarios, i.e. $\omega \in \{1, \ldots, 3\}$ with respective probabilities $(\pi_1, \ldots, \pi_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, four segments that are assumed to have equal size ($N_i = N = 10$), and seven products. Reservation prices and fixed costs are given in Table 1. We further assume that there are no variable costs, which is equivalent to the assumption that reservation prices are cost adjusted. Let $y(\bar{\omega})$ be the solution of (19) where all random variables have been replaced by their means. This modified problem is sometimes refered to as expected value (EV) problem. $\mathcal{M}(y(\bar{\omega}))$ describes the expected result using EV (EEV). We found that EEV was about 397.33 $(= \frac{1}{3} \times 500 + \frac{1}{3} \times 490 + \frac{1}{3} \times 340 - 13 - 10 - 23 = E_\omega[\mathcal{Q}(\omega, y(\bar{\omega}))] - \sum_{l \in \mathcal{L}} F_l y_l(\bar{\omega}))$ with $y(\bar{\omega}) = (0,0,1,0,1,1,0)$ as the product line to be selected. However, the heuristic profit for the hierarchical model (HHP), i.e. the objective of the heuristically solved hierarchical problem, accounted for 458.5 that was about

[3] A constant temperature approach, i.e., $\gamma = 1$, is proposed by [1] in order to approximate the distribution of a stochastic objective function.

corr. of reservation prices across segments

	$l=3$	$l=5$	$l=6$
$corr(r_{i7}(\bar{\omega}),r_{il}(\bar{\omega}))$	0.12	0.91	0.70
$E_\omega[corr(r_{i7}(\omega),r_{il}(\omega))]$	0.02	0.34	0.23

(a)

effect from altering fixed costs

	line configuration y_l					VSS (VSS/EEV)
fixed cost	$l=1$	3	5	6	7	
$1 \times F_l$		1 (1)	1 (1)	1 (1)	1 (0)	61.17 (15.4%)
$2 \times F_l$		1 (1)	0 (1)	1 (1)	1 (0)	24.67 (7.0%)
$5 \times F_l$	1 (0)	1 (1)	1 (1)			16.14 (8.0%)

hierarchical (average)

(b)

Table 2. Structure of correlation and line configuration

15% above the EEV[4] with $y = (0,0,1,0,1,1,1)$ as optimal product line. A thorough analysis was performed to motivate the addition of product 7 to the hierarchical solution. A closer look at the correlations of reservation prices across segments (s. Table 2(a)) reveals that the mean correlation of scenario specific reservation prices indicates that product 7 is rather decoupled from the others (second row in Table 2(a)). This is not true, if we consider a situation where reservation prices are averaged in advance which is indicated by the first row in Table 2(a). We therefore conclude, that the introduction of product 7 is due to its ability to further segment the market. Obviously, the hierarchical approach provides a more comprehensive product line since fixed costs are quite low. Thus, we conducted sensitivity analysis by means of increasing F_l. As indicated by table 2(b) and as we would have expected, overall variety will be reduced, when fixed costs become dominant. However, the relative value of the hierarchical approach remains substantial.

5 Conclusions

We presented a new view on product line design in an uncertain environment. The framework of our model is given by former deterministic approaches where we built especially on the work of Spence [14]. Since exact decomposition algorithms suffer from the structure of the second stage problem a hybrid heuristic approach is proposed. Our results indicate (although for a small example) that the value of the hierarchical approach is substantial, since uncertainty is anticipated by selecting more comprehensive product lines in order to be more flexible whatever future will look like.

[4] The absolute value of the stochastic solution (VSS=HHP-EEV) was about 61.17.

References

1. Alrefaei, M. H., Andraottir, S. (1999) A Simulated Annealing Algorithm with Constant Temperature for Discrete Stochastic Optimization. Management Science **45**(5), 748–764
2. Bertsimas, D., Shioda, R. (2003) Restaurant Revenue Management. Operations Research **51**(3), 472–486
3. Bhattacharya, S., Krishnan, V. (1998) Managing New Product Definition in Highly Dynamic Environments. Management Science **44**(11), 51–64
4. Birge, J. R., Dempster, M. A. H. (1996) Stochastic Programming Approaches to Stochastic Scheduling. Journal of Global Optimization **9**(3-4), 417–451
5. Bourgeois, L. J., Eisenhardt, K. M. (1988) Strategic Decision Process in High Velocity Environments: Four Cases in the Microcomputer Industry. Management Science **34**(7), 816–835
6. Cooper, R. G., Kleinschmidt, E. J. (1987) New Products: What Separates Winners From Loosers. Journal of Product Innovation Management **4** (3), 169–184
7. Dobson, G., Kalish, S. (1988) Positioning and Pricing a Product Line. Marketing Science **7**(2), 107–125
8. Gollmer, R., Nowak, M. P., Römisch, W., and Schultz, R. (2000) Unit Commitment in Power Generation - a Basic Model and Some Extensions. Annals of Operations Research **96**, 167–189
9. Kall, P., Wallace, S. W. (1997) Stochastic Programming - Interscience Series in Systems and Optimization. Wiley, Chichester
10. Lageweg, B. J., Lenstra, J. K., Rinnooy Kan, A. H. G., Stougie, L. (1985) Stochastic Integer Programming by Dynamic Programming. Statistica Neerlandica **39**(2), 97–113
11. Laporte, G., Louveaux, F. V., van Hamme, L. (2002) An Integer L-Shaped Algorithm for the Capacitated Vehicle Routing Problem with Stochastic Demands. Operations Research **50**(3), 415–423
12. Milliken, F. (1987) Three Types of Perceived Uncertainty about the Environment: State, Effect & Response Uncertainty. Academy of Management Review **12**, 133–143
13. Schultz, R. (2003) Stochastic Programming with Integer Variables. Mathematical Programming **97B**, 285–309
14. Spence, A. M. (1980) Multi-Product Quantity-Dependent Prices and Profitability Constraints. Review of Economic Studies **47**, 821–841
15. Urban, G. L., Hauser, J. R. (1993) Design and Marketing of New Products (2nd edition) Prentice Hall, Englewood Cliffs
16. von Hippel, E. (1992) Adapting Market Research to the Rapid Evolution of Needs for New Products and Services. Working paper (Sloan School of Management); WP 3374-92

Optimising energy models for hydrothermal generation systems to derive electricity prices

Dominik Möst, Ingela Tietze-Stöckinger, Wolf Fichtner, Otto Rentz

Institute for Industrial Production,
Universität Karlsruhe, Hertzstraße 16, 76187 Karlsruhe
{dominik.moest, ingela.tietze-stoeckinger, wolf.fichtner, otto.rentz}@wiwi.uni-karlsruhe.de

Abstract. Within this paper, the energy model PERSEUS-HYDRO and its application to Switzerland with a focus on the possibility to provide long-term electricity prices will be presented. Within this model perfect energy markets are assumed and thus electricity prices can be derived from system-marginal-costs. Main features of the plant dispatch within a system with hydro and thermal power will be shown. Also, the integration of energy exchanges (e.g. EEX) are foreseen within the optimising model. As the plant dispatch mainly influences the marginal costs, the important effects and coherences will be highlighted. As prices on the energy exchange have to be set as exogenous input, their influence on the marginal costs will also be discussed.

1 Introduction

The world wide liberalisation of energy markets has far-reaching implications for energy utilities. Competition on national and supranational level leads to a situation in which strategic planning tasks such as capacity expansion and long term production planning become increasingly important for energy utilities. Especially electricity prices get more and more important for investment strategies of energy utilities.

Due to the fact, that water power plants in the Alps have a relevant effect on electricity prices in Europe, it is important to analyse energy systems with a high share of water power in detail. Modelling a generation system with a high share of water power plants (pump storage, reservoir and run-of the river plants) poses additional and different requirements compared to a conventional plant portfolio for the derivation of electricity prices.

Hence, the present paper aims to describe a model application for the analysis of water power plant operation and competitiveness as well as the possibility to derive future electricity prices from a linear optimisation approach. Therefore, the model approach and the specific characteristics of water power plants in an energy model will be presented. Price information based on system-marginal-costs will be discussed and the influence of the plant dispatch and energy exchanges on the marginal costs will be presented. Furthermore, the application of the developed model to the Swiss energy sector will be shown.

2 A modelling approach to improve the representation of water power plants

The energy and material flow model family PERSEUS (Programme Package for Emission Reduction Strategies in Energy Use and Supply) has been developed at the Institute for Industrial Production (IIP) in order to be able to address questions related to energy systems not only on national, but also on regional and utility level. The modelling approach is based on a detailed representation of energy conversion technologies and the interconnecting flows of energy and material (i. e. primary energy carriers, emissions of pollutants etc.). The structure of the model is equivalent to a directed graph: Energy and material flows are conform with arrows. Conversion technologies and their processes correspond to the nodes of the graph. The models follow linear and mixed integer programming approaches. A detailed description of the entire modelling package can be found in [4].

In the following, this paper will focus on a specific model of the PERSEUS family, the PERSEUS-HYDRO model, which aims at the modelling of energy systems with a large-scale of water power plants and supporting energy utilities by providing information on future electricity prices. Within prior model approaches, the dispatch of water power plants has been set exogenously, while within this approach the dispatch is optimised. Interested readers on further methods of modelling water power systems are referred to [2, 3].

Let us suppose, the energy system to be analysed consists of a set of $units=\{1,...,m\}$ of existing power plants (units) and power plant options, which can be built up in the future. Each of the power plants $unit \in Unit$ contains at least one process $proc \in Proc$ ($Proc=\{1,...,n\}$), which reproduces the operation mode of the power plants. The processes represent the transformation process of the in- and outgoing energy carriers. The typical time horizon (about 30 years) is divided into single periods $t \in T$, where each period is subclassified into time intervals $seas \in Seas$. Time intervals are necessary to draw up the energy demand (load curve) for typical days within one period. Each time interval represents a time range of a typical day (e.g. 9.00 to 12.00 o'clock of a summer working day) for all days with this characteristic. Thus the energy demand within one time interval $seas \in Seas$ and one period $t \in T$ represents an average energy demand, which has to be multiplied with the amount of characteristic days within this period. The exogenous energy demand $D_{t,seas}$ has to be satisfied by the power plants. Variables of the model approach are the process levels $PL_{proc,t,seas}$ of the various operation modes of the energy units and the energy flows $FL_{prod',prod,t,seas}$ between the nodes of the system. Furthermore, the capacity of power plant options (new plants) $NewCap_{unit,t}$ is an integer decision variable. Due to the fact that water power cascades are modelled very detailed, also water flows $FL_{prod',prod,t,seas}$ and water basins $ST_{prod,t,seas}$ are decision variables. Furthermore, the purchase and selling of electricity at an energy exchange is model by decision variables $B_{imp_{b,t,seas}}$ and $C_{exp_{b,t,seas}}$.

Hence the optimisation model can be formulated as follows:
Target function: cost minimization:

$$\min \sum_{t\in T}\alpha_t \cdot \left[\begin{array}{l} \displaystyle\sum_{imp\in IMP}\sum_{seas\in S}\sum_{prod\in PROD} FL_{imp,prod,t,seas}\cdot C_{fuel_{prod,t,seas}} \\ \displaystyle +\sum_{seas\in S}\sum_{proc\in PROC} PL_{proc,t,seas}\cdot C_{var_{proc,t,seas}} \\ \displaystyle +\sum_{u\in U}\begin{bmatrix} Cap_{unit,t}\cdot\left(C_{fix_{unit,t}}+C_{wat_{unit,t}}\right)+ \\ NewCap_{unit,t}\cdot C_{inv_{unit,t}} \end{bmatrix} \\ \displaystyle +\sum_{b\in B}\sum_{seas\in S} B_{imp_{b,t,seas}}\cdot C_{buy_{b,t,seas}} -\sum_{b\in B}\sum_{seas\in S} B_{exp_{b,t,seas}}\cdot C_{sell_{b,t,seas}} \end{array}\right] \tag{1}$$

and simplified versions of selected important restrictions:
Energy balance equations:

$$\begin{aligned} &\sum_{imp\in IMP} FL_{imp,prod,ec,t} + \sum_{prod'\in PROD_{prod,ec}} FL_{prod',prod,ec,t} + \sum_{proc\in GENPROC_{prod,ec}} PL_{proc,t}\cdot\lambda_{proc,ec} \\ &= \sum_{exp\in EXP} FL_{prod,exp,ec,t} + \sum_{prod'\in PROD'_{prod,ec}} FL_{prod,prod',ec,t} + \sum_{proc\in DEMPROC_{prod,ec}} PL_{proc,t}\cdot\frac{\lambda_{proc,ec}}{\eta_{proc,ec}} \\ &\forall t\in T;\quad \forall prod\in PROD;\quad \forall ec\in EC_{non-seas} \end{aligned} \tag{2}$$

Demand equations:

$$\sum_{prod\in Prod}\sum_{exp\in Exp} FL_{prod,exp,t,seas} \geq D_{t,seas} \quad \forall prod\in Prod; \forall seas\in S; \forall t\in T \tag{3}$$

Process utilization equation:

$$\begin{aligned} &Cap_{unit,t}\cdot Avai_{unit,t}\cdot h_{seas} \geq \sum_{proc\in PROC_{unit}} PL_{proc,seas,t} \\ &\forall t\in T;\quad \forall unit\in UNIT;\quad \forall seas\in SEAS \end{aligned} \tag{4}$$

Cross linking of water power plants

$$\sum_{prod\in PROD} FL_{prod,prod',\text{"water"},t,seas} = \sum_{prod''\in PROD} FL_{prod',prod'',\text{"water"},t,seas}\cdot\delta_{prod',\text{"water"}} \tag{5}$$

$$with\ \delta_{prod,prod',prod'',\text{"water"}} = \frac{BFH_{prod',prod''}}{BFH_{prod,prod'}}$$

$$\forall seas \in SEAS;\ \forall t \in T; \forall prod' \in PROD_{hydro}$$

Decision variables:

$$PL_{p,t,seas} \in R^{+} \tag{6}$$

$$FL_{prod',prod,t,seas} \in R^{+} \tag{7}$$

$$Cap_{unit,t},\ NewCap_{unit,t} \in Z^{+} \tag{8}$$

$$ST_{prod,t,seas} \in R^{+} \tag{9}$$

$$B_{imp_{b,t,seas}},\ B_{\exp_{b,t,seas}} \in R^{+} \tag{10}$$

The target function (1) of the optimising model is the minimization of costs. Therefore cost coefficients are assigned to the decision variables; fuel costs $C_{fuel_{prod,t,seas}}$ to the energy flows $FL_{imp,prod,t,seas}$ and variable costs $C_{var_{proc,t,seas}}$ to the process level $PL_{proc,t,seas}$ of the various operation modes of the energy units. Fix costs $C_{fix_{unit,t}}$, water taxes $C_{wat_{unit,t}}$ and specific investments $C_{inv_{unit,t}}$ of new plants are dedicated to the capacity decision variables $Cap_{unit,t}$ and $NewCap_{unit,t}$. Furthermore, electricity stock prices $C_{buy_{b,t,seas}}$ and $C_{sell_{b,t,seas}}$ are assigned to the purchase and selling electricity decision variables $B_{imp_{b,t,seas}}$ and $B_{exp_{b,t,seas}}$. Within equation (2) the energy balance for each node of the modelling system is postulated, where $\lambda_{proc,ec}$ denotes the ratio of the energy carrier *ec* of the total in- or output (negative values for input, positive for output) and $\eta_{proc,ec}$ the efficiency of the process *proc*. The exogenous energy demand $D_{t,seas}$ has to be satisfied at any time, which is formulated in equation (3). The process utilization equation (4) ensures, that for each time interval *seas* (e.g. summer working day 9 to 12 o'clock) and for each period *t* the installed capacity $Cap_{unit,t}$ is sufficient to generate enough electricity with the process level $PL_{proc,t,seas}$ based on the length of the time interval h_{seas} and the availability $Avai_{unit,t}$ of the units. Within water cascades the same water flow rate generates a different amount of electricity in each water power plant. Thus, the head of the turbines have to be considered. Within equation (5) the cross linking of water power cascades is considered, therefore the relation of the head of two cross linked turbines $\delta_{prod,prod',prod'',\text{"water"}}$ is taken into account, where $BFH_{prod,prod'}$ is the head of the turbine *prod'*. Additional equations which are not listed above allow to consider further restrictions like e.g. restrictions on the purchase and selling of electricity at the energy exchange, base load restrictions, load variation restrictions, emission restrictions and the supply of control energy.

3 Reflecting electricity pricing in a hydro-thermal generation system

In the following the two terms marginal costs and system-marginal-costs will be used. Thereby marginal costs serve as characterisation for the real market, while system-marginal-costs mean the marginal costs calculated by the optimising model.

One reason (beside the optimised plant dispatch) for the detailed modelling of water power plants can be seen in their variable costs, which are normally zero or near to zero. Thereby three main types of water power plants can be classified [5]:

Run-of the river power plants do not have a water storage and produce continuously electricity depending on the water-flow of the river. Hence, they are used for base load production.

Reservoir power plants have the possibility to store water inflow in a lake or basin. They can further be classified by the size of the storage (day-, week or year-storage). The production of electricity depends on the available water and the electricity prices. Normally, they provide peak load.

Pump storage power plants have the possibility to pump water from an energetic lower to a higher storage. Thus, they can be used to transfer base load electricity to peak load electricity.

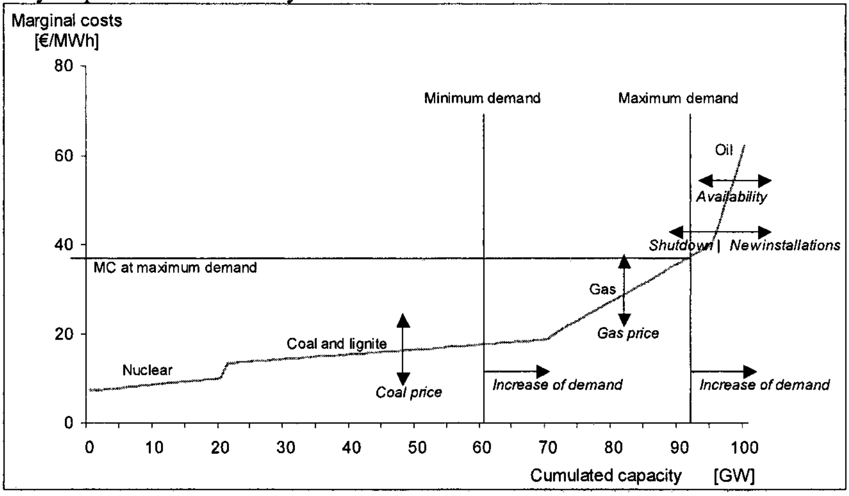


Fig. 1.: Exemplary merit order curve and fundamental influences on the electricity price

Marginal costs of conventional power plants are derived from their variable costs and the plant dispatch normally takes place on the basis of the sorted marginal costs called merit order under consideration of technical restrictions. Nuclear and coal plants are used for base load, while gas and water power plants (storage and pump-storage) are used for peak load. For water power plants, marginal costs can not be derived from their variable costs. Due to the fact, that the amount of available water of each unit determines the production of these water power plants, the marginal costs have to be derived from opportunity costs. This means that for each water power plant the available and expected water (e.g. of one year/month) has to be faced with the expected electricity prices. For modelling water power plants in a linear optimising approach, this means that each water power unit has to be modelled in detail, including cross linking and networking with other water power plants and also the water inflow to basins and lakes. In Figure 1 some fundamental effects on the electricity prices are demonstrated, but it must be pointed out, that the considerations neglect strategic behaviour of the suppliers, which can significantly influence short-term electricity prices. An increase of the maximum demand would lead to higher prices as the intercept point moves more to the right of the supply curve. When the capacity gets narrow, prices can highly increase. Also the gas and coal price, the availability of plants and the installation of new power plants have an effect on the electricity price as shown in Figure 1.

3.1 Price information based on system-marginal-costs and the influence of plant dispatching and energy exchanges

As already described, the energy model determines the future capacity development and plant dispatch by minimising the total costs over all considered years. Perfect foresight of all market participants with regard to market signals like quantity supplied, demand data, future fuel prices, etc. is assumed.

System-marginal-costs characterize the costs, which arise from a marginal increase of the demand $D_{t,seas}$ within one time interval *seas* and one period *t* for the total system. In other words, system-marginal-costs are the opportunity costs of the exogenous demand.

The assignment of the variation of the target function to one time interval *seas* has to be critically reflected, because the target function comprises all time intervals and periods. The plant dispatch in one period *t* and one time interval *seas* can affect the system-marginal-costs in other periods and time intervals. An example is the plant dispatch of a reservoir power plant, which has only a limited water quantity. A higher production in one time interval directly decreases the production in another time interval.

Also the installation of new plants has an influence on several time intervals. New plants will be installed if the capacity is not sufficient to satisfy the exogenous demand. Then the system-marginal-costs will be very high within the time interval of the short capacity, while in other time intervals this additional capacity can be used as well. The reason of these price peaks is the shortage of capacity, contrarily to the real market where strategic behaviour has a higher influence on price peaks. The described (spanning) effects have to be taken into account when deriving electricity prices on system-marginal-costs.

Also the integration of energy exchanges in the optimising model influences the system-marginal-costs. The effect of the energy exchanges will be outlined for three different cases:

1. The equation, which limits the selling of electricity is binding. That means that more electricity could be sold at the energy exchange to the present price, if there would not be a restriction on the quantity. That signifies that the system-marginal-costs are below the exogenous energy exchange price.
2. A binding purchase-equation represents practically the same case. The system-marginal-costs are higher than the energy exchange prices. This means, that a purchase of more electricity from the energy exchange would be cheaper for the total energy system.
3. If none of the two equations is binding, the quantity of purchased or selled electricity is optimal for the total energy system. Thus, the system-marginal-costs are equivalent to the energy exchange prices.

The discussed effects describe the influence of the energy exchange within one period *t* and one time interval *seas* on the system-marginal-costs. Nevertheless, the integration of energy exchanges represents a good possibility to model for example bordering regions or further purchase/selling channels in liberalised markets.

4 Application of the developed model on the Swiss energy system

The developed PERSEUS-HYDRO model has been applied to the Swiss energy system. All water power plants above a capacity size of 10 MW have been modelled in detail to analyse their ability to compete in the European energy sector. In total about 200 water power plants are modelled as single units within the natural water cascades. The resulting optimising model consists of about 380.000 continuous and 1.300 discrete variables and about 260.000 restrictions with about 1.400.000 non-zero elements. Solving the model on a PC with Pentium III - 900 MHz and about 1.6 GB main memory results in calculation times of about 15 minutes with the commercial solver CPLEX 7.0.

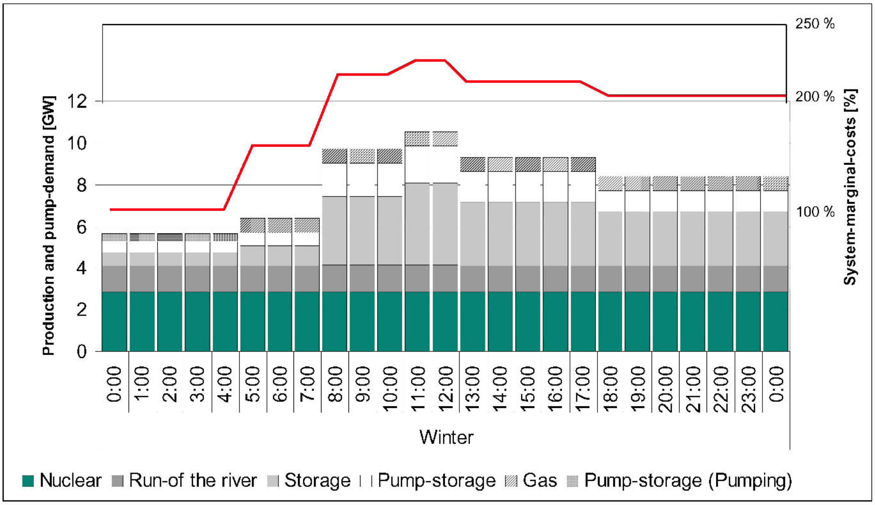


Fig. 2. : Power plant dispatch and system-marginal prices at a winter working day in 2030

Due to the confidential nature of results achieved, the presentation of exemplary model results is in relative figures and not in absolute ones. Figure 2 shows the plant dispatch and the system-marginal-costs of a characteristic winter working day in 2030 in Switzerland. Thereby, nuclear power plants and run-of the river plants are dispatched for base load. Peak load is provided by storage, pump-storage and the combined gas cycle power plant. The pump-process of the pump-storage plant is dispatched in the night hours of the working day. The system-marginal-costs increase with the energy demand until high noon and decline after high noon with the lower demand. At noontime, the system-marginal-costs are about the double price as in the night hours.

Furthermore, the model results show that yearly average system-marginal-costs increase over all periods. This can be attributed to the reduction of overcapacities.

5 Implementation of the models

The models described above have been implemented as a PC version that can be run on commercial PCs. However, due to their high complexity and the resulting large problem size, they require state-of-the-art hardware components.

The PERSEUS models are equipped with an MS Access based data management system that permits easy data handling and a fully automated link to the mathematical module. They are programmed in GAMS (c.p. [1]). In order to solve the problem, commercial solvers like CPLEX can be applied.

6 Acknowledgements

The development and application of the described model have been content of a research project together with the Centre for Energy Policy and Economics (CEPE) at the ETH Zürich, the BKW Energie AG, the Kraftwerke Oberhasli and ETRANS. In this context we would like to thank our project partners, especially M. Balmer and A. Fetz, for providing all the data and for the fruitful discussions and PSEL (Projekt- und Studienfonds der Elektrizitätsswirtschaft) for financing.

References

[1] Brooke, A.; Kendrick, D.; Meeraus, A. and Raman, R. (1998) GAMS - A User's Guide (Edition December 1998). Washington: GAMS Development Corporation

[2] Bushnell, J. B. (2003) A Mixed Complementarity Model of Hydrothermal Electricity Competition in the Western United States.

[3] Edwards, B. K. (2003) The Economics of Hydroelectric Power. Edward Elgar Publishing Limited

[4] Fichtner, W. (1999) Strategische Optionen der Energieversorger zur CO2-Minderung: ein Energie-und Stoffflussmodell zur Entscheidungsunterstützung. Berlin: Erich Schmidt

[5] Filippini, M.; Banfi, S.; Luchsinger, C. and Wild, J. (2001) Perspektiven für die Wasserkraftwerke in der Schweiz - Langfristige Wettbewerbsfähigkeit und mögliche Verbesserungspotentiale. Studie im Auftrag des Bundesamtes für Energie

Simulation of the epidemiology of *Salmonella* in the pork supply chain

M.A. van der Gaag[a], H.W. Saatkamp[b], F. Vos, M. van Boven[a], P. van Beek[b] and R.B.M. Huirne[a]

a) Animal Sciences Group Wageningen UR, Lelystad, the Netherlands b) Business Economics, Wageningen University, The Netherlands (corresponding author: monique.vandergaag@wur.nl)

Abstract
A major food safety issue in pork is Salmonella contamination. A stochastic state-transition simulation model was described to simulate the spread of Salmonella from multiplying through slaughter, with special emphasis for critical control points to prevent or reduce Salmonella contamination. Design of Experiments and metamodelling were used for a sensitivity analysis. The finishing stage and the slaughterhouse appeared to be the most important stages in the supply chain to reduce the prevalence of Salmonella contaminated carcasses.

1. Introduction

Public health and food safety is a major issue in current agri-business. In developed countries, infections with Salmonella species contribute a great deal to food borne diseases and therefore are accountable for considerable societal costs (Anonymous 2001). Salmonellosis is caused by bacteria from the genus Salmonella, and can result in severe gastro-enteritis, enteric fevers and septicaemia. It is estimated that 90% of the human salmonellosis is food borne (Anonymous 2001) and about 15% of the food borne salmonellosis originate from pork (Berends 1998). To reduce the incidence and the societal costs, more insight is needed in the impact of measures to reduce or prevent Salmonella in food products. For an effective control, the entire supply chain must be involved, which include primary producers and slaughterhouses. Currently, there is only limited insight in costs and benefits of Salmonella control, although costs of implementing measures can be very high. Hence, improved insight in the (cost)effectiveness of certain control measures will contribute to decision making. Therefore, a detailed stochastic state-transition model was designed to simulate the introduction and spread of Salmonella in the pork supply chain in order to estimate cost-effectiveness of various kind of control measures against Salmonella.

2. Model description

2.1 Salmonella control in pork production

An animal that is infected with Salmonella can start shedding bacteria, and therefore become infectious within four hours. The seroconversion period (to reach detectable antibody levels after infection) is about two weeks. Upon recovery, an animal can remain in a carrier state (bacteria in intestines or lymph nodes, but no shedding), and may become serological negative again. At the slaughter stage, bacteria from the intestines or lymph nodes can contaminate the carcass and thereby contaminate pork products. The prevalence (i.e. percentage of infected pigs in a population) can be measured by serological or bacteriological tests.
The pork supply chain consists of several stages, quite often located on different farms. Figure 1 shows the different stages in the chain: breeding, multiplying, finishing, transportation, slaughtering (lairage and slaughter), processing, retail, and consumer. All stages have relations with partners that are linked to the main food supply chain, such as feed companies and service suppliers. The model includes the stages called multiplying through slaughter. The multiplying stage and the first transport stage are combined in the model, and will be called the multiplying stage, from now on. The slaughter stage includes the slaughtering of the pigs, and the chilling of the carcasses.

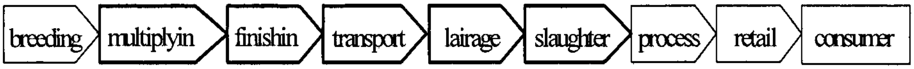


Fig. 1. Schematic structure of the pork supply chain. The bold outlined stages are included in the simulation model.

Table 1 shows specific control measures and routines that can be implemented per stage and per risk profile, to prevent or reduce the introduction and spread of Salmonella. Farms or firms in a stage that take these control measures, are considered to have a low-risk profile. A farm or firm with a high-risk profile does not take these control measures and may follow routines that have a higher risk of introduction or spread of Salmonella.

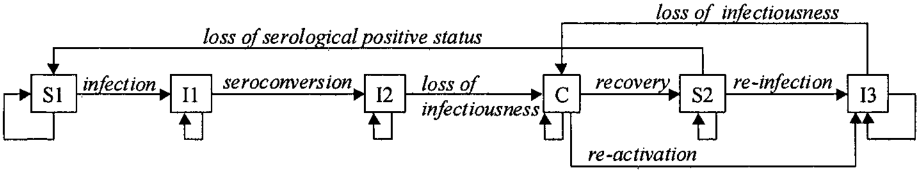
2.2 The simulation model

The aim of the simulation model is to mimic the dynamics of Salmonella within and between the various stages of the production chain, subject to various conditions and measures to prevent or reduce introduction and spread; the model outcome is the prevalence of Salmonella at the various stages. The basic unit of the model is a group of 100 animals that remains together during the entire production process. Various kinds of contact structures among individual farms and firms exists, which is a key issue for the introduction and spread of Salmonella in the supply chain, since a farm with a high prevalence of Salmonella can contaminate several farms/firms in the subsequent stage.

The dynamics of Salmonella infections in pigs was modelled using a discrete time stochastic state-transition approach including Monte Carlo simulation (Alban and Stärk 2002), including the state vector **x**(t) and the vector with transition probabilities $p_{i,j}(t)$, from now on written as $p_{i,j}$. The time step used in the model equals one day. State vector **x**(t) contains six mutually exclusive states that animals can be in, described in Figures 2A and 2B, and the 100 animals of one group are distributed over these states. In each time step, transition probabilities are recalculated taking into account the control measures and various decisions made, thereby obtaining a new transition matrix.

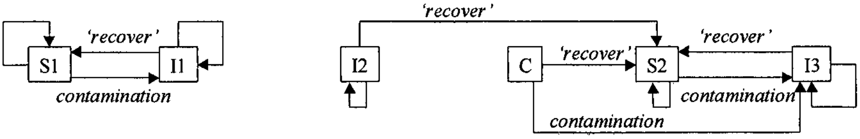
Table 1: Major control measures and routines per stage that determine the risk profile

Stage	Low risk	High risk
Multiplying	* certified Salmonella (S.) free feed * fermented feed for piglets * functional hygiene lock * cleansing and disinfecting per round	* pelleted feed (unknown S. status) * free entrance for visitors * no structural rodent control
Finishing	* purchase of certified S. free piglets * fermented feed / non-heated barley * functional hygiene lock * no contacts between compartments	* pelleted feed (unknown S. status) * free entrance for visitors * no structural rodent control * no all-in-all-out
Transport	* cleansing and disinfecting each ride * smooth, good cleansable materials * quiet driving and short distances	* no fasting of pigs before transport * pigs from several compartments in finishing farm mixed in one truck
Lairage	* good hygiene * reduced duration in lairage (<2 hours)	* mixing of pigs from different origin * open fences between compartments
Slaughtering	* cleansing equipment during the day * careful evisceration	* no direct packing of rectum * less time for evisceration



S1 :	susceptible 1	: Salmonella free animals, negative serology
I1 :	infectious 1	: infected and infectious, negative serology
I2 :	infectious 2	: infected and infectious, positive serology
C :	carrier	: carrier animals, not shedding, positive serology
S2 :	susceptible 2	: susceptible again, positive serology
I3 :	infectious 3	: infected and infectious (>1st infection)

Figure 2A: States and transitions for live animals



S1 :	susceptible 1	: Salmonella free carcass, negative serology
I1 :	infectious 1	: contaminated carcass, negative serology
I2 :	infectious 2	: contaminated carcass, positive serology
C :	carrier	: carrier, not a state of a carcass
S2 :	susceptible 2	: Salmonella free carcass, positive serology
I3 :	infectious 3	: contaminated carcass (>1st infection)

Figure 2B: States and transitions for carcasses

3. Sensitivity analysis and statistical methods

The goal of sensitivity analysis is to determine how sensitive the output is to changes of the inputs and if the directions of output changes are consistent with the expected directions (Kleijnen and Sargent 2000). In our study, Design of Experiments (DOE) with meta-modelling is executed. This design is a specific fractional factorial design that minimises the number of combinations of factor levels (scenarios) that have to be run. Design of Experiments (DOE) and metamodelling with regression analysis indicates which factors are most important and which factors deserve most attention in further research. This approach is preferred over the more common approach of changing one factor at the time; DOE may not only be used for sensitivity analysis, but also for validation (Kleijnen 1998).

The simulation model has 26 input variables and parameters (together named 'factors'), and this specific design enables the estimation of main effects not confounded with two-way interactions (so called resolution IV design).
A metamodel is an approximation of the input-output transformation that is implied by the simulation model (Kleijnen and Sargent 2000). The metamodel is based on the scenarios implied by the specific design. Our metamodel was specified as the following simple first-order polynomial with k variables for each scenario i:

$$Yi = \beta_0 + \Sigma^{k}_{h=1} \beta_h x_{i,h} + e_i$$

where Yi denotes a characteristic of the simulation output of scenario i, β_0 is the intercept, β_h the main effect of variable h, $x_{i,h}$ the value of the standardised variable h in scenario i, and e_i the approximation error plus intrinsic noise in scenario i.
The output of the model is the distribution of carcasses over the states. This distribution over states can be characterised through different quantities, such as its mean and standard deviation. These quantities are estimated based on simulation iterations (Kleijnen and Sargent, 2000); in our sensitivity analysis the number of iterations was set to 1000 timesteps.

The following four output quantities (Yi) were considered (i = 1, .., 4):
Y1: mean number of contaminated carcasses in a group that leave the last stage (i.e. mean number of carcasses that are bacteriological positive; i.e. in state I1, I2, and I3);
Y2: standard deviation (sd) of *Y1*;
Y3: mean number of serological positive carcasses in a group that leave the last stage (i.e. mean number of carcasses in state I2, I3, and S2);
Y4: standard deviation (sd) of *Y3*.
To estimate the metamodel, ordinary least squares (OLS), we use linear regression using a factor selection procedure called the backward method. This means that all factors are entered in the regression model and then sequentially removed, if not significant. A factor was considered to have a significant main effect if its p<0.05. Also two-way interactions were calculated: factors that were found to have significant main effects, were next included and tested for two-way interaction using least squares linear regression with the backward selection method.

4. Results

Figure 3 shows the cumulative distribution functions for the number of contaminated carcasses per group that leave the slaughterhouse in four different scenarios: (1) all factors at the minimum value, (2) all factors at the maximum value, (3) all factors at the average value between the minimum and maximum, and (4) all factors at the default value.

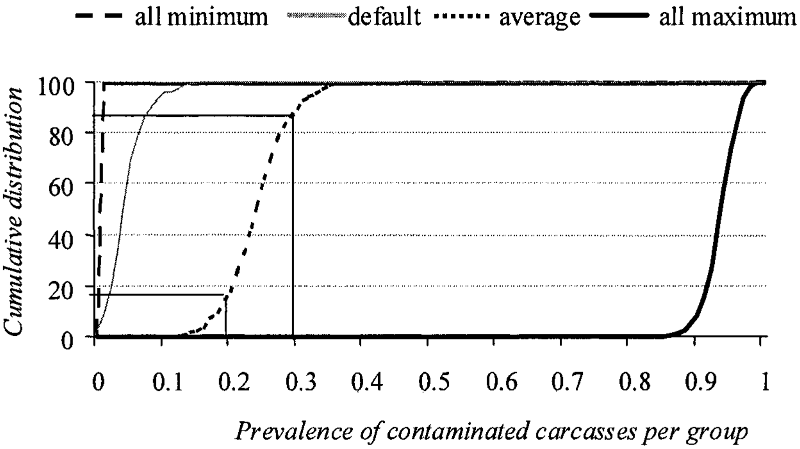


Figure 3: Distribution function for the number of contaminated carcasses per group that leave the slaughterhouse: four different scenarios (1000 timesteps)

The significant main effects and two-way interactions for each of the four output quantities are shown in Table 2. A positive main effect indicates that changing the factor from its minimum to its maximum value results in an increase of the output value. For instance, increasing of the factor $p_{S1,I2}$ in the multiplying stage (from minimum value of 0.1% to maximum value of 25%) will increase the mean number of serological positive carcasses (*Y3*) with 19.9. In *Y2* the factors Qevisc and Qproc had significant main effects, but after adding the two-way interaction to the regression model, only the interaction remained significant. The same is applicable for other main effects, when interaction is introduced. Testing the interactions of the factors for output *Y1* results in three significant interactions at the slaughterhouse.

Table 2: Significant main effects and two-way interactions for each output quantity *Y1* through *Y4* ($p<0.05$)

Main factor/ Interaction	Stage	*Y1* (mean I)	*Y2* (sd *Y1*)	*Y3* (mean serol.pos.)	*Y4* (sd *Y3*)
R^2_{adj}		0.88	0.37	0.82	0.19
Intercept		-14.8	2.8	-6.3	4.1
$p_{S1,I2}$	Multiply			19.9	-6.1
$\beta 1$	Finishing	12.9		44.1	14.5
$\beta 1 * PE_{finishing}$	Finishing			26.4	
Qevisc	Slaughter	-8.2			
Qproc	Slaughter	-7.7			
Qevisc*Qproc	Slaughter	35.1	5.3		
Qdown	Slaughter	-8.2			
Qdown*Qevisc	Slaughter	19.1			
Qdown*Qproc	Slaughter	19.1			

5. Discussion and conclusions

The metamodels for *Y1* and *Y3* had a high R^2_{adj} (0.88 and 0.82), which indicate that the metamodel is a good predictor for the entire simulation model. The R^2_{adj} for *Y2* and *Y4* were low (0.37 and 0.19) and therefore the results must be handled with more care. The most important output quantity with respect to food safety is *Y1*, the mean number of contaminated carcasses that leave the slaughterhouse. The significant main factor for *Y1* from the finishing stage is the rate of infection within a group. This rate may be decreased by feeding strategies and by reducing the possibilities of animals having contact with manure. The slaughtering stage seemed to have even a higher impact on the mean number of contaminated carcasses. And several interactions were found to be significant. To reduce (cross)contamination of carcasses, it is important to have good hygiene in the slaughterhouse, such as rectum removal in bags, careful evisceration and disinfecting of the equipment. The serological status of the carcasses (*Y3*) depends on two main factors and one interaction. A higher starting prevalence ($p_{S1,I2}$) in the multiplying stage and a higher $\beta 1$ results in more infected and therefore serological positive animals. The $PE_{finishing}$ had an understandable interaction effect with $\beta 1$; if the probability that an animal in a group becomes infected increases, the infection rate within the group ($\beta 1$) becomes more important.
It was concluded that:

the default model is responding according to what would be expected based in current epidemiological knowledge;

DOE is an efficient way of reducing required time for sensitivity analysis and enabling the estimation of the impact of interaction between model parameters;

the simulation model is sufficiently credible for use in subsequent simulation experiments focused on evaluation of the cost-effectiveness of control measures against Salmonella.

References

Alban L, Stärk KDC (2002) Simulating Salmonella prevalence from the growing pig to the slaughtered carcass: where should the effort be put to increase food safety? Proceedings for Society for Veterinary Epidemiology and Preventive Medicine Cambridge, England, 13 pp.

Anonymous (2001) Report to the European Parliament and to the Council on the measures to be put in force for the control and prevention of zoonoses, Commission of the European Communities, Brussels, 2001/0176, 2001/0177 (Downloadable from website http://europa.eu.int/eur-lex/en/com/pdf/2001/en_501PC0452_01.pdf)

Berends BR, van Knapen F, Mossel DAA, Burt SA, Snijders JMA (1998) Impact on human health of *Salmonella* spp. on pork in The Netherlands and the anticipated effects of some currently proposed control strategies. International Journal of Food Microbiology, 44:3, pp.219-229

Kleijnen JPC (1998) Experimental design for sensitivity analysis, optimisation, and validation of simulation models. In: Banks, J. (Ed.). Handbook of simulation, Wiley, New York, pp. 173-223

Kleijnen JPC, Sargent RG (2000) A methodology for fitting and validating metamodels in simulation. European Journal of Operational Research 120, pp. 14-29

Modeling Signal Transduction of Neural System by Hybrid Petri Net Representation

Shih Chi Peng[1], Hsu-Ming Chang[1], D. Frank Hsu[2], Chuan Yi Tang[1]

[1.] Dept. of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan, ROC
shihchi@url.com.tw; hmchang@life.nthu.edu.tw; cytang@cs.nthu.edu.tw

[2.] Dept. of Computer and Information Science, Fordham University, New York, NY 10023, USA
hsu@cis.fordham.edu

Abstract

Biological neural system can be considered as a series of biochemical reactions and signal transmission. It is important to provide an intuitive representation of the neural system to biologists while keeping its computational consistency. In this paper, we propose a method to exploit Hybrid Petri Net (HPN) for intuitive representation and quantitative modeling. The HPN is an extension of Petri Nets and represented by a directed, bipartite graph in which nodes are either discrete/continuous places (such as ion channels) or discrete/continuous transitions (such as phosphorylation), where places represent conditions and transitions represent activities. It can easily model the interactions among receptors, ionic flows (such as calcium), G-proteins, protein kinases and transcription factors that are very complicate in terms of the dynamics of all participants and their correlations. We demonstrate that, in the biological neural system, it is possible to translate and map these complex phenomena into HPNs in a natural manner. In our model, the dynamic properties of the neural signal processing can be examined, especially the interactions among neural modulators and signal transduction pathways. With such a mechanism model in hand, our ability to collaborate with neural scientists is greatly enhanced so as to simulate and examine the robustness of the neural transmission under the local biochemical perturbations.

1. Introduction

Signal transduction is a major mechanism that a cell uses to convert an external stimulation into a series of biochemical reaction. Many molecules can be viewed as triggers or effectors at different stages of the highly complicated transduction activities. We hope to use a computational model to help understanding the mechanism of signal transduction in a neuron when we analyze experimental data and conduct simulations. Nevertheless, as a first step, we need a representation of the complex system with relevant information.

With proper representations, every component in a network can be analytically coupled and evaluated. Representations of bio-system can be loosely classified into four main groups [1]: ontologies, database models, structural network models and quantitative analysis models. Theses traditional models are based on systems of ordinary differential equations (ODEs) such as Michaelis-Menten equations [2], graph theory model [3], and discrete event technique [4]. In recent years some Hybrid Petri Net (HPN) models have been proposed for gene regulatory networks and metabolic networks [2] [5]. HPNs are well suited for describing and analyzing the structure and function of dynamic systems. In bio-system, HPN's enable us to maintain various characteristics of the biological process such as DNA or protein concentration using continuous value. HPN's also allow us to model and visualize system concurrency.

The cellular signal to trigger the whole transduction network can be as simple as a small ionic current passing through channels on the cell membrane, after ligands binding to receptors. Some kinases are activated there after. Eventually, various kinases cause side chains of some peptides such as tyrosine (tyrosine kinase) or serine/threonine (serine or threonine kinase) to be phosphorylated. The conformation of the protein is altered and its activity changed.

In this paper, we focus on modeling signal transduction process of the neural system using Hybrid Petri Net representation. In Section 2, we review and define the modeling concept of a Hybrid Petri Net in the context of a neuron system. In Section 3, we use the HPN to model the signal transduction pathways in a neuron. An example is given to illustrate our framework. Section 4 discusses issues for future work.

2. Hybrid Petri Net

Petri Net was developed in the 1960's by Carl Adam Petri [6]. It is a mathematical and graphical modeling tool used in various applications domain including communication protocols, production systems and flexible manufacturing systems. Traditional Petri Net can only be used in a model with discrete events and not suited for continuous system like fluid mechanics.

Hybrid Petri Net (HPN) is an extension of a Petri Net that handles continuous events by real number. There are two kinds of places and transitions in HPN, discrete/continuous places and discrete/continuous transitions. A discrete place and a discrete transition hold nonnegative integer number tokens as its content. A continuous place can contain nonnegative real number. A continuous transition fires continuously by a given speed function and values in the places. In addition, two kinds of arcs are defined. Normal arc makes the tokens in a place increase or decrease and trigger next transition. Passing arc doesn't decrease tokens but trigger connected transition. The graphical notations are shown in Figure 1

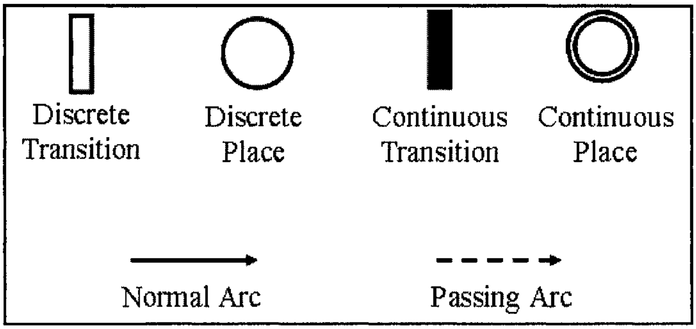


Fig. 1. Elements of Hybrid Petri Net

Now we define and denote HPN as a 7-tuple (P, T, h, Pre, Post, A, M_0), where

- **P** = $\{P_1,\ldots, P_n\}$ is a finite, non empty set of places
- **T** = $\{T_1,.., T_m\}$ is a finite, non empty set of transitions
- **h**: $P\cup T\rightarrow \{d,c\}$ indicates places or transitions whether it is a discrete or continuous one.
- **Pre** **(P_i,T_j)** and **Post** **(T_i,P_j)** defines arc from a place P_i to a transition T_j and a transition T_i to a place P_j respectively, where the arc has a weight of non-negative integer if $h(P_i)= d$, and has a speed function if $h(P_i) = c$.
- **A**: $Pre\cup Post\rightarrow \{n,p\}$ indicates whether an arc is normal arc or a passing arc
- **M_0** is initial marking of the node in the HPN.

HPN is a transition triggered networks. Transition is enabled as fallows [7]. A discrete transition is enabled if each input place of the transition T_j has $M_n(P_i) \geq$ Pre(Pi, T_j). For a continuous transition, it is enabled by the two conditions: (1) each input to the discrete place P_i of transition $T_{j,}$ $Mn(P_i) \geq Pre(P_i, T_j)$ or (2) each input to a continuous place P_i of transition T_j, $M_n(P_i) > 0$. We assign a delay time d_t to discrete transitions and a speed function to continuous transitions. Additional, transitions fire with some conditions like if A more then Q_1 and B more then Q_2 then fires. There are three steps in the mechanism of a transition. As an example, the continuous transition shown in figure 2, the transition first consumes tokens from the input places P_1 by S_1. Then, the transition is fired according to the function T. Finally, products are produced to the output place at speed S_2, Post(P_i, T_j), illustrates the mechanisms of such a transition.

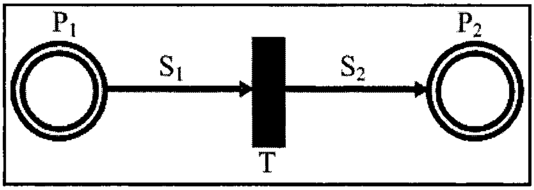


Fig. 2. A continuous transition transforms tokens from P_1 to P_2

3. Modeling Signal Transduction in a Neuron

In this section, some patterns of the biochemical pathway will be defined by HPN. These patterns provide basic blocks to represent signal transduction. It can help to analysis the pathway and easier building the model.

3.1 HPN Pattern of Biochemical Pathway in Signal Transduction

Signal transduction has series of biochemical reaction such as phosphorylation, enzyme reaction, receptor binding, etc. we first describe and define these reactions by using HPN notations. Pattern definitions of these reactions as fallows:

- **Reactant-product relation**

A reactant-product relation is a common biochemical reaction through which reactants are consumed via a reaction and then products are obtained. Figure 3 shows the reaction "A+B -> C+D".

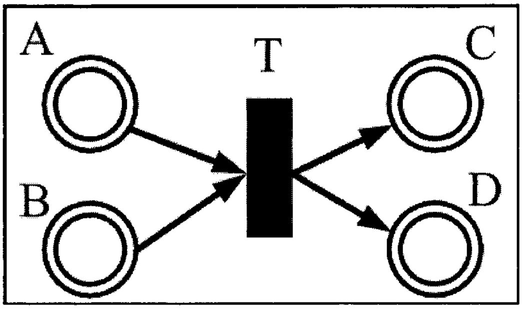


Fig. 3. Transition T consume A and B then produce C and D

- **Catalytic reaction**

Catalytic reaction is a major reaction in signal transduction including enzyme reaction and phosphorylation. Enzyme or protein kinase catalyzes the reaction but does not degrade. There are two kinds of catalytic reactions in a neuron or between neurons. One type is "activation", and the other is "inhibition". Figure 4 is an example of activation where C is a catalytic of the reactant-product reaction "A to B". The passing arc from the continuous to continuous transition T_1 means C would not degrade by transition T_1. On the other hand, the normal arc from C to T_2 represents a degradation of C.

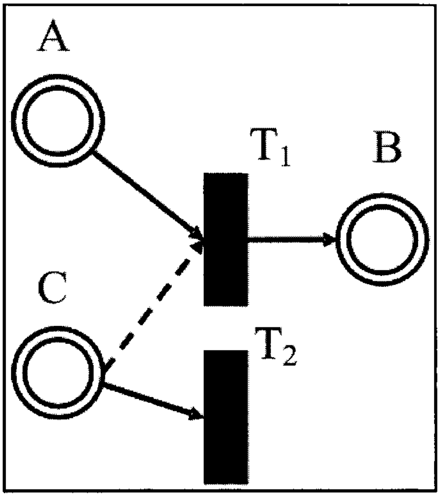


Fig. 4. Catalytic reaction - activation

Inhibition is illustrated in Figure 5 which shows that the amount of quantity in a-B (activated B) and i-B (inactivated B) depends on the amount of quantity in A but in A opposite manner. If A increases, a-B is decreased and i-B is increased. On the other hand, decreasing A would decrease i-B and increase a-B.

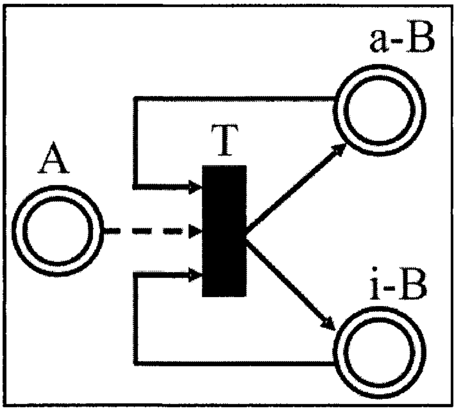


Fig. 5. Catalytic reaction - inhibition

- **Receptor binding**

Receptor binding reactions often occur at the beginning of a signal transduction process. Figure 6 shows that molecule A binds to receptor B which then increases C. Here receptors are presented by discrete place because the number of receptors is a integer.

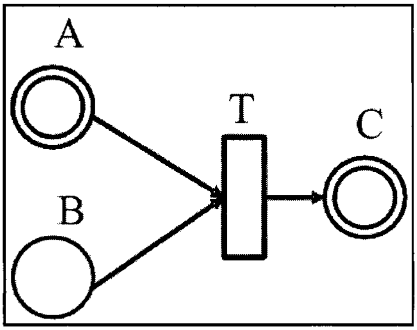


Fig. 6. Receptor binding

3.2 Modeling Signal Transduction of Striatonigral Neuron

Signaling pathway in some neuron has been explored by Greengard [8]. Figure 7 shows one of the major pathways. Dopamine receptors (D_1) are bound by dopamine, cause an increasing the level of cyclic AMP (cAMP) in the cell, and then active protein kinase A (PKA). One of the substrates of PKA is dopamine and cAMP-regulated phosphoprotein (DARPP-32), which is phosphorylated to pDARPP-32 by PKA. The other antagonistic path is started with Glutamate bind to N-methyl-D-asparate (NMDA) receptors, results in an increase of intracellular calcium and followed by the activation of PP2B. Finally, PP2B causes dephosphorylation of pDARPP-32. DARPP-32 plays a central role in the path way, pDARPP-32 inhibits PP-1, PP-1 inhibit NMDA receptor. That forms a loop to control the NMDA receptor. This is a major mechanism that dopamine undertakes to regulate excitability of neurons in the brain.

In our model, Figure 8 represents the correlation of these molecules in the signal transduction pathway in a clean fashion. Using the patterns defined in Section 3.1, it is natural and feasible to describe signal transduction pathways in and between neurons. Although the model is static in structure, it provides representation of complex system in signal transduction, and good for dynamic simulation after every place and transition is supplied with quantitative information. For example, the Hill equation [9] can be used as a quantitative model for transitions so that concentration variations in enzymatic or other biochemical reactions may be simulated. We hope that the model would provide a common language among biologists, computer scientists, and neuroscience scientists.

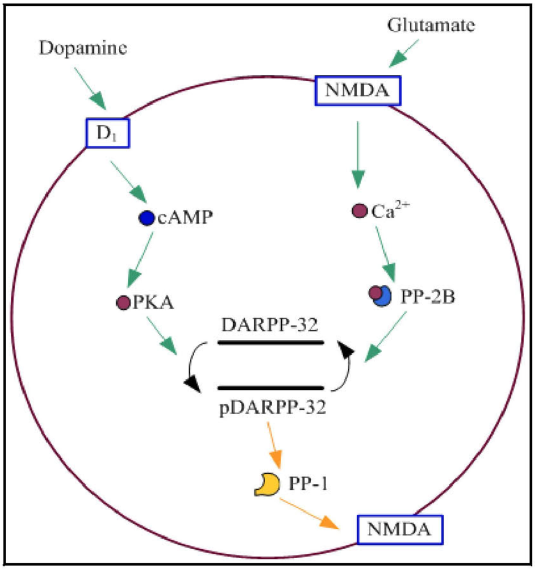


Fig. 7. Signal transduction in a neuron explored by Greengard (2001)

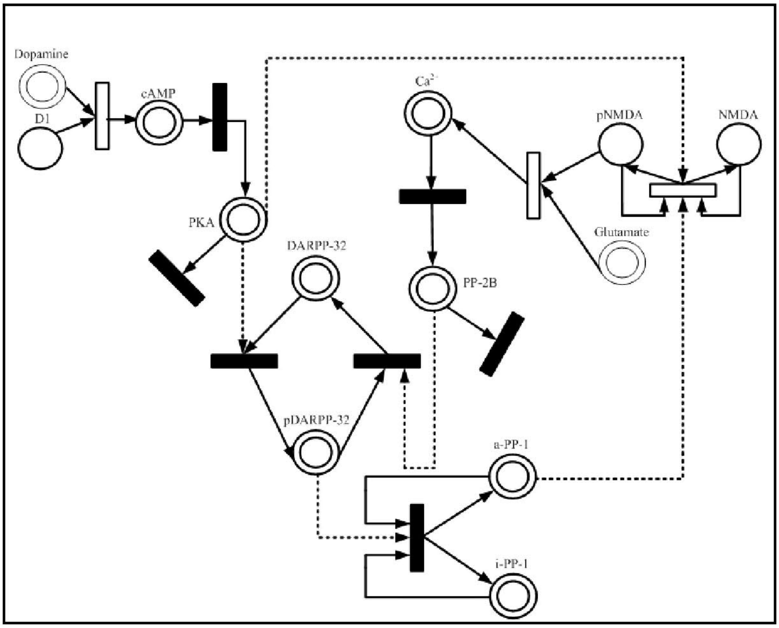


Fig. 8. Representation of signal transduction of a neuron by Hybrid Petri Net. See text for detailed descriptions. The PKA and PP-2B have its own degradation transitions.

4. Future Works

Although we give example depicting an intracellular pathway, we will extend this concept to include intercellular processes. In this regard, we will model signal transmission between neuron cells based on HPN. One application will be extended to the neural networks of memory in human brain.

In the near future, we will extend our modeling study from structure to functional level. We intend to add molecular dynamics to the transitions in HPN model by studying the bio-physical and bio-chemical properties in physiological and pharmacological process of the neural system.

Although we are using the HPN to model the intra- and inter- cellular structure and function of the neural systems, we would like to keep our eyes open. In particular, we do not restrict ourselves to the model-driven approach in which the model is assumed and fixed and the work is concentrated on classifying or associating structure, function, and behavior on neural system. We aim to study the large data set generated by observations, experiments, and clinical testing. We are also looking for specific pattern in the data so as to refine our model. By combining both the model-driven and data-driven approaches, our information-based synergy is likely to lead to contributions to enhance our understanding of neural systems and neuroscience in general.

Reference

(1) J.W. Pinney, D.R. Westhead and G.A. McConkey (2003) Petri Net representations in systems biology. Biochemical Society Transactions Vol. 31, part 6, pp 1513-1515

(2) H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui and S. Miyano (2003) Biopathways representation and simulation on hybrid functional Petri net. In Silico Biology 3, 0032.

(3) M.C.Kohn and D.R.Lemieux (1991) Identification of regulatory properties of metabolic networks by graph theoretical modeling. Journal of Theoretical Biology, Vol.150, pp 3-25

(4) P.A.Meric and M.J. Wise (1998) Quantitative scalable discrete-event simulation of metabolic pathway. Intelligent System for Molecular Biology, AAAI Press, pp.187-194

(5) H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano (2000) Hybrid Petri net representation of gene regulatory network. In: Pac Symp Biocomput, pp 341-352

(6) C.A.Petri (1962) PhD Thesis, Institut für Instrumentelle Mathematik, Bonn

(7) M. Svadova (2001) Modelling Hybrid Dynamic Systems Using Hybrid Petri Nets. In: 13th Int. Conference on Process Control

(8) Paul Greengard (2001) The neurobiology of slow synaptic transmission. Science; Vol. 294, pp 1024-1030

(9) A.V. Hill (1910) The heat produced by contracture and muscular tone. Journal of Physiol. Lond.40, pp. 389-403

(10) Mor Peleg, Iwei Yeh and Russ B. Altman (2002) Modeling biological processes using Workflow and Petri Net models. Bioinformatics Vol. 18 no.6, pp 825-837

Mathematical Modeling and Approximation of Gene Expression Patterns

F.B. Yılmaz[1], H. Öktem[1], and G.-W. Weber[1]

Institute of Applied Mathematics, Middle East Technical University, 06531, Ankara, Turkey

Abstract. This study concerns modeling, approximation and inference of gene regulatory dynamics on the basis of gene expression patterns. The dynamical behavior of gene expressions is represented by a system of ordinary differential equations. We introduce a gene-interaction matrix with some nonlinear entries, in particular, quadratic polynomials of the expression levels to keep the system solvable. The model parameters are determined by using optimization. Then, we provide the time-discrete approximation of our time-continuous model. Finally, from the considered models we derive gene regulatory networks, discuss their qualitative features and provide a basis for analyzing networks with nonlinear connections.

Keywords:
Gene Expression, Gene Regulation, Mathematical Modeling, Gene Network, Inference, Optimization, Dynamical Systems.

1 Introduction

Organisms contain a genetic material which has two main characteristics related to our study: *storage* and *expression* of information. The genetic material has some expressive parts, which are called *genes*. Each gene *stores* interrelated information necessary for *synthesis* of some particular *protein*. *Expression* of this information, i.e., *protein synthesis*, is a foundation for most of the phenotypic processes. Proteins function as *enzymes* and they *stabilize* the *transition states* of the reactions by increasing the rate of these reactions. In addition to this role, together with the genes, proteins *regulate* the synthesis of themselves. This interrelated structure, i.e., genes encode for proteins which regulate expression of genes, constructs a *gene regulatory network* where nodes stand for the genes and the regulatory influences are represented by directed edges between these nodes.

DNA arrays technology, described in [3], makes it possible to measure the expression levels on a genome-wide scale and, consequently, to infer the underlying network structures from experimental measurements, which is called *gene expression profiling.* The expression levels of genes show the relation between genotype and phenotype; this helps understanding biological processes. In our study, given a finite set of expression levels, we propose *a continuous* *model* to describe the *dynamics* of the gene regulatory network, provide a *time-discrete equation* and compare linear and nonlinear approaches.

2 A Time Continuous Model

Given a finite number of gene expression levels for n genes, say $\bar{E}_0, \bar{E}_1, ..., \bar{E}_{l-1}$, where each $\bar{E}_m \in \mathbb{R}^n$ is a column vector representing the gene expression profile at time $\bar{t}_m$, these times satisfying $\bar{t}_m < \bar{t}_{m+1}$ $(m \in \{0, 1, ..., l-2\})$, then the question is how to *infer* the underlying gene regulatory network at the system level. Although the current sampling times of the experimental data constitutes an important challenge in *identification*, continuous nature of the underlying dynamics should be described by a continuous model [1,8].

Chen et al. [4] propose to model the gene expression by the differential equation $\dot{E} = ME$, where $E(t)$ and $\dot{E}(t)$ represents the concentrations and concentration changes of *mRNAs* and *proteins* at time t, respectively. Here, M is a matrix with constant entries representing the interactions between regulatory factors in the cell. In other words, $f_{j,i}(x) = a_{j,i}x$ $(a_{j,i} \in \mathbb{R})$ is used as the regulatory function for the influence level of the i^{th} regulatory factor to j^{th} regulatory factor, where $x = E_i$ represents the expression level of the i^{th} regulatory factor. *Chen et al.* [4] use *Minimum Weight Solutions to Linear Equations (MWSLE)* to determine the regulatory influences. Unfortunately, the algorithm is NP-complete and does not guarantee the solution.

Another continuous model proposed by *De Hoon et al.* [5], is based on the described model. They first approximate the differential equation by a difference equation and use *maximum likelihood estimation* to estimate M, then determine the number and places of the nonzero parameters in the matrix by the so-called *Akaikes Information Criterion [9].* In a more flexible approach, *Sakamoto and Iba* [10] choose the model $\dot{E}_j = f_j(E_1, E_2, E_3, ..., E_n)$ $(j = 1, 2, ..., n)$ with n being the number of genes and f being a function in $E_1, E_2, ..., E_n$. *Sakamoto and Iba* found the functions f_j by genetic programming combined with the least mean square method.

In [1,8], a new approach is introduced based on these models [4,5,10]: $\dot{E} = M(E)E$. Here, in the right-hand side of the equation, we see that the interaction matrix M depends on the current state, E. The study does not use particular functions, e.g., linear or piecewise linear, for inference, instead they suggest using any model class function and *restrict* the solution space to identify a *unique* regulatory network. We remind that the characteristics related to the model function reflect the dynamics in the interactions and incorporate any preinformation about the expected expression levels in mid- and long-term. In a process of *statistical learning* [9], where *training* and *testing* are iteratively coupled, the model class assumptions about the functions $f_{j,i}$ can step by step become improved.

Based on this approach, we assume that sampling times of gene expressions are sufficiently small to prevent aliasing in the sampling of the continuous system and model gene expression patterns by the following general equation:

$$(\mathcal{CE}) \quad \dot{E} = F(E).$$

The right-hand side of this equation, $F(E)$, where $F = (F_1, F_2, ..., F_n)^T$, consists of the sum of the *quadratic (constant, linear)* functions:

$$F_j(E) = f_{j,1}(E_1) + f_{j,2}(E_2) + ... + f_{j,n}(E_n);$$

n being the number of genes considered. In other words, we use quadratic polynomials $f_{j,i}(x) = a_{j,i}x^2 + b_{j,i}x + c_{j,i}$, where $a_{j,i}, b_{j,i}, c_{j,i} \in \mathbb{R}$, to represent the influence of *gene i* to *gene j*. Please note that, herewith, our approach represents the dynamic nature of the regulatory regulations by all constant, linear or quadratic functions.

3 Time-Discrete Equation

A numerical solution for an ODE provides a finite set of values of the solution function which reflects the behavior of the cell metabolism identified by the differential equation. Let us refer to the system $(\mathcal{CE})$. For a particular state, the next states are generated iteratively, i.e., $\hat{E}(t_k)$, $\hat{E}(t_{k+1}), ..., \hat{E}(t_{k+s-1})$ are approximated in consecutive order, where $k, s \in \mathbb{N}$, $s > 1$. Given $E(t_0)$, some initial state, we start with the initial state $\hat{E}(t_0) = E(t_0)$ and predict the next state, which is then used to predict the next states. In our example, we take $h_k = \bar{h}_m$ $(k = m \in \{0, 1, 2, ..., l-2\})$ for the approximation part to be able to compare the experimental data with approximated expression levels, i.e., to be able to compare $\bar{E}_0, \bar{E}_1, \bar{E}_2, ..., \bar{E}_{l-1}$ with $\hat{E}_0, \hat{E}_1, \hat{E}_2, ..., \hat{E}_{l-1}$. Please note that the time differences between approximations can be adjusted according to the underlying biological motivation.

Very analogously from the viewpoint of definition of the time-discrete dynamics provided in [1,8], we present the following discrete equation:

$$(\mathcal{DE}) \quad \hat{E}_{k+1} = \mathbb{F}_k(\hat{E}_k) \quad (k \in \mathbb{N}_0),$$

where $\mathbb{F}_k = ((\mathbb{F}_k)_1, (\mathbb{F}_k)_2, ..., (\mathbb{F}_k)_n)^T$ is a tuple of functions defined in terms of the expression levels of the considered genes for the k^{th} time step, i.e., $\hat{E}_k$, such that:

$$(\mathbb{F}_k)_j(\hat{E}_k) := \ (\hat{E}_k)_j + h_k \sum_{i=1}^{n} f_{j,i}((\hat{E}_k)_i).$$

Please note that in the given formula, $(\hat{E}_k)_j$ denotes the concentration of $gene_j$ at the current state, i.e., at k^{th} time step. In this definition, functions $f_{j,i}(x)$ are the influence functions defined in the previous section. In the linear approach, influence functions are the same for all time steps, but in our case, for each time step we have a distinct set of functions $F_1, F_2, ..., F_n$.

Given an initial metabolic state, the time-discrete equation $(\mathcal{DE})$ given above, calculates the next metabolic state recursively, i.e., the right-hand side of the equation is a matrix multiplication of gene interaction matrices calculated at previous states and the initial metabolic state.

4 Optimization

Although many inverse problems are known to be hard to solve [2], our inverse problem has at least one solution $\hat{M}$ because of the continuous dependence of the nonnegative objective function on M and because of this objective function's (generically) quadratic growth.

When we have a linear system, if the number of knowns (conditions) and unknowns are equal and if we have a full-rank matrix, there is a unique solution. Furthermore, if the number of knowns is even larger, i.e., *overdetermined system*, then we search for the curve that fits to the experimental data best. Given an accuracy criterion, we can numerically search for the curve that fits the data when the needed accuracy value is reached.

4.1 Approximation to the Derivative

Microarray experiments provide us a finite set $\{\bar{E}_0, \bar{E}_1, \bar{E}_2, ..., \bar{E}_{l-1}\}$, where $\bar{E}_m \in \mathbb{R}^n$ is the experiment result given at time $\bar{t}_m$, where $\bar{t}_m < \bar{t}_{m+1}$ and $m \in \{0, 1, 2, ..., l-2\}$. In addition to this information, using *a finite difference quotient*, we get an approximation of the left hand-side of the dynamical equation $\dot{E} = F(E)$ as follows:

$$\dot{\bar{E}}_m := \begin{cases} \frac{\bar{E}_{m+1}-\bar{E}_m}{\bar{t}_{m+1}-\bar{t}_m}, & \text{if } 0 \leqslant m < l-1 \\ \frac{\bar{E}_m-\bar{E}_{m-1}}{\bar{t}_m-\bar{t}_{m-1}}, & \text{if } m = l-1. \end{cases}$$

4.2 Linear Approach

For a static regulatory network, the model parameters, i.e., the entries in the regulatory matrix $M \in \mathbb{R}^{n \times n}$, are determined by the least squares principle which takes the form:

$$\min_{M=(M_{j,i})} \sum_{m=0}^{\ell-1} ||M\bar{E}_m - \dot{\bar{E}}_m||^2 .$$

Here, $\bar{E}_m$ and $\dot{\bar{E}}_m$ are column vectors for the experimental data and difference quotients of the expression levels of n genes at time $\bar{t}_m$ for $m \in \{0, 1, ..., l-1\}$, and $||.||$ is the Euclidian norm.

This optimization problem can *equivalently* be divided into n subproblems, since there is no coupling between the equations corresponding to any two different genes. In fact, in the j^{th} subproblem, we want to minimize the approximation errors with respect to the entries of $(M^T)_j$, i.e., the j^{th} row vector of M written as a column vector:

$$\min_{(M^T)_j} \sum_{m=0}^{\ell-1} (\bar{E}_m^T (M^T)_j - (\dot{\bar{E}}_m)_j)^2 .$$

Please note that the approximated change in the expression levels and expression levels at time point $\bar{t}_m$ are represented by $\bar{E}_m$ and $\dot{\bar{E}}_m$ in this formula. All of these n subproblems are of *canonical* linear least squares form, i.e., they have a vectorial unknown.

4.3 Polynomial Approach

As in the previous section, we have some approximations when we infer the underlying regulatory network by interactions represented by quadratic polynomials. For all genes $j \in \{1, 2, ..., n\}$ and all samples $m \in \{0, 1, ..., l-1\}$, the approximation changes in the following way:

$$(\dot{\bar{E}}_m)_j = (f_{j,1}((\bar{E}_m)_1)+f_{j,2}((\bar{E}_m)_2)+f_{j,3}((\bar{E}_m)_3)+...+f_{j,n}((\bar{E}_m)_n))+(AE_m)_j.$$

Here, $(AE_m)_j$ is used for the approximation error, $f_{j,i} : \mathbb{R} \to \mathbb{R}$ denotes the influence of $gene_i$ to the trancription rate of $gene_j$, being a quadratic function defined as follows: $f_{j,i}(x) = a_{j,i}x^2 + b_{j,i}x + c_{j,i}$, where $x = E_i$ denotes the concentration of $gene_i$ and $a_{j,i}, b_{j,i}, c_{j,i} \in \mathbb{R}$.The minimization problem can again equivalently be separated into n subproblems. For example, the following least squares is used for the j^{th} subproblem:

$$\min_{\breve{A}_j} \sum_{m=0}^{\ell-1} (\breve{E}_m \breve{A}_j - (\dot{\bar{E}}_m)_j)^2,$$

where $\breve{E}$ is a collection of Vandermonde matrices for polynomials where

$$\breve{E}_m = \left[1 \ (\bar{E}_m)_1 \ ((\bar{E}_m)_1)^2 \ ... \ 1 \ (\bar{E}_m)_n \ ((\bar{E}_m)_n)^2 \right]$$

is the m^{th} row vector of the matrix $\breve{E}$, and $\breve{A}_j = ((\breve{A}_j)_1, (\breve{A}_j)_2, ..., (\breve{A}_j)_{3n})^T$ is a column vector consisting of the coefficients of the polynomials for the regulating functions $f_{j,1}$, $f_{j,2}$, ..., $f_{j,n}$.

For a numerical comparison, we used the following randomly generated sample, where the number of observed genes is $n = 3$ and the number of time points is $l = 5$, given in Table 1.

Table 1. *Expression scores of three genes on five time points.*

		$gene_1$	$gene_2$	$gene_3$			
at $\bar{t}_0$	[	0,7778	0,9726	0,7802	]	=	$\bar{E}_0$
at $\bar{t}_1$	[	0,2324	0,6999	0,1815	]	=	$\bar{E}_1$
at $\bar{t}_2$	[	0,7026	0,9657	0,0424	]	=	$\bar{E}_2$
at $\bar{t}_3$	[	0,0032	0,8392	0,0503	]	=	$\bar{E}_3$
at $\bar{t}_4$	[	0,4908	0,0975	0,2250	]	=	$\bar{E}_4$

We applied the approximation algorithm for both the linear case and for our non-linear case and used residuals and residual norms to compare the success of the approaches. Please remember that since we divided our optimization problem into n subproblems, where n denotes the number of genes being considered, we should make n comparisons. For the j^{th} optimization subproblem, we use the *residuals*, i.e., for linear case $(\hat{R}_j)_m = \bar{E}_m^T(\hat{M}^T)_j - (\dot{\bar{E}}_m)_j$ and for non-linear case $(\hat{R}_j)_m = F_j(\bar{E}_m) - (\dot{\bar{E}}_m)_j$ $(m = 0, 1, ..., l-1)$, to express the least squares error at time t_m. Thus, the *residual vectors* $\hat{R}_j \in \mathbb{R}^l$, $\hat{R}_j = ((\hat{R}_j)_0, (\hat{R}_j)_1, ..., (\hat{R}_j)_{l-1})^T$, are key components to compare the approaches. We can also use the norm of the residual vectors for comparisons which is given in the following way: $norm(\hat{R}_j) = \sum_{m=0}^{l-1}((\hat{R}_j)_m)^2$.

Table 2 shows the residual norms for the example we gave in the previous paragraphs. For both the linear and the quadratic case, we have three rows, each standing for a subproblem. For example, if we want to compare the approximation errors for the first gene, i.e., the difference between the experimental data and the approximated values for the first gene, we concentrate on the first row vectors.

Table 2. *Residual norms for linear and quadratic approximation.*

	Linear Approach	Quadratic Approach
1	1,1780	$7{,}1491 * 10^{-31}$
2	0,9676	$5{,}6091 * 10^{-30}$
3	0,1368	$3{,}6482 * 10^{-31}$

5 Gene Network Analysis

For any modeling function, we can construct a weighted directed graph corresponding to a gene regulatory network, where each gene corresponds to a vertex in the graph. The influences are represented by the directed edges and the weights of the influences are described by the modeling functions. In our approach, we use *nonlinear* interactions between genes, in particular, quadratic polynomials. Thus, the weights are defined in terms of these polynomials.

We represent the influence of $gene_i$ *to* $gene_j$ by a quadratic function $w_{j,i} = f_{j,i}(x) = a_{j,i}x^2 + b_{j,i}x + c_{j,i}$, where $a_{j,i}, b_{j,i}, c_{j,i} \in \mathbb{R}$ and $x = E_i$ denotes the expression level of $gene_i$. Assume that all genes which have an influence on the expression level of $gene_i$, are represented by a set $I_j = \{gene_{j,i_1^j}, gene_{j,i_2^j}, ..., gene_{j,i_{n_j}^j}\}$. Then, the change in the expression level of $gene_j$ is approximated by the following summation:

$$F_j(E) = \sum_{\kappa=1}^{n_j} f_{j,i_\kappa^j}(E_{i_\kappa^j}) = \sum_{\kappa=1}^{n_j} (a_{j,i_\kappa^j}(E_{i_\kappa^j})^2 + b_{j,i_\kappa^j}E_{i_\kappa^j} + c_{j,i_\kappa^j}).$$

Given a regulatory network, we can analyze the underlying topology depending on the biological questions to be answered. If we are given a network where the vertex set consists of genes in a particular metabolic reaction and we are asked to find the effects of lack of any gene in this reaction, we search for cut vertices in the given network and analyze the network *connectivity* when any *fixed* cut vertex is removed from the network. If the question concerns the influence of a fixed gene to others, we analyze the shortest paths from the fixed vertex to others. Please refer to [6] for the graph theoretical foundations and algorithms.

If we had regulatory functions, i.e., $f_{j,i} = a_{j,i}$, we would use Dijkstra's Algorithm [7] to find the influence level of one gene to others or a similar algorithm to find the influence levels of all genes to one gene. For further questions, there are also other algorithms proposed for static networks. In our problem, we do not have a fixed network topology, so the data structures and algorithms that will be used to analyse the regulatory network should be decided carefully according to the current biological motivation.

6 Summary and Conclusion

The approximation errors for linear and nonlinear polynomials are quite different. For the linear case, the relative error is quite high, i.e., the model approximates the experimental data not accurately, while quadratic polynomials can approximate the data with negligible approximation errors. Increasing the degree of the polynomials provides a better fit to the training data. However, this results in higher sensitivity to measurement noise and process uncertainties. Therefore, polynomials with the smallest degree providing a reasonable approximation should be preferred.

Since the enzyme concentrations have multiplicative effects, there are strong nonlinearities in gene regulation. This is the most important reason for the poor performance of linear approximation. However, the residual error does not give us the total error but it gives the *minimum* possible error because there also exist both measurement noise and process uncertainties.

A perfect fit also includes all those errors as if they were part of the systems dynamics; this will result in very high prediction errors for long term. Modeling the system by a more limited range of functions is a popular method to attenuate the noise. If sufficient data were available, the correct comparison should be as follows: We use only a portion of data to train our system and estimate the parameters. Let us denote this first portion of data by $\bar{E}_0, \bar{E}_1, ..., \bar{E}_{l_1-1}$. Then, we test the model with some of the remaining data, e.g., $\bar{E}_{l_1}, \bar{E}_{l_1+1}, ..., \bar{E}_{l_1+l_2-1}$ $(l = l_1 + l_2)$. If the test ends with a reasonable approximation error, we could consider using our model to predict the later behavior. However, available data are very limited for a real test.

References

1. Akhmet M. U., Gebert J., Öktem H., Pickl S. W. and Weber G.-W. (2004) *An improved algorithm for analytical modeling and anticipation of gene expression patterns*, to appear in Journal of Computational Technologies.
2. Aster, R. C., Borchers, B. and Thurber, C. H. (2004) *Parameter Estimation and Inverse Problems*, Academic Press.
3. Baldi, P. and Hatfield, G. W. (2002) *DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling*, Cambridge University Press.
4. Chen, T. and He, H. L. (1999) *Modeling gene expression with differential equations*, PSB 4, pp:29-40.
5. de Hoon, M. J. L., Imoto, S. and Miyano, S. (2002) *Inferring gene regulatory networks from time-ordered gene expression data using differential equations*, Discovery Sciences, 2534, pp:267-274, Springer-Verlag.
6. Diestel, R. (1997) *Graph Theory*, New York, Springer.
7. Dijkstra, E. W. (1959) *A note on two problems in connection with graphs*, Numerical Math. 1, pp:269-271.
8. Gebert, J., Lätsch, M., Pickl, S.W., Weber, G.-W. and Wünschiers, R. (2004) *Genetic networks and anticipation of gene expression patterns*, CASYS'03 - Sixth International Conference, 718, pp:474-485.
9. Hastie, T., Tibshirani, R. and Friedman, J. (2001) *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer.
10. Sakamoto, E. and Iba, H. (2001) *Inferring a system of differential equations for a gene regulatory network by using genetic programming*, Proc. Congress on Evolutionary Computation, pp:720-726.
11. Yılmaz, F.B. (2004) *A Mathematical Modeling and Approximation of Gene Expression Patterns by Linear and Quadratic Regulatory Relations*, METU, IAM, MSc Thesis.

On the Empirical Linkages between Stock Prices and Trading Activity on the German Stock Market

Roland Mestel[1], Henryk Gurgul[2], Paweł Majdosz[3]

[1] Institute of Banking and Finance, University of Graz, Austria; roland.mestel@uni-graz.at
[2] Department of Applied Mathematics, University of Science and Technology, Cracow, Poland; hgurgul@neostrada.pl
[3] Department of Econometrics, School of Economics and Computer Science, Cracow, Poland; pmajdosz@go2.pl

Abstract In this study the joint dynamics between stock prices and trading volume are investigated using data from the German stock market. Our results indicate no relations (contemporaneous as well as dynamic) between return levels and trading volume but strong linkages between return volatility and volume data. On including trading volume in the conditional volatility framework (GARCH-type) we provide empirical evidence for the importance of volume data as an indicator for the flow of information on the market. Applying Granger's test for causality we detect also feedback relations between trading volume and return volatility. These findings corroborate our assumption that trading volume indirectly contains information about stock prices due to its relation to return volatility.

1 Introduction

Empirical investigations on stock markets traditionally focus primarily on stock prices and their behavior over time. Conditional upon the available set of information about a company, its stock price reflects investors' expectations concerning the future performance of the firm. The arrival of new information causes investors to adapt their expectations and is the main source for price movements.

However, since investors are heterogeneous in their interpretations of new information, prices may remain unchanged even though new information is revealed to the market. This will be the case if some investors interpret the news as good whereas others find it to be bad. Another situation in which relevant information may leave stock prices unchanged can occur when investors interpret the information identically but start with diverse expectations. From this it follows that changes in stock prices reflect an aggregation or averaging of investors' adapted beliefs.

On the other hand, stock prices may only change if there is positive trading volume. One important question arising from this context is whether volume data are simply a descriptive parameter of the trading process or may contain unique information that can be exploited for modeling stock returns or return volatilities. As with prices, trading volume mainly reflect the available set of relevant information on the market. Unlike stock prices, however, a revision in investors' expectations always leads to an increase in trading volume which therefore reflects the sum of investors' reactions to news. This summation process leading to trading volume preserves differences existing between investors' reactions to the arrival of new information, differences which may get lost in the averaging process that fixes prices. Studying the joint dynamics of stock prices and trading volume therefore improves the understanding of the dynamic properties of stock markets.

Based on the above, a considerable body of literature has emerged which examines the role of trading volume in return formation. Karpoff (1987), Hiemstra and Jones (1994), Brailsford (1996) and Lee and Rui (2002) investigated the relationship between trading volume and price changes per se, mainly using index data. While the results of these studies do differ, on the whole they support the existence of a positive volume-price relationship. The linkage between stock return volatility and trading volume was pointed out, among others, by Karpoff (1987), Brock and LeBaron (1996), and Lee and Rui (2002). These studies uniformly report a strong relationship (contemporaneous as well as dynamic) between return volatility and trading volume. Lamoureux and Lastrapes (1990) were the first to apply stochastic time series models of conditional heteroscedasticity to explore the contemporaneous relationship between volatility and volume data for the US market. The authors find the persistence in stock return variance to vanish for the most part when trading volume is included in the conditional variance equation.

The present study concentrates on the role of trading volume in the process that generates stock returns and return volatilities on the German stock market, namely the stocks of companies listed in the DAX of Deutsche Börse. Unlike most other studies on this issue, we use individual stock data instead of index data. In addition, our investigations cover not only contemporaneous but also dynamic (causal) relationships. This is important, since we are mainly interested in whether trading volume acts as a determinant of stock return levels and/or return volatilities.

The paper is organized as follows: Section 2 outlines the data and provides some basic statistics. Section 3 describes our econometric approach to evaluate the contemporaneous relationship between return series and trading volume. Section 4 extends the analysis by examining dynamic (causal) relations. Section 5 finally summarizes the main results.

2 Data Sample and Basic Statistics

2.1 Data and Sample Period

Our data set comprises daily stock price and trading volume series for all companies listed in the DAX of Deutsche Börse. DAX measures the performance of the 30 largest German companies in terms of order book volume and market capitalization. It is based on prices generated in Xetra. All time series are derived from Reuters. The investigation covers the period January 1999 to November 2003. To avoid our results being unduly influenced by observations surrounding the terrorist attacks on September 11, we exclude from our sample the days from September 11th to September 14th, 2001. The companies were selected on the basis of their being DAX members on November 30th, 2003. We included all those companies that had been quoted in the index for at least 200 trading days over the period under study. For the 30 companies comprising the DAX at our target date, only Conti did not fulfill our terms of admission, reducing our sample to 29 companies.

Continuously compounded stock returns are calculated from daily stock prices at close, adjusted for dividend payouts and stock splits. To proxy return volatility we use squared values of daily percentage stock returns. To measure trading volume the daily turnover ratio is computed for each company. This is defined as the daily number of shares traded divided by the total number of shares outstanding.

2.2 Basic Statistics

We start our investigation with some basic descriptive analysis of the time series of percentage stock returns and trading volume. The mean daily stock return over the period under study ranges from –0.209% (Lufthansa) to +0.093% (Altana) with a median of –0.016%. Standard deviation is lowest for Deutsche Börse (0.017%) and highest for semiconductor producer Infineon (0.044%).

The stylized fact of a 'fat-tailed and highly-peaked' distribution, widely reported for return series, is mostly present in our data. The median of stock return kurtosis is 5.45 and ranges from 24.68 (Bayer) to 3.64 (Siemens). Return skewness is highest for Bayer (1.05) and lowest for Deutsche Börse (–1.09) with a median of 0.07. Applying Jarque-Bera and chi-square goodness-of-fit tests for normality we additionally find strong support for the hypothesis that our return time series do not come from a normal distribution. Concerning autocorrelation properties, the Ljung-Box Q-test statistics for 15th order autocorrelation provide evidence of no significant serial correlation in about 60% of all cases. The remaining return series show significant low-order autocorrelation.

Unlike stock returns, both return volatility and trading volume commonly display strong persistence in their times series. By means of Ljung-Box Q(15)-

statistics we find strong support for the hypothesis that trading volume (turnover ratio) exhibits serial autocorrelation. In accordance with the stylized facts of volume series listed by Andersen (1996) our volume data show high non-normality (positive excess kurtosis and skewed to the right). In addition, we find that log-values of trading volume can be assumed to follow a normal distribution.

To proxy return volatility we use squared values of daily stock returns. These time series display the usual time dependency of stock returns in the second order moment (volatility persistence) implying, among other things, that returns cannot be assumed to be i.i.d. As for trading volume, the null hypothesis of squared returns coming from a normal distribution is strongly rejected.

3 Contemporaneous Relationship

3.1 Stock Returns and Trading Volume

The empirical procedure in this section tests the contemporaneous relations between stock returns R_t and trading volume V_t. We apply a multivariate simultaneous equation model defined by the two structural equations:

$$R_t = \alpha_0 + \alpha_1 \cdot V_t + \alpha_2 \cdot R_{t-1} + \varepsilon_t ; \tag{3.1}$$

$$V_t = \beta_0 + \beta_1 \cdot R_t + \beta_2 \cdot V_{t-1} + \beta_3 \cdot V_{t-2} + \xi_t . \tag{3.2}$$

The model parameters are represented by α_i and β_j while ε_t and ξ_t denote white noise variables. When estimating the structural coefficients, one has to take into account that the joint determined endogenous variables in each equation are not independent of the disturbances. Therefore we apply Full-Information Maximum Likelihood (FIML) methodology. FIML generates asymptotically efficient estimators and has the additional advantage that the cross-equation correlations of the error terms are also taken into consideration.

The findings corroborate our expectations of no contemporaneous relationship between stock return levels and trading volume. Across the whole sample, the parameter α_1 (equation (3.1)) is statistically significant at the 5% level in only 3 cases, whereas parameter β_1 (equation (3.2)) turns out to be insignificant in all cases. Since the majority of our return series exhibit no serial correlation we find parameter α_2 in equation (3.1) to be significant in only 5 cases. The strong time dependence in trading volume time series is clearly supported by the highly significant values found for the parameters β_2 (20 cases) and β_3 (14 cases) in equation (3.2). As one can expect, the sign of these coefficients is always positive, implying positive autocorrelation in volume data.

Although we find stock return levels and trading volume to be independent that does not mean that no relationships can be found in these market data at all.

It is often reported that price fluctuations tend to increase in the case of high trading volume. Thus, there might exist a relation between higher order moments of stock returns and trading volume. In addition, we might ask whether this volatility-volume relation is the same irrespective of the direction of the price change or whether trading volume might be accompanied mainly by a large rise or fall in stock prices.

We investigate this by use of a bivariate regression model which relates trading volume to squared stock returns by the following equation:

$$V_t = \alpha_0 + \phi_1 \cdot V_{t-1} + \phi_2 \cdot V_{t-2} + \alpha_1 \cdot R_t^2 + \alpha_2 \cdot D_t \cdot R_t^2 + \varepsilon_t \,. \tag{3.3}$$

In this model D_t denotes a dummy variable that equals 1 where the corresponding return R_t is negative, and is equal to 0, where this is not the case. Note that the estimate of parameter α_1 measures the relationship between return volatility and trading volume irrespective of the direction of the price change. The estimate of α_2, however, measures the degree of asymmetry in the relationship. To avoid the problem of serially correlated residuals, as documented in Brailsford (1996), we include lagged values of V up to lag 2. After this, we find the error term ε_t in equation (3.3) to be largely uncorrelated.

Applying the ML method to estimate equation (3.3) we find parameter ϕ_1 to be significant across the whole sample and parameter ϕ_2 to be significant in 28 out of 29 cases. In addition, we find parameter α_1 to be positive and significant for all companies, supporting our earlier hypothesis of a strong contemporaneous relationship between R and V. However, parameter α_2 turns out to be significant in only 10 cases. Among these, α_2 has a positive (negative) sign in 6 (4) cases. We conclude that for our sample, heavy price fluctuations are always accompanied by an increase in trading volume irrespective of the direction of the price change.

3.2 Conditional Volatility and Trading Volume

The finding of a contemporaneous relationship between trading volume and squared stock returns raises the question whether trading activity can be identified as one potential source for the observed serial dependence (persistence) in return volatility. This is motivated by the theoretical work on the *Mixture of Distribution Hypothesis (MDH)* (e.g. Clark (1973); Epps and Epps (1976); Lamoureux and Lastrapes (1990); Andersen (1996)). This hypothesis states that stock returns are generated by a mixture of distributions in which the number of information arrivals into the market represents the stochastic mixing variable. Return data can be viewed as a stochastic process, conditional on the information flow, with a changing second order moment reflecting the intensity of information arrivals. Under the assumptions of the MDH model, innovations in the information process lead to momentum in stock return volatility. Since the flow of information into the market is widely unobservable, trading volume is suggested to be an appropriate proxy. Variations in trading volume therefore are assumed to be caused solely by the arrival of new information.

We specify the stochastic process of stock returns by means of an augmented Market Model with an autoregressive term of order 1 in the conditional mean equation (equation 3.4). The conditional variance (equation 3.5) is modeled as a modified GARCH (1,1) process as proposed by Glosten et al. (1993) with trading volume as an additional predetermined regressor. Contrary to simple GARCH models, this model captures the asymmetric (leverage) effect that occurs when good news (manifested in an unexpected increase in prices) induce a decrease in predictable future volatility whereas negative innovations to returns (bad news) cause volatility to rise.

Our model is represented by the following two equations:

$$R_t = \alpha + \beta\cdot R_{m,t} + \phi\cdot R_{t-1} + \varepsilon_t\,; \tag{3.4}$$

$$\sigma_t^2 = \omega + \beta_1\cdot\sigma_{t-1}^2 + \beta_2\cdot\varepsilon_{t-1}^2 + \beta_3\cdot S_{t-1}^{-}\cdot\varepsilon_{t-1}^2 + \gamma\cdot V_t\,, \tag{3.5}$$

where ε_t is assumed to be distributed as N(0,σ_t^2) conditioned on the set of information available at t–1, σ_t^2 stands for the conditional variance of ε_t, and S_{t-1}^{-} is a dummy variable which takes a value of 1 in the case of the innovation ε_{t-1} being negative, and 0 otherwise. The parameters of equations (3.4) and (3.5) are estimated by means of ML method. Note that in (3.5) the sum of parameters β_1 and β_2 is a measure of the persistence in the variance of the unexpected return ε_t taking values between 0 and 1. The more this sum tends to unity the greater the persistence of shocks to volatility (volatility clustering). β_3 takes into account potential asymmetries in the relation between return innovation and conditional volatility.

We first estimate the parameters of interest in equation (3.5) under the assumption that γ is equal to 0 (restricted variance equation). From this we find parameter β_1 to be significant in all but 4 cases and β_2 to be significant across the whole sample. For 23 companies the observed sum (β_1+β_2) lies within the range [0.9 - 1], indicating high persistence in conditional volatility. β_3 turns out to be positive and significant in 14 out of 29 cases illustrating the existence of asymmetric reactions of conditional variance to return innovations.

In the next step we are interested in the unrestricted equation for conditional variance. We find parameter γ to be positive and highly significant in all but 3 cases. Our data demonstrate a considerable decrease in the persistence of volatility when trading volume is included in equation (3.5). The sum of parameters β_1 and β_2 on average decline by about 70%. In only 5 cases, (β_1+β_2) remains within the range [0.9 - 1]. That means that trading volume as a proxy for the flow of information mostly rids the ARCH effect in stock returns. Especially parameter β_2 turns out to be mostly insignificant when contemporaneous trading volume is considered. Concerning the asymmetric effect, we find parameter β_3 to remain significant in the unrestricted version of conditional variance in 9 cases.

4 Causal Relations

So far our investigations have concentrated exclusively on the contemporaneous interactions between trading volume and stock returns, and trading volume and return volatility. In this section we extend our analysis by examining dynamic (causal) interactions between these variables. Testing for causality is important since this can help to better understand the dynamics of stock markets and can also have implications for other markets (e.g. options markets).

We investigate the question of causal relations by means of Granger's test for causality (Granger (1969)). A variable Y is said to not Granger-cause a variable X if the distribution of X, conditional on past values of X alone, equals the distribution of X, conditional on the past of both X and Y. On the other hand, if this equality does not hold, Y is said to Granger-cause X, denoted by $Y \overset{G.c.}{\rightarrow} X$. However, this does not mean that Y causes X in the more common sense of the term but only indicates that Y precedes X. To implement a test for Granger causality we apply a bivariate vector autoregression (VAR) of the form:

$$R_t = \mu_R + \textstyle\sum_{i=1}^{p} \alpha_i \cdot R_{t-i} + \sum_{i=1}^{p} \beta_i \cdot V_{t-i} + \varepsilon_t , \qquad (4.1)$$

$$V_t = \mu_V + \textstyle\sum_{i=1}^{p} \alpha_i \cdot V_{t-i} + \sum_{i=1}^{p} \beta_i \cdot R_{t-i} + \xi_t . \qquad (4.2)$$

To decide upon the appropriate order p of the VARs we use the corrected R^2 and the Akaike Information Criterion (AIC). The parameters α_i and β_i in equation (4.1) and (4.2) are estimated by use of OLS. The null hypothesis of R (V) not to Granger-cause V (R) implies that β_i (i = 1, ...,p) are all equal to 0. We test the null by use of simple F-test.

Table 1 reports our results of Granger testing for unidirectional causality between return levels and trading volume on the one hand, and squared stock returns and trading volume on the other hand.

Table 1. Number of rejected null hypotheses based on Granger's test for causality between stock returns and trading volume, and between squared stock returns and trading volume

	$R \overset{G.c.}{\rightarrow} V$	$V \overset{G.c.}{\rightarrow} R$	$R^2 \overset{G.c.}{\rightarrow} V$	$V \overset{G.c.}{\rightarrow} R^2$
Sample size: 29 companies	7	1	22	15

Order p in regressions (4.1) and (4.2) equals 2.

In line with our expectations, we find only weak evidence of causality between stock return levels and trading volume in either direction. In 7 (1) cases, stock returns (trading volume) precede trading volume (stock returns). From this, one can conclude that short-run forecasts of current or future stock returns in most cases cannot be improved by knowledge of recent trading volume data and vice versa.

We re-run the bivariate VAR in (4.1) and (4.2) substituting squared values of stock returns for return levels. Table 1 reports strong evidence for the existence of causal relations between these variables, being stronger from R^2 to V than in the opposite direction. Preceding return volatility can be seen as some evidence that new information arrival might follow a sequential rather than a simultaneous process. Concerning the role of trading volume our results indicate that data on trading activity has, at least to some extent, additional explanatory power for subsequent price changes that is independent of price series.

5 Conclusion

This article studies the joint dynamics of daily trading volume and stock returns for German companies listed in the DAX. A main issue is whether volume data are simply a description of trading activities or may contain unique information that can be exploited for modeling stock returns or return volatilities. Our findings indicate that there is no evidence of a contemporaneous relationship between stock return levels and trading volume. By means of Granger causality tests we also find dynamic relations between these data to be mostly negligible. From this, we conclude that short-run forecasts of current or future stock returns cannot be improved by knowledge of recent volume data and vice versa. This is in keeping with the efficient capital market hypothesis.

However, our data display extensive interactions between trading volume and stock price fluctuations. In a first step we find squared stock returns and trading volume to be contemporaneously related, implying that both time series might be driven by the same underlying process. In addition our results provide evidence that this volatility-volume relation is independent of the direction of the observed price change. We extend our investigations to a conditional asymmetric volatility framework in which trading volume serves as a proxy for the rate of information arrival on the market. The results support the suggestions from the Mixture of Distribution Hypothesis that ARCH is a manifestation of the daily time dependence in the rate of new information arrival. Finally, we examine dynamic relations between return volatility and trading volume data, and find strong evidence for the existence of causal relations. On average, causality is stronger from volatility to volume than vice versa. We interpret this as an indication that new information arrival on the market might follow a sequential rather than a simultaneous process.

References

Andersen TG (1996) Return volatility and trading volume: An information flow interpretation of stochastic volatility. Journal of Finance 51 (1): 169-204

Brailsford TJ (1996) The empirical relationship between trading volume, returns and volatility. Accounting and Finance 35 (1): 89-111

Brock WA, LeBaron BD (1996) A dynamic structural model for stock return volatility and trading volume. The Review of Economics and Statistics 78 (1): 94-110

Clark PK (1973) A subordinated stochastic process model with finite variance for speculative prices. Econometrica 41 (1): 135-155

Epps TW, Epps ML (1976) The stochastic dependence of security price changes and transaction volumes: Implications for the mixture-of-distribution hypothesis. Econometrica 44 (2): 305-321

Glosten LR, Jagannathan R, Runkle DE (1993) On the relation between the expected value and the volatility of the nominal excess return on stocks. Journal of Finance 48(5): 1779-1801

Granger C (1969) Investigating causal relations by economic models and cross-spectral methods. Econometrica 37 (3): 424-438

Hiemstra C, Jones JD (1994) Testing for linear and nonlinear Granger causality in the stock price - volume relation. Journal of Finance 49 (5): 1639-1664

Karpoff JM (1987) The relation between price changes and trading volume: A survey. Journal of Financial and Quantitative Analysis 22 (1): 109-126

Lamoureux CG, Lastrapes WD (1990) Heteroscedasticity in stock return data: Volume versus GARCH effects. Journal of Finance 45 (1): 221-229

Lee B-S, Rui OM (2002) The dynamic relationship between stock returns and trading volume: Domestic and cross-country evidence. Journal of Banking and Finance 26 (1): 51-78

Numerical Transform Inversion for Autocorrelations of Waiting Times

Hans Blanc

Tilburg University, Dept. Econometrics & Operations Research,
P.O. Box 90153, 5000 LE Tilburg, The Netherlands (blanc@uvt.nl)

Abstract. The generating function of the autocorrelations of successive waiting times in a stationary M/G/1 or in a stationary GI/M/1 system can be expressed in terms of the probability generating function of the number of customers served in a busy period. The latter function is only implicitly determined as a solution to a functional equation. More explicit expressions have been obtained with the aid of Lagrange's theorem on the reversion of power series, but they involve increasingly higher order derivatives of a function which comprises several Laplace-Stieltjes transforms. A recently discovered substitution method for contour integrals allows the numerical inversion of an implicitly determined generating function without the numerical solution of the functional equation for many complex values.

1 Introduction

Autocorrelations of waiting times in queueing systems are useful in determining the variance of the mean of a sample of successive waiting times and provides an indication of how long a simulation should be run. They also give an indication how long it takes for a given backlog of work to fade away. The generating functions of the autocorrelations of the waiting times in stationary M/G/1 and GI/M/1 systems as determined by Blomqvist [5] and Pakes [8], respectively, involve the probability generating function (PGF) of the distribution of the number of customers served in a busy period. The latter functions are only implicitly determined as solutions to functional equations. Standard methods for the numerical inversion of generating functions require the values of these functions at many complex arguments, cf. Abate & Whitt [1,2]. A recent substitution method for contour integrals described in Blanc [3,4] allows the numerical inversion of implicitly determined generating functions without the need for numerical solution of the functional equations. This provides an efficient way to compute the autocorrelations of the waiting times for these systems.

Section 2 provides a short summary of a standard method for the numerical inversion of generating functions. Section 3 contains some general properties of waiting times in stationary GI/G/1 systems and introduces some notations. Section 4 deals with the derivation of an alternative contour integral for the numerical inversion of the GF of the series of autocorrelations of the waiting times in stationary M/G/1 systems. Section 5 is devoted to a similar substitution, but for stationary GI/M/1 systems. The latter two sections contain several examples. Some final remarks on the accuracy of the method can be found in Sect. 6.

2 Numerical inversion of generating functions

The terms of a sequence of real numbers $\{g_k;\ k=0,1,2,\ldots\}$ with $|g_k| \le 1$ for all k can be recovered from its generating function (GF) by means of a contour integral in the complex plane over a circle around the origin with radius r, $0<r<1$:

$$G(z) \doteq \sum_{k=0}^{\infty} g_k z^k, \quad |z|<1, \qquad g_k = \frac{1}{2\pi i} \oint_{|z|=r} G(z)\, \frac{dz}{z^{k+1}}, \quad k=0,1,\ldots . \tag{1}$$

The contour integral can be converted into an integral over a real interval by means of the substitution $z = re^{iu}$ and by some symmetry properties of the GF: for $0<r<1,\ k=0,1,\ldots,$

$$g_k = \frac{1}{\pi r^k} \int_0^{\pi} [\cos(ku)\ \Re G(re^{iu}) + \sin(ku)\ \Im G(re^{iu})]\, du\ ; \tag{2}$$

here, $i=\sqrt{-1}$ and $\Re z$ ($\Im z$) denotes the real (imaginary) part of a complex number z. The case $k=0$ is simple: $g_0 = G(0)$. For $k>0$, Abate & Whitt [1] describe the following method for evaluating the above type of integrals with a prescribed accuracy of, say, ε. Application of the trapezoidal rule with a step size of π/k to (2) yields, for $k=1,2,\ldots,$

$$g_k \approx \frac{1}{kr^k} \left[\frac{1}{2}\{G(r) + (-1)^k G(-r)\} + \sum_{j=1}^{k-1} (-1)^j\, \Re G(re^{ij\pi/k}) \right], \tag{3}$$

while the prescribed accuracy and an upper bound on the discretization error lead to the choice of $r = \sqrt[2k]{\varepsilon},\ k=1,2,\ldots;$ to avoid roundoff problems, approximately $\frac{3}{2}\gamma$-digit precision is required to obtain $\varepsilon = 10^{-\gamma}$ accuracy. The numerical experiments for this paper have been performed with $\gamma = 10$ and about 16-digit precision.

3 Successive waiting times

The starting point for the study of the autocorrelations of waiting times is Lindley's relation for the waiting times of two successive customers in a general GI/G/1 system with service in order of arrival:

$$W_{k+1} = \max\{0, W_k + B_k - A_{k+1}\}, \quad k=1,2,\ldots; \tag{4}$$

here, W_k denotes the waiting time of the kth customer, B_k the service time of the kth customer and A_{k+1} the interarrival time between the kth and the $(k+1)$st customer, after a tagged customer 0. The aim of this paper is the study of the stationary k-step autocorrelations defined by

$$\hat{\varrho}_k\{W\} \doteq \left[E\{W_k W_0\} - E^2\{W\}\right] \Big/ \sigma^2\{W\}, \quad k = 0, 1, 2, \ldots; \tag{5}$$

here, $E\{W\}$ and $\sigma^2\{W\}$ denote the mean and the variance of the stationary distribution of a waiting time W.

Daley [7] proved that the sequence $\hat{\varrho}_k\{W\}$ monotonically converges to zero as $k \to \infty$ for all GI/G/1 systems with $E\{B_k^3\}$ finite. Blomqvist [6] proved the following heavy-traffic limits:

$$\lim_{\rho \uparrow 1} (1-\rho)^2 \sum_{k=1}^{\infty} \hat{\varrho}_k\{W\} = C_A^2 + C_B^2, \tag{6}$$

$$\lim_{\rho \uparrow 1} \frac{1 - \hat{\varrho}_k\{W\}}{(1-\rho)^2} = \frac{2k}{C_A^2 + C_B^2}, \quad k = 1, 2, \ldots, \tag{7}$$

with $\rho \doteq E\{B_k\}/E\{A_k\} < 1$ the load of the system, and C_A (C_B) the coefficient of variation of the interarrival (service) time distribution.

4 The M/G/1 system

Consider an M/G/1 system with arrival rate λ and with $\beta(\zeta)$ the Laplace-Stieltjes transform (LST) of the service time distribution. The PGF of the distribution of the number of customers served in a busy period, J, satisfies the following functional equation:

$$E\{z^J\} = \nu(z), \qquad \nu(z) = z\beta(\lambda[1 - \nu(z)]), \quad |z| \leq 1. \tag{8}$$

Blomqvist [5] has derived the following relation for the GF of the series of autocorrelations of successive waiting times in stationary M/G/1 systems:

$$\sum_{k=0}^{\infty} \hat{\varrho}_k\{W\} z^k = \frac{1}{1-z}\left[1 - \frac{(1-\rho)zE\{W\}}{\lambda(1-z)\sigma^2\{W\}} + \frac{C(z)}{\lambda\sigma^2\{W\}}\right], \quad |z| < 1; \tag{9}$$

here, the function $C(z)$ and its power-series expansion are defined by

$$C(z) \doteq \sum_{k=1}^{\infty} C_k z^k = -\frac{\nu(z)\Omega'(\lambda[1 - \nu(z)])}{1 - \nu(z)}, \quad |z| < 1, \tag{10}$$

and the LST of the waiting time distribution, $\Omega(\zeta)$, is given by the Pollaczek-Khintchine formula

$$\Omega(\zeta) \doteq E\{e^{-\zeta W}\} = \frac{(1-\rho)\zeta}{\zeta - \lambda[1 - \beta(\zeta)]}, \quad \Re\zeta \geq 0. \tag{11}$$

Daley [7] argues that by term by term inversion of the GF (9) and by applying Lagrange's Expansion Theorem, the k-step autocorrelation of waiting times in M/G/1 systems can be written as:

$$\hat{\varrho}_k\{W\} = 1 - \frac{k(1-\rho)E\{W\} - \sum_{j=1}^{k} C_j}{\lambda\sigma^2\{W\}}, \quad k = 1, 2, \ldots, \tag{12}$$

with, for $k = 1, 2, \ldots,$

$$C_k = \frac{1}{k!}\frac{d^{k-1}}{d\zeta^{k-1}}\left\{[-\lambda\beta(\zeta)]^k\left[\frac{\lambda}{\zeta^2}\,\Omega'(\zeta) + \frac{\zeta-\lambda}{\zeta}\,\Omega''(\zeta)\right]\right\}\Bigg|_{\zeta=\lambda}. \tag{13}$$

Clearly, the evaluation of these coefficients become more and more cumbersome as k increases. Only for the case of an M/M/1 system, Daley [7] has found the following general explicit expressions: for $k = 1, 2, \ldots,$

$$C_k = \frac{1-\rho}{\lambda}\left\{k\rho - (1+\rho)\sum_{n=1}^{k}\frac{k+1-n}{n}\binom{2n-2}{n-1}\left[\frac{\rho}{(1+\rho)^2}\right]^n\right\}. \tag{14}$$

Alternatively, contour integrals can be used to determine the coefficients of the function $C(z)$, cf. (1), (10): for $k = 1, 2, \ldots,$

$$C_k = \frac{1}{2\pi i}\oint_{|z|=r} C(z)\,\frac{dz}{z^{k+1}} = \frac{-1}{2\pi i}\oint_{|z|=r}\frac{\nu(z)\Omega'(\lambda[1-\nu(z)])}{1-\nu(z)}\,\frac{dz}{z^{k+1}}. \tag{15}$$

To avoid the computation of the PGF $\nu(z)$ at the values $re^{ij\pi/k}$ by iterative solution of (8) when approximating the above integral by the trapezoidal rule, cf. (3) and Abate & Whitt [2], we substitute $w = \nu(z)$ in the contour integral as in Blanc [3,4]. Since $\nu'(0) = \beta(\lambda) > 0$, cf. (8), this mapping has an inverse in a neighborhood of the origin. Moreover, it follows from (8) that this inverse is explicitly given by $z = w/\beta(\lambda[1-w])$. Hence, this substitution leads to the representation, for $k = 1, 2, \ldots,$

$$C_k = \frac{-1}{2\pi i}\oint_{|w|=r}\frac{\Omega'(\lambda[1-w])}{1-w}\,\beta^k(\lambda[1-w])\left[1 + \frac{\lambda w\beta'(\lambda[1-w])}{\beta(\lambda[1-w])}\right]\frac{dw}{w^k}. \tag{16}$$

The image of a circle $|z| = r$ under the mapping $w = \nu(z)$ is not a circle but a contour with the origin in its interior. Since the integrand in the w-plane has no singularities in $\Re w < 1$ other than $w = 0$, this contour can be replaced by a circle $|w| = r$ by Cauchy's Theorem. The equivalence of (16) with (13) follows by integration by parts and then substitution of $w = 1 - \zeta/\lambda$.

Figure 1 shows the autocorrelations $\hat{\varrho}_1\{W\}$ (left) and $\hat{\varrho}_2\{W\}$ (right) for stationary M/Γ_Ψ/1 systems as functions of the load ρ for various values of the shape parameter Ψ of the gamma (Erlang) distribution, including the limiting case of an M/D/1 system. The behavior as $\rho \uparrow 1$ is in agreement with (7).

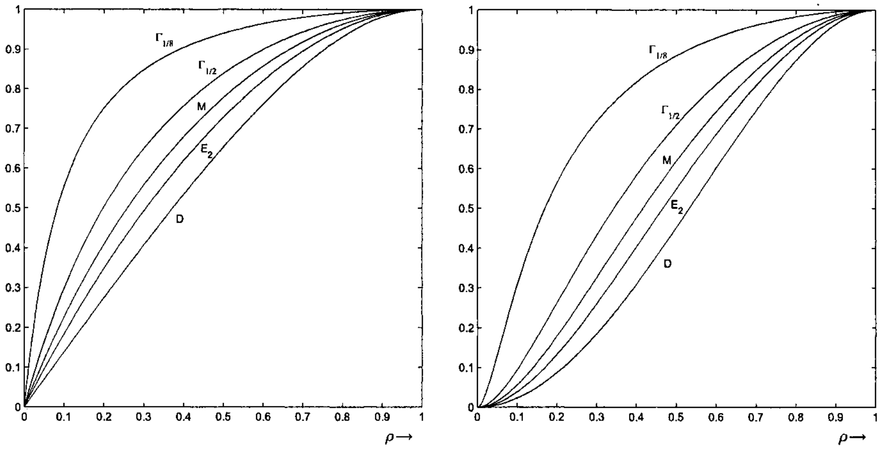


Fig. 1. The 1- and 2-step autocorrelations of the waiting times for $M/\Gamma_\Psi/1$ systems

The behavior of $\hat{\varrho}_k\{W\}$ as $\rho \downarrow 0$ is $O(\rho^k)$ for all distributions, $k = 1, 2$. Table 1 contains autocorrelations $\hat{\varrho}_k\{W\}$ for the same type of systems for a load of $\rho = 0.9$ for various values of k. It clearly shows a strong dependence of $\hat{\varrho}_k\{W\}$ on the coefficient of variation of the service time distribution for larger values of k. Table 1 also contains partial sums and the total sum of the autocorrelations. Daley [7] proved for general M/G/1 systems that if $E\{B_k^4\} < \infty$,

$$\sum_{k=1}^{\infty} \hat{\varrho}_k\{W\} = \frac{\rho}{1-\rho} + \frac{\lambda[E\{W^3\} - E\{W\}E\{W^2\}]}{2(1-\rho)\sigma^2\{W\}}. \tag{17}$$

The table clearly indicates that the autocorrelations vanish much more slowly as $k \rightarrow \infty$ with increasing variability in the service times.

Table 1. Autocorrelations of waiting times for M/G/1 systems with load $\rho = 0.9$

	M/D/1	M/E$_8$/1	M/E$_2$/1	M/M/1	M/$\Gamma_{1/2}$/1	M/$\Gamma_{1/8}$/1
$\hat{\varrho}_1\{W\}$	0.9815	0.9834	0.9873	0.9904	0.9936	0.9979
$\hat{\varrho}_5\{W\}$	0.9178	0.9256	0.9421	0.9554	0.9694	0.9895
$\hat{\varrho}_{10}\{W\}$	0.8511	0.8644	0.8929	0.9163	0.9417	0.9793
$\hat{\varrho}_{100}\{W\}$	0.3134	0.3468	0.4302	0.5127	0.6206	0.8327
$\hat{\varrho}_{1000}\{W\}$	0.0005	0.0009	0.0038	0.0119	0.0403	0.2663
$\sum_{k=1}^{100} \hat{\varrho}_k\{W\}$	55.7786	58.5151	64.9451	70.8584	78.0847	90.9522
$\sum_{k=1}^{1000} \hat{\varrho}_k\{W\}$	92.9737	103.8391	136.0534	177.4536	252.5748	521.7843
$\sum_{k=1}^{\infty} \hat{\varrho}_k\{W\}$	93.0484	104.0009	136.9021	180.8182	268.7079	796.2974

5 The GI/M/1 system

Consider an GI/M/1 system with $\alpha(\zeta)$ the LST of the interarrival time distribution and with service rate μ. The PGF of the distribution of the number of customers served in a busy period, J, is determined via the functional equation:

$$E\{z^J\} = \frac{\mu[z-\chi(z)]}{\mu[1-\chi(z)]}, \qquad \chi(z) = z\alpha(\mu[1-\chi(z)]), \quad |z| \le 1. \tag{18}$$

Pakes [8] has derived the following relation for the GF of the series of autocorrelations of successive waiting times in stationary GI/M/1 systems:

$$\sum_{k=0}^{\infty} \hat{\varrho}_k\{W\} z^k = \frac{1}{1-z}\left[1 - \frac{w_0}{1-w_0^2}\left\{\frac{z[1-w_0-w_0\Xi(z)]}{(1-z)\chi'(1)} - \frac{w_0^2 \Xi(z)}{1-w_0}\right\}\right], \qquad |z|<1; \tag{19}$$

here, the function $\Xi(z)$ and its power-series expansion are defined by

$$\Xi(z) \doteq \sum_{k=1}^{\infty} \Xi_k z^k = \frac{\chi(z)}{1-\chi(z)}, \quad |z| \le 1, \tag{20}$$

and the probability w_0 of zero waiting time is given by $w_0 = 1 - \chi(1)$. From the above GF it follows that, for $k = 1, 2, \dots,$

$$\hat{\varrho}_k\{W\} = 1 - \frac{w_0}{1-w_0^2}\left\{\frac{1-w_0}{\chi'(1)}\left[k - \frac{w_0}{1-w_0}\sum_{j=1}^{k}(k-j)\Xi_j\right] - \frac{w_0^2}{1-w_0}\sum_{j=1}^{k}\Xi_j\right\} \tag{21}$$

with, for $k = 1, 2, \dots,$

$$\Xi_k = \frac{1}{2\pi i}\oint_{|z|=r} \frac{\chi(z)}{1-\chi(z)}\,\frac{dz}{z^{k+1}} = \frac{1}{2\pi i k}\oint_{|w|=r\chi(1)} \frac{\alpha^k(\mu[1-w])}{(1-w)^2}\,\frac{dw}{w^k}. \tag{22}$$

The second integral follows from the first one by substituting $w = \chi(z)$, with inverse $z = w/\alpha(\mu[1-w])$, and integration by parts. The image of the circle $|z| = r$ under the mapping $w = \chi(z)$ is replaced by the circle $|w| = r\chi(1) < r$, which yields higher accuracy than the circle $|w| = r$ (see also Blanc [4]).

Figure 2 shows the autocorrelations $\hat{\varrho}_1\{W\}$ (left) and $\hat{\varrho}_2\{W\}$ (right) for stationary Γ_Θ/M/1 systems as functions of the load ρ for various values of the shape parameter Θ of the gamma (Erlang) interarrival time distribution, including the limiting case of an D/M/1 system. The behavior as $\rho \uparrow 1$ is in agreement with (7). The behavior of $\hat{\varrho}_k\{W\}$, $k = 1, 2$, as $\rho \downarrow 0$ depends on the shape of the interarrival time distribution and differs from that for the corresponding M/Γ_Θ/1 system. Table 2 contains autocorrelations $\hat{\varrho}_k\{W\}$ for the same type of systems for a load of $\rho = 0.9$ for various values of k. It also

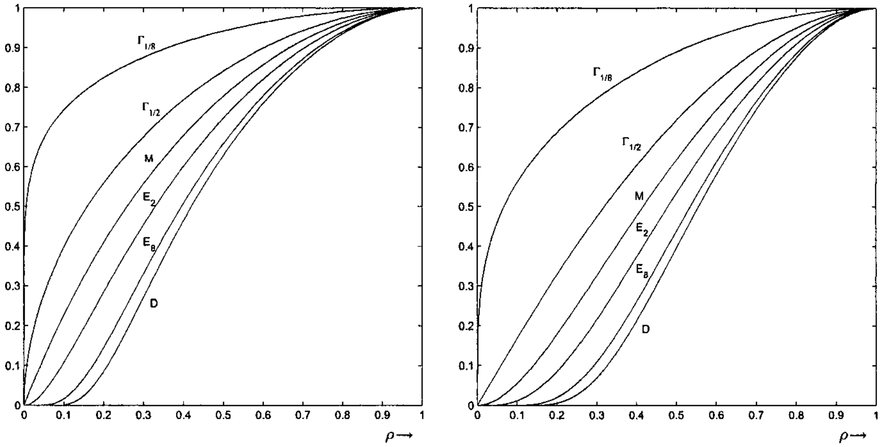


Fig. 2. The 1- and 2-step autocorrelations of the waiting times for Γ_Θ/M/1 systems

contains partial sums and the total sum of the autocorrelations. Pakes [8] proved that for general GI/M/1 systems,

$$\sum_{k=1}^{\infty} \hat{\varrho}_k\{W\} = \frac{1}{w_0[1+\mu\alpha'(\mu w_0)]} - 1 + \frac{\mu^2\alpha''(\mu w_0)}{2(1+w_0)[1+\mu\alpha'(\mu w_0)]^2}. \tag{23}$$

Tables 1 and 2 indicate that the variability in the arrival process and that in the service times have a comparable influence on the autocorrelations for a load ρ close to 1, in agreement with (6) and (7).

Finally, Fig. 3 shows the autocorrelations $\hat{\varrho}_k\{W\}$ for the stationary M/M/1 system as functions of the load ρ for various values of k, illustrating the slower convergence to zero of this sequence as the load ρ increases.

Table 2. Autocorrelations of waiting times for GI/M/1 systems with load $\rho = 0.9$

	D/M/1	$\mathrm{E_8}$/M/1	$\mathrm{E_2}$/M/1	M/M/1	$\Gamma_{1/2}$/M/1	$\Gamma_{1/8}$/M/1
$\hat{\varrho}_1\{W\}$	0.9823	0.9840	0.9875	0.9904	0.9934	0.9978
$\hat{\varrho}_5\{W\}$	0.9182	0.9260	0.9423	0.9554	0.9693	0.9892
$\hat{\varrho}_{10}\{W\}$	0.8497	0.8634	0.8927	0.9163	0.9417	0.9791
$\hat{\varrho}_{100}\{W\}$	0.2978	0.3347	0.4252	0.5127	0.6244	0.8360
$\hat{\varrho}_{1000}\{W\}$	0.0004	0.0007	0.0034	0.0119	0.0426	0.2796
$\sum_{k=1}^{100} \hat{\varrho}_k\{W\}$	54.7233	57.7258	64.6466	70.8584	78.2723	91.0793
$\sum_{k=1}^{1000} \hat{\varrho}_k\{W\}$	88.0303	99.5006	133.5981	177.4536	256.5856	532.0567
$\sum_{k=1}^{\infty} \hat{\varrho}_k\{W\}$	88.0754	99.6138	134.3489	180.8182	273.9847	834.2013

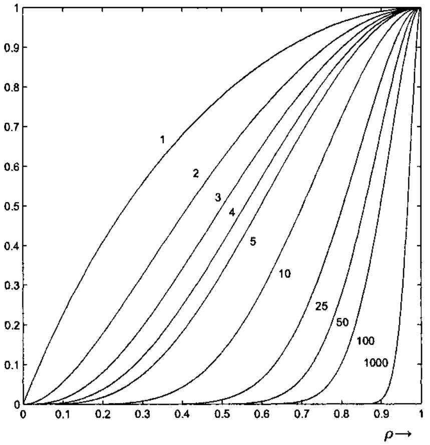


Fig. 3. The k-step autocorrelations of the waiting times for M/M/1 systems

6 Conclusion

This paper has shown that autocorrelations of waiting times can be efficiently computed for M/G/1 and GI/M/1 systems by numerical inversion of the GF after a substitution. The accuracy has been checked by comparing the results for the M/M/1 system obtained by the M/G/1 and the GI/M/1 approach. The absolute difference in the partial sum of the first 1000 autocorrelations for $\rho = 0.9$ is $5 \cdot 10^{-7}$ for the M/G/1 approach and $2 \cdot 10^{-6}$ for the GI/M/1 approach compared with computations based on the explicit expression (14).

References

1. Abate, J., W. Whitt. Numerical inversion of probability generating functions, *Oper. Res. Lett.* **12** (1992), 245–251.
2. Abate, J., W. Whitt. Solving probability transform functional equations for numerical inversion, *Oper. Res. Lett.* **12** (1992), 275–281.
3. Blanc, J.P.C. On the numerical inversion of busy-period related transforms, *Oper. Res. Lett.* **30** (2002), 33–42.
4. Blanc, J.P.C. Computation of autocorrelations of interdeparture times by numerical transform inversion, *Annals Oper. Res.* **112** (2002), 83–100.
5. Blomqvist, N. The covariance function of the M/G/1 queueing system, *Skand. Aktuarietidskr.* **50** (1967), 157–174.
6. Blomqvist, N. Estimation of waiting time parameters in the GI/G/1 queueing system, part II: heavy traffic approximations, *Skand. Aktuarietidskr.* **52** (1969), 125–136.
7. Daley, D.J. The serial correlation coefficients of waiting times in a stationary single server queue, *J. Austral. Math. Soc.* **8** (1968), 683–699.
8. Pakes, A.G. The serial correlation coefficients of waiting times in the stationary GI/M/1 queue, *Ann. Math. Statist.* **42** (1971), 1727–1734.

A Note on the Relationship between Strongly Convex Functions and Multiobjective Stochastic Programming Problems

Vlasta Kaňková

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4, 182 08 Praha 8, Czech Republic
e-mail: kankova@utia.cas.cz

Abstract. We consider multiobjective optimization problems in which objective functions are in the form of mathematical expectation of functions depending on a random element and a constraints set can depend on a probability measure. An efficient points set characterizes the multiobjective problems very often instead of the solution set in one objective case. A stability of the efficient points set (w.r.t. a probability measures space) and empirical estimates have been already investigated in the case when all objective functions were assumed to be strongly convex. The aim of the contribution is to present a modified assertions under rather weaker assumptions.

1 Introduction

Multiobjective optimization problems in which objective functions are in the form of mathematical expectation of functions depending on a random element and a constraints set can depend on the probability measure correspond to many economic activities (see e.g. [8]). Namely very often economic and social phenomena are influenced by random factors and, simultaneously, it is reasonable to evaluate them with several objective functions. We introduce the above mentioned problem in the following form.

Find

$$\inf \mathsf{E}_{F^\xi} g_i(x, \xi), \quad i = 1, 2, \ldots, l \quad \text{subject to } x \in \mathcal{K}_{F^\xi} \tag{1}$$

where g_i, $i = 1, \ldots, l$ are functions defined on $R^n \times R^s$, ξ is an s–dimensional random vector, F^ξ and P_{F^ξ} denote the distribution function and the probability measure of ξ; $\mathcal{K}_{F^\xi} \subset R^n$ is a nonempty set that generally can depend on F^ξ. (R^n, $n \geq 1$ denotes n–dimensional Euclidean space.)

Different approaches to the multiobjective problems have been introduced in the literature (see e.g. [2], [4], [15]). A vector of the objective functions is there often replaced by one "suitable" objective function (see e.g. the Markowitz model for a portfolio selection). However, evidently, an efficient points set is an essential characterization of the multiobjective problem everywhere.

A complete information on the probability measure P_{F^ξ} is a necessary assumption to treat the problem (1). In applications this assumption is fulfilled very seldom, consequently, very often P_{F^ξ} has to be replaced by its approximation. However, to replace responsible P_{F^ξ} by some approximation it is necessary to deal with a stability (considered w.r.t. a probability measure space) and with corresponding statistical estimates. To this end a generalization of the Markowitz approach (known as the weight approach) has been employed. We define the corresponding problem by the relations.

Find

$$\inf\{\mathsf{E}_{F^\xi} g^\lambda(x,\, \xi) | x \in \mathcal{K}_{F^\xi}\}, \tag{2}$$

$$g^\lambda(x,\, z) = \sum_{i=1}^{l} \lambda_i g_i(x,\, z), \quad \lambda \in \Lambda, \quad x \in X, \quad z \in R^s,$$

$$\Lambda \qquad = \{\lambda \in R^l : \lambda = (\lambda_1,\, \dots,\, \lambda_l),\ \lambda_i \in \langle 0,\, 1\rangle,\ i = 1,\, \dots l;\ \sum_{i=1}^{l} \lambda_i = 1\}.$$

The problem (2) is one–objective (parametric) optimization problem. It follows from the theory of the deterministic multiobjective problems that (under additional assumptions) there exists a relationship between the (properly) efficient points of the problem (1) and a solutions set of the problem (2). Consequently, to obtain assertions on the stability and empirical estimates concerning the problem (1) the results achieved for one–objective problems can be employed. To recall suitable results for one–objective case we can mention e.g. the papers [3], [6], [7], [14], [16]. However, to obtain stability results concerning the problem (1), an assumption on strongly convexity of the functions g_i, $i = 1,\, \dots,\, l$ has been supposed (see e.g. [10]). The aim of this contribution is to introduce a modified assertions under rather weaker assumptions.

2 Some Definitions and Auxiliary Assertions

A multiobjective deterministic optimization problem can be introduced as the problem.

Find

$$\min f_i(x),\ i = 1,\, \dots,\, l \quad \text{subject to } x \in \mathcal{K}. \tag{3}$$

f_i, $i = 1,\, \dots,\, l$ are functions defined on R^n, $\mathcal{K} \subset R^n$ is a nonempty set.

Definition 1. The vector x^* is an efficient solution of the problem (3) if and only if there exists no $x \in \mathcal{K}$ such that $f_i(x) \le f_i(x^*)$ for $i = 1,\, \dots,\, l$ and such that for at least one i_0 one has $f_{i_0}(x) < f_{i_0}(x^*)$.

Definition 2. The vector x^* is a properly efficient solution of the multiobjective optimization problem (3) if and only if it is efficient and if there exists a scalar $M > 0$ such that for each i and each $x \in \mathcal{K}$ satisfying $f_i(x) < f_i(x^*)$ there exists at least one j such that $f_j(x^*) < f_j(x)$ and

$$\frac{f_i(x^*) - f_i(x)}{f_j(x) - f_j(x^*)} \leq M. \tag{4}$$

Proposition 1. [5] Let $\mathcal{K}$ be a convex set and let f_i, $i = 1, \ldots, l$ be convex functions on $\mathcal{K}$. Then x^0 is a properly efficient solution of the problem (3) if and only if x^0 is optimal in

$$\min_{x \in \mathcal{K}} f^{\lambda}(x) \quad \text{for some} \quad \lambda \in \Lambda, \quad f^{\lambda}(x) = \sum_{i=1}^{l} \lambda_i f_i(x). \tag{5}$$

Remark 1. Let $l \geq 2$, x_{λ}, $\lambda = (\lambda_1, \ldots, \lambda_l)$, $\lambda \in \Lambda$ be a solution of the problem (5). It follows from the proof of Theorem 1 in [5] that M corresponding $x := x_{\lambda}$ fulfils the relation (4) with $M = (l-1) \max_{i,j} \frac{\lambda_i}{\lambda_j}$. Consequently, omitting the value of the parameter $\lambda = (\lambda_1, \ldots, \lambda_l)$ (in the relation (5)) with a "small" λ_i for at least one $i \in \{1, \ldots, l\}$ we obtain the relationship given by Proposition 1 for all "reasonable" values of the parameter λ.

Lemma 1. Let $\mathcal{K} \subset R^n$ be a nonempty set, $f_i, i = 1, \ldots, l$ be functions defined on R^n. Let, moreover, f^{λ} be defined by the relation (5). We obtain.

1. If $f_i, i = 1, \ldots, l$ are Lipschitz functions on $\mathcal{K}$ with the Lipschitz constants $\bar{L}_i$, then f^{λ}, $\lambda \in \Lambda$ is a Lipschitz function on $\mathcal{K}$.
2. If $f_i, i = 1, \ldots, l$ are bounded functions on $\mathcal{K}$, ($|f_i(x)| \leq M$, $x \in \mathcal{K}$, $i = 1, \ldots, l$), then for every x, f^{λ} is a Lipschitz function on Λ.

Definition 3. Let h be a real–valued function defined on a convex set $\mathcal{K} \subset R^n$. h is a strongly convex function with a parameter $\rho > 0$ if

$$h(\bar{\lambda} x^1 + (1 - \bar{\lambda}) x^2) \leq \bar{\lambda} h(x^1) + (1 - \bar{\lambda}) h(x^2) - \bar{\lambda}(1 - \bar{\lambda}) \rho \|x^1 - x^2\|^2$$

for every x^1, $x^2 \in \mathcal{K}$, $\bar{\lambda} \in \langle 0, 1 \rangle$. ($\| \cdot \|$ denotes the Euclidean norm in R^n.)

Lemma 2. [6] Let $\mathcal{K}$ be a nonempty, compact, convex set. Let, moreover, h be a strongly convex with a parameter $\rho > 0$, continuous function defined on $\mathcal{K}$. If x^0 is defined by the relation $x^0 = \arg \min\limits_{x \in \mathcal{K}} h(x)$, then

$$\|x - x^0\|^2 \leq \frac{2}{\rho} |h(x) - h(x^0)| \quad \text{for every } x \in \mathcal{K}.$$

Lemma 3. Let $\mathcal{K}$ be a nonempty convex set, $\varepsilon \in (0, 1)$. Let, moreover, f_i, $i = 1, \dots, l$ be convex functions on $\mathcal{K}$. If

1. f_1 is a strongly convex (with a parameter $\rho > 0$) on $\mathcal{K}$, then f^{λ} defined by (5) is for $\lambda \in \Lambda$, $\lambda_1 \in (\varepsilon, 1)$ a strongly convex function on $\mathcal{K}$. [2.] f_i, $i = 1, \dots, l$ are strongly convex function on $\mathcal{K}$ with a parameter ρ, then f^{λ} defined by (5) is a strongly convex function on $\mathcal{K}$.

If F and G are two arbitrary s–dimensional distribution functions, then the Kolmogorov metric $d_K(P_F, P_G)$ is defined by

$$d_K(P_F, P_G) := d_K(F, G) = \sup_{z \in R^s} |F(z) - G(z)|. \tag{6}$$

To define the Wasserstein metric $d_{W_1}(P_F, P_G) := d_{W_1}(F, G)$ let $P_F, P_G \in \mathcal{P}(R^s)$ denote the set of (all) Borel probability measures on R^s and let

$$\mathcal{M}_1(R^s) = \{\nu \in \mathcal{P}(R^s) : \int\limits_{R^s} |z| |\nu(dz) < \infty\}.$$

If $\mathcal{D}(\nu, \mu)$ denotes the set of those measures in $\mathcal{P}(R^s \times R^s)$ whose marginal measures are ν and μ, then

$$d_{W_1}(\nu, \mu) = \inf\{ \int\limits_{R^s \times R^s} \|z - \bar{z}\| \kappa(dz \times d\bar{z}) : \kappa \in \mathcal{D}(\nu, \mu)\}, \quad \nu, \mu \in \mathcal{M}_1(R^s). \tag{7}$$

(For more details about the Wasserstein metric see e.g. [13].)

3 Problem Analysis

Evidently the multiobjective stochastic programs (1) can be characterized by the sets $\mathcal{G}_{F^\xi}$, $\mathcal{X}_{F^\xi}$ and $\bar{\mathcal{G}}_{F^\xi}$ (see also [10]), defined by the relations

$$\begin{aligned}
\mathcal{G}_{F^\xi} &= \{y \in R^l : y_j = \mathsf{E}_{F^\xi} g_j(x, \xi),\ j = 1, \dots, l \quad \text{for some}\, x \in \mathcal{K}_{F^\xi};\\
&\quad y = (y_1, \dots, y_l)\},\\
\mathcal{X}_{F^\xi} &= \{x \in R^n : x \text{ is a properly efficient point of the problem (1)}\},\\
\bar{\mathcal{G}}_{F^\xi} &= \{y \in R^l : y_j = \mathsf{E}_{F^\xi} g_j(x, \xi),\ j = 1, \dots, l \quad \text{for some } x \in \mathcal{X}_{F^\xi};\\
&\quad y = (y_1, \dots, y_l)\}.
\end{aligned} \tag{8}$$

If we replace in (1) F^{ξ} by another s–dimensional distribution function G, then we obtain $\mathcal{K}_G$ instead of $\mathcal{K}_{F^\xi}$ in (1) and $\mathcal{G}_G$, $\mathcal{X}_G$ and $\bar{\mathcal{G}}_G$ instead of $\mathcal{G}_{F^\xi}$, $\mathcal{X}_{F^\xi}$, $\bar{\mathcal{G}}_{F^\xi}$ in (9). In [10] upper bounds of the values

$$\Delta[\mathcal{G}_{F^\xi}, \mathcal{G}_G], \quad \Delta[\mathcal{X}_{F^\xi}, \mathcal{X}_G]^2, \quad \Delta[\bar{\mathcal{G}}_{F^\xi}, \bar{\mathcal{G}}_G],$$

were introduced. To obtain these results the functions g_i, $i = 1, \ldots, l$ were assumed to be strongly convex w.r.t. $x \in R^n$. This assumption is rather strong and in many applications not fulfilled. On the other hand very often the functions g_i, $i = 1, \ldots, l$ are convex and, simultaneously, there exists $j \in \{1, \ldots, l\}$ such that g_j is strongly convex on R^n (see e.g. the Markowitz model). Employing Remark 1 and Lemma 1, Lemma 2 and Lemma 3 it seems reasonable to define for $\varepsilon \in (0, 1)$, furthermore, the sets Λ^{ε}, $\bar{\mathcal{X}}^{\varepsilon}_{F^{\xi}}$, $\bar{\mathcal{G}}^{\varepsilon}_{F^{\xi}}$.

$$\begin{aligned}
\Lambda^{\varepsilon} \; &= \{\lambda \in \Lambda : \lambda_i \in (\varepsilon, 1) \text{ for at least one} i \in \{1, \ldots l\}\}, \\
\bar{\mathcal{X}}^{\varepsilon}_{F^{\xi}} &= \{x \in X : x \text{ is a properly efficient point of the problem } (1) \\
&\qquad \text{corresponding to some } \quad \lambda \in \Lambda^{\varepsilon}\}, \\
\bar{\mathcal{G}}^{\varepsilon}_{F^{\xi}} &= \{y \in R^l : y_j = \mathsf{E}_{F^{\xi}} g_j(x, \xi),\, j = 1, \ldots, l \quad \text{for some } x \in \bar{\mathcal{X}}^{\varepsilon}_{F^{\xi}}; \\
&\qquad y = (y_1, \ldots, y_l)\}.
\end{aligned} \tag{9}$$

Remark 2. Evidently, a relationship between the problems (3) and (5) is given by Proposition 1; consequently the relationship between "properly" efficient points of the problem (3) and the solution set of the problem (5) is guaranteed by the assertion of Lemma 2. To apply these assertions to problem (1), g_i, $i = 1, \ldots, l$ has been assumed to be strongly convex w.r.t. x. According to Remark 1 we can see that omitting (properly) efficient points corresponding $\lambda \in \Lambda$, with at least one "small" component, we can obtain stability results corresponding to every "essential" efficient points subset. To obtain this modified assertion only for one $i \in \{1, \ldots, l\}$ the function $g_i(x, z)$ was supposed to be strongly convex w.r.t. x.

4 Main Results

To introduce the modified stability assertions, let $X \subset R^n$ fulfill the relations

$$\mathcal{K}_{F^{\xi}} \subset X, \quad X = \prod_{i=1}^{n} \langle a_i, b_i \rangle \quad \text{for some} \quad a_i, b_i \in R^1, a_i < b_i, . \tag{10}$$

Furthermore, we introduce the following system of the assumptions.

A.1 $\mathcal{K}_{F^{\xi}}$ is a nonempty compact set,
A.2 a. g_i, $i = 1, \ldots, l$ are uniformly continuous functions on $X \times R^s$,
b. g_i, $i = 1, \ldots, l$ are uniformly continuous, bounded on $X \times R^s$,
c. for every $x \in X$, g_i, $i = 1, \ldots, l$ are Lipschitz functions of $z \in R^s$ with the Lipschitz constants not depending on $x \in X$,
A.3 a. $\mathcal{K}_{F^{\xi}}$ is a convex, nonempty, compact set,
b. g_i, $i = 1, \ldots, l$ are convex functions on X; there exists $i \in \{1, \ldots, l\}$ such that g_i is a strongly convex function on X,

c. g_i, $i = 1, \ldots, l$ are strongly convex functions on X,

A.4 for every $z \in Z_{F^\xi} \bigcup Z_G$, g_i, $i = 1, \ldots, l$ are Lipschitz functions on X with the Lipschitz constants not depending on $z \in Z_{F^\xi} \bigcup Z_G$. (

Proposition 2. Let $\varepsilon \in (0, 1)$, $l \geq 2$, the relation (10) be fulfilled. If g_i, $i = 1, \ldots, l$ are bounded on $X \times Z_{F^\xi}$, then there exists $\bar{L}$ such that

$$|\mathsf{E}_{F^\xi} g^{\lambda'}(x, \xi) - \mathsf{E}_{F^\xi} g^{\lambda''}(x, \xi)| \leq \bar{L} \|\lambda' - \lambda''\|, \quad \lambda', \lambda'' \in \Lambda, \quad x \in X.$$

According to Proposition 2 for every $\lambda \in \Lambda$ there exists $\lambda^\varepsilon \in \Lambda^\varepsilon$ such that

$$|\mathsf{E}_{F^\xi} g^{\lambda}(x, \xi) - \mathsf{E}_{F^\xi} g^{\lambda^\varepsilon}(x, \xi)| \leq \bar{L}\, l\, \varepsilon \quad \text{for every } x \in X.$$

Theorem 1. Let $\varepsilon \in (0, 1)$, $P_{F^\xi} \in \mathcal{M}_1(R^s)$, the relation (10) and the assumptions A.1, A.2a, A.2c, A.4 be fulfilled. Let, moreover, G be an arbitrary s–dimensional distribution function such that there exists a constant $\bar{C}_K$ fulfilling the relations

$$P_G \in \mathcal{M}_1(R^s), \quad \mathcal{K}_G \subset X, \quad \Delta[\mathcal{K}_{F^\xi}, \mathcal{K}_G] \leq \bar{C}_K \, [d_K(F^\xi, G)]^{\frac{1}{s}}.$$

then there exist constants $\bar{C}_{\mathcal{G}_{F^\xi}}, C_{\mathcal{G}_{F^\xi}} > 0$ such that

$$\Delta[\mathcal{G}_{F^\xi}, \mathcal{G}_G] \leq C_{\mathcal{G}_{F^\xi}} d_{W_1}(P_{F^\xi}, P_G) + \bar{C}_{\mathcal{G}_{F^\xi}} \, [d_K(F^\xi, G)]^{\frac{1}{s}}. \tag{11}$$

If, moreover,

1. the assumptions A.3a and A.3b are fulfilled, then there exists a constant $C^\varepsilon_{\mathcal{X}_{F^\xi}}$, $\bar{C}^\varepsilon_{\mathcal{X}_{F^\xi}}$, > 0 such that
$$\Delta[\mathcal{X}^\varepsilon_{F^\xi}, \mathcal{X}^\varepsilon_G]^2 \leq C^\varepsilon_{\mathcal{X}_{F^\xi}} d_{W_1}(P_{F^\xi}, P_G) + \bar{C}^\varepsilon_{\mathcal{X}_{F^\xi}}, \; [d_K(F^\xi, G)]]^{\frac{1}{s}}, \tag{12}$$
2. the assumptions A.3a and A.3c are fulfilled, then there exists a constant $C_{\mathcal{X}_{F^\xi}} > 0$, $\bar{C}_{\mathcal{X}_{F^\xi}} > 0$, such that
$$\Delta[\mathcal{X}_{F^\xi}, \mathcal{X}_G]^2 \leq C_{\mathcal{X}_{F^\xi}}(1)\, d_{W_1}(P_{F^\xi}, P_G) + \bar{C}_{\mathcal{X}_{F^\xi}} \, [d_K(F^\xi, G)]]^{\frac{1}{s}}, \tag{13}$$
3. the assumptions A.3a, A.3b are fulfilled, then there exist constants $C^\varepsilon_{\bar{\mathcal{G}}_{F^\xi}}$, $\bar{C}^\varepsilon_{\bar{\mathcal{G}}_{F^\xi}}$, $K^\varepsilon_{\bar{\mathcal{G}}_{F^\xi}} > 0$ such that
$$\begin{aligned}\Delta[\bar{\mathcal{G}}^\varepsilon_{F^\xi}, \bar{\mathcal{G}}^\varepsilon_G] \leq\; & C^\varepsilon_{\bar{\mathcal{G}}_{F^\xi}} \sqrt{d_{W_1}(P_{F^\xi}, P_G)}(K^\varepsilon_{\bar{\mathcal{G}}_{F^\xi}} + \sqrt{d_{W_1}(P_{F^\xi}, P_G)}) + \\ & \bar{C}^\varepsilon_{\bar{\mathcal{G}}_{F^\xi}} [d_K(F^\xi, G)]^{\frac{1}{s}}.\end{aligned} \tag{14}$$

It happens (in applications) rather often that the distribution function F^ξ has to be replaced by an empirical measure. To deal with empirical estimates, let $\{\xi^k\}_{k=1}^N$, $N = 1, 2, \ldots$ be a sequence of independent s–dimensional random vectors with a common distribution function F^ξ. We denote by the symbol F^ξ_N the empirical distribution function determined by $\{\xi^k\}_{k=1}^N$.

Theorem 2. Let $\varepsilon \in (0, 1)$, X, $\mathcal{K}_{F^{\xi}}$ be nonempty, compact sets, $t > 0$, the relation (10) and the assumptions A.1, A.2b, A.4 be fulfilled. Let, moreover, there exist constants $K^1 := K^1(X, t) > 0$, $k^1 > 0$ such that

$$P\{\mathcal{K}_{F^{\xi}_N} \subset X, \quad \Delta[\mathcal{K}_{F^{\xi}}, \mathcal{K}_{F^{\xi}_N}] > t\} \le K^1(X, t) \exp\{-k^1 N t^2\},$$

then there exist constants $K^1_{\mathcal{G}_{F^{\xi}}} := K^1_{\mathcal{G}_{F^{\xi}}}(X, t) > 0$, $k^1_{\mathcal{G}_{F^{\xi}}} > 0$ such that

$$P\{\Delta[\mathcal{G}_{F^{\xi}}, \mathcal{G}_{F^{\xi}_N}] > t\} \le K^1_{\mathcal{G}_{F^{\xi}}}(X, t) \exp\{-k^1_{\mathcal{G}_{F^{\xi}}} N t^2\}, \quad N = 1, 2, \ldots.$$

If, moreover,

1. the assumptions A.3a, A.3b are fulfilled, then there exist constants $K^{\varepsilon}_{\mathcal{X}_{F^{\xi}}} := K^{\varepsilon}_{\mathcal{X}_{F^{\xi}}}(X, t) > 0$, $k^{\varepsilon}_{\mathcal{X}_{F^{\xi}}} > 0$ such that

$$P\{[\Delta[\bar{\mathcal{X}}^{\varepsilon}_{F^{\xi}}, \bar{\mathcal{X}}^{\varepsilon}_{F^{\xi}_N}]^2 > t\} \le K^{\varepsilon}_{\mathcal{X}_{F^{\xi}}}(X, t) \exp\{-k^{\varepsilon}_{\mathcal{X}_{F^{\xi}}} N t^2\}, \quad N = 1, 2, \ldots.$$

2. the assumptions A.3a, A.3c are fulfilled, then there exist constants $K_{\mathcal{X}_{F^{\xi}}} := K_{\mathcal{X}_{F^{\xi}}}(X, t) > 0$, $k_{\mathcal{X}_{F^{\xi}}} > 0$ such that

$$P\{[\Delta[\bar{\mathcal{X}}_{F^{\xi}} \bar{\mathcal{X}}_{F^{\xi}_N}]^2 > t\} \le K_{\mathcal{X}_{F^{\xi}}}(X, t) \exp\{-k_{\mathcal{X}_{F^{\xi}}} N t^2\}, \quad N = 1, 2, \ldots.$$

Remark 3. It happens rather often that there exist a natural numbers s_i, $i = 1, \ldots, l$ and functions $h^*_{i,j}$, $g^*_{i,j}$, $j = 1, \ldots, s_i$, $i = 1, \ldots, l$ defined on R^n and R^s such that

$$g_i(x, z) = \sum_{j=1}^{s_i} h^*_{i,j}(x)\, g^*_{i,j}(z), \quad x \in X, \quad z \in Z_{F^{\xi}}. \tag{15}$$

The form (16) of the functions g_i, $i = 1, \ldots, l$ is very suitable from the applications point if view (see e.g. [11]). Moreover, in this special case the results of Theorem 2 can be (for finite N) rather improved.

Employing the approach of the paper [1] we can easy see that (under some additional assumptions) the exponentional rate convergence can be obtained even in the case when g_i, $i = 1, \ldots, l$ are unbounded functions. However, to deal with this case is over the possibilities of this contribution.

5 Conclusion

The contribution deals with the stability and empirical estimates in the case of multiobjective stochastic programming problems. The new modified results are introduced under relative weak assumptions. These results are based on the theory of the deterministic multiobjective optimization problems and the theory of strongly convex functions. The exact proofs of the presented assertions are omitted as their are over the possibilities of this contribution. However the main idea of the proofs is very similar to the main idea of the proof of the assertions introduced in [10] (for details see [9]). In this place we would like mention only that an assertion introduced in Remark 1 must be included.

Acknowledgement. This research was supported by the Grant Agency of the Czech Republic under Grants 402/02/1015 and 402/04/1294.

References

1. L. Dal, C. H. Chen and J. R. Birge (2000): Convergence properties of two-stage stochastic programming. J. Optim. Theory Appl. **106**, 3, 489–509.
2. Caballero, R., Cerdá, E., Muñoz, M. M., Rey, L., Stancu-Minasian, I. M. (2001): Efficient solution concepts and their relations in stochastic multiobjective programming. J. Opt. Theory and Appl. **110**, 1, 53–74.
3. Dupačová, J., Wets, R, J.-B. (1984): Asymptotic bahaviour of statistical estimates and optimal solutions of stochastic optimization problems. Ann. Statist., **16**, 1517–1549.
4. Dupačová, J., Hurt, J., Štěpán, J. (2002): Stochastic Modeling in Economics and Finance. Kluwer, Dordrecht.
5. Geoffrion, A. M. (1968): Proper efficiency and the theory of vector maximization. J. Math. Anal. Appl., **22**, 3, 618–630.
6. Kaňková, V., Lachout, P. (1992): Convergence rate of empirical estimates in stochastic programming. Informatica, **3**, 4, 497–523.
7. Kaňková, V. (1994): A note on estimates in stochastic programming. J. Comp. Math., **56**, 97–112.
8. Kaňková, V. (1998b): A note on analysis of economic activities with random elements. In: Mathematical Methods in Economy, Cheb 1998 (M. Plevný and V. Friedrich, eds.), University of West Bohemia, Cheb 1999, 53–58.
9. Kaňková, V. (2002): Stability in Multiobjective Stochastic Programming Problems. Research Report UTIA AS CR No 1990.
10. Kaňková, V. (2003): A remark on multiobjective stochastic optimization problems: Stability and empirical estimates. In: Operations Research Proceedings 2003, Springer, Berlin 377–386.
11. Kaňková, V. (2004): Multiobjective programs and Markowitz model. In: Quantitative Methods in Economics 2004 (M. Lukáčik, ed.) The Slovak Society for Operations Research and University of Economics in Bratislava, Bratislava 2004,109–117.
12. Prékopa, A. (1995): Stochastic Programming. Akadémiai Kiadó, Budapest and Kluwer, Dordrecht.
13. Rachev, S T. (1991): Probability Metrics and the Stability of Stochastic Models. Wiley, Chichester.
14. Römisch, W., Schulz, R. (1993): Stability of solutions for stochastic programs with complete recourse. Math. Oper. Res., **18**, 590–609.
15. Stancu-Minasian, I. M. (1984): Stochastic Programming with Multiple Objective Functions. D. Reidel Publishing Company, Dordrecht.
16. Vogel, S. (1992): On stability in multiobjective programming – a stochastic approach. Mathematical programming **50** (1992), 197–236.

Two-Step Drawing from Urns

Stephan Kolassa and Stefan Schwarz

Institut für Angewandte Mathematik, Friedrich-Schiller-Universität Jena, 07740 Jena, Germany, {kolassa,schwarz}@minet.uni-jena.de

Abstract. Consider the following situation of two-step shortlisting: two experts Alice and Bob are faced with a large number of alternatives which they can only observe imprecisely. They have to choose one of the alternatives, without knowing which one is best. Alice first compiles a shortlist of alternatives by choosing her k best observations. Bob then chooses his best observation among the shortlisted alternatives. Previous research showed that this procedure sometimes yielded worse results than if a single expert made the entire decision himself. Here, we consider an urn containing $n-1$ homogeneous balls and one ball with larger weight. When drawing balls at random from the urn, the probability of drawing any one ball is proportional to its weight. Alice draws k balls and puts them in another urn, from which Bob then draws a single ball. Which value of k maximizes the probability that Bob draws the distinguished ball?

1 Introduction

Situations of two-step shortlisting by imperfect experts can be formally characterized as follows: consider two experts Alice and Bob, faced with a large number of alternatives which they can evaluate only imprecisely. Alice chooses a subset of k alternatives, and Bob then chooses one alternative from this *shortlist* compiled by Alice, with the goal of choosing the unknown best one among the original alternatives. One hopes that the imprecisions of the experts cancel each other out to a certain degree, leading to better results than if only a single expert made the decision.

Previous research on two-step shortlisting by imperfect experts yielded surprising results [1,3]. Even if both experts are equally imprecise, as measured by the distributions of stochastic noise terms, the shortlisting can actually lead to worse results than if a single expert made the entire decision alone: a *shortlisting valley* appeared.

Here, we examine a model of sequential drawing from an urn. The objective of Alice and Bob is to draw the heaviest ball. We restrict ourselves to the special case of $n-1$ homogeneous balls and one distinguished ball which is heavier than the others. Initial results for this model have been given in [4].

The probability of drawing the heaviest ball first increases and then decreases with the shortlist size k (Prop. 1). The optimal value of k is largest for small weights of the distinguished ball. It decreases monotonically from $\sqrt{2n}$ to 2 with increasing distinguished weight (Prop. 2 and 3).

2 The Model

We consider an urn $\mathcal{U}$ containing $n \geq 3$ balls, which we identify with the numbers $1, \ldots, n$. Ball 1 has weight $p > 1$, and the other balls have weight 1. Whenever a ball is drawn from an urn, the probability of any particular ball to be drawn is proportional to its weight.

Alice draws k balls at random from $\mathcal{U}$, one after the other, and puts them in a second urn $\mathcal{S}$ (for "shortlist"). Bob then draws a single ball from $\mathcal{S}$.

We are interested in how the probability that Bob draws the distinguished ball (the *hitting ratio*) varies with $k \in \{1, \ldots, n\}$. In particular, we will investigate the value k^* which yields the highest hitting ratio.

3 Unimodality

Proposition 1. *The hitting ratio is unimodal in k, i.e., it first increases and then decreases.*

It turns out that the analysis is facilitated by assuming different "subjective" weights of ball 1 for Alice and Bob: we let $p > 1$ denote the "subjective" weight of ball 1 for Alice as above and $q > 1$ the "subjective" weight for Bob.

Proof. The probability for Alice to pass on the correct ball in $\mathcal{S}$ is the complement of the probability that she passes on only "bad" balls:

$$1 - \frac{n-1}{n-1+p} \cdot \frac{n-2}{n-2+p} \cdot \ldots \cdot \frac{n-k}{n-k+p}.$$

The hitting ratio h for parameters p, k and q therefore is

$$h(n \xrightarrow{p} k \xrightarrow{q} 1) = \left(1 - \frac{n-1}{n-1+p} \cdot \ldots \cdot \frac{n-k}{n-k+p}\right) \cdot \frac{q}{k-1+q}. \tag{1}$$

First, for fixed n, k and p, we compute the value $q_k(p)$ for which

$$h(n \xrightarrow{p} k \xrightarrow{q_k(p)} 1) = h(n \xrightarrow{p} k+1 \xrightarrow{q_k(p)} 1) \quad \text{for } 1 \leq k < n.$$

Some calculations yield that

$$q_k(p) = \frac{(n-1+p)\ldots(n-k+p)(n-k-1+p)}{(n-1)\ldots(n-k)p} - \frac{n-k-1}{p} - k \tag{2}$$

and that

$$q_1(p) < q_2(p) < \cdots < q_{n-2}(p) < q_{n-1}(p).$$

We have $h(n \xrightarrow{p} k \xrightarrow{q} 1) < h(n \xrightarrow{p} k+1 \xrightarrow{q} 1)$ for $q > q_k(p)$ and $h(n \xrightarrow{p} k \xrightarrow{q} 1) > h(n \xrightarrow{p} k+1 \xrightarrow{q} 1)$ for $q < q_k(p)$.

Now suppose that $q_{k^*-1}(p) < q < q_{k^*}(p)$ for some k^*. Then, for $k < k^*$, we have $q > q_k(p)$ and therefore $h(n \xrightarrow{p} k+1 \xrightarrow{q} 1) > h(n \xrightarrow{p} k \xrightarrow{q} 1)$. On the other hand, $k \geq k^*$ implies that $q < q_k(p)$ and $h(n \xrightarrow{p} k \xrightarrow{q} 1) > h(n \xrightarrow{p} k+1 \xrightarrow{q} 1)$. Thus, we have

$$\begin{aligned} & h(n \xrightarrow{p} 1 \xrightarrow{q} 1) < h(n \xrightarrow{p} 2 \xrightarrow{q} 1) < \cdots < h(n \xrightarrow{p} k^* \xrightarrow{q} 1) \\ \text{and } & h(n \xrightarrow{p} k^* \xrightarrow{q} 1) > h(n \xrightarrow{p} k^*+1 \xrightarrow{q} 1) > \cdots > h(n \xrightarrow{p} n \xrightarrow{q} 1), \end{aligned}$$

and $h(n \xrightarrow{p} k \xrightarrow{q} 1)$ is unimodal in k with a maximum at k^*. □

Figure 1 shows the functions q_k as given by (2) for $n = 7$.

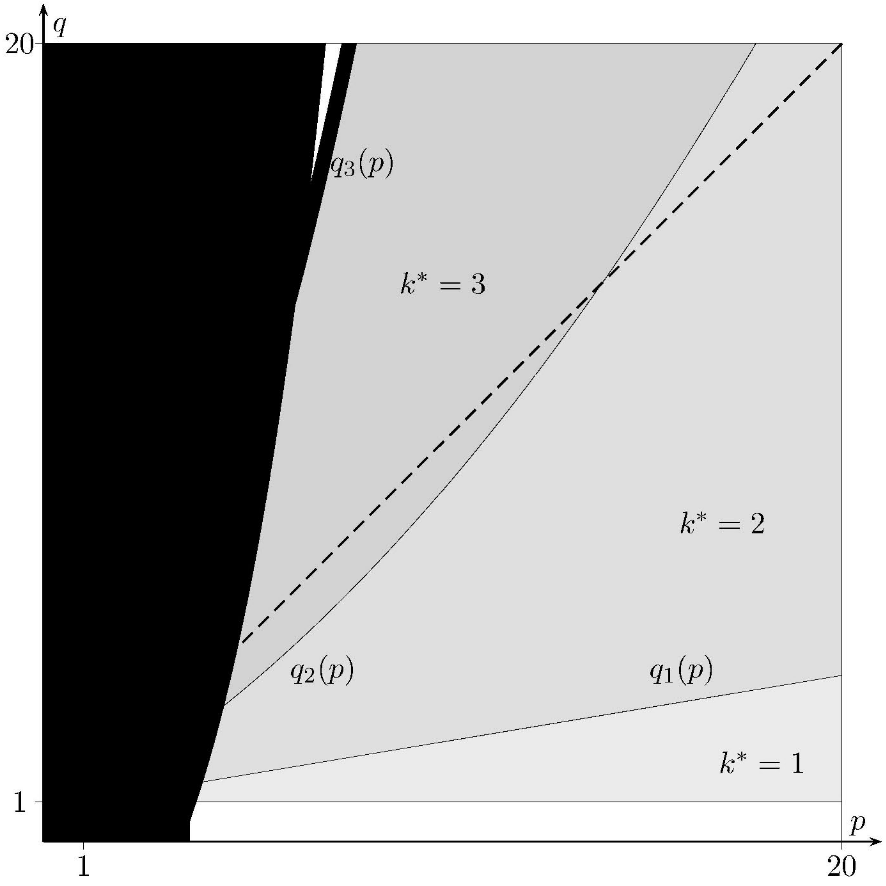


Fig. 1. The case $n = 7$: the functions $q_k(p)$ for $1 \leq k \leq 6$ and $1 \leq p \leq 20$. For $q_{k^*-1}(p) < q < q_{k^*}(p)$, the value k^* yields the highest hitting ratio. The shaded regions show combinations of p and q for which the given value of k^* is optimal. Along the dashed diagonal, we have $p = q$

Lemma 1. *The function $q_k(p)$ is a polynomial of degree k in p over $\mathbb{Q}$ with positive coefficients.*

Proof. Expanding the numerator of the first term in (2), in most of the terms a p appears which can be canceled against the p in the denominator. The only term without a p appears when we multiply only the terms of the form $n-i$ in the brackets:

$$\frac{(n-1)\dots(n-k)(n-k-1)}{(n-1)\dots(n-k)p} = \frac{n-k-1}{p}$$

which is canceled by the second term in (2). Thus $q_k(p)$ is a polynomial. Its degree is obvious, as well as that all coefficients are positive. □.

4 The Maximal Possible k^*

Proposition 2. *Let n be fixed, and consider the case $p = q$. The following statements hold for the value k^* of k which yields the highest hitting ratio:*

- $\lim_{p\to\infty} k^* = 2$.
- *With decreasing $p = q$, the value k^* increases monotonically.*
- *The maximal possible value $k^*_{\max}$ of k^* occurs for $p = 1+\varepsilon$ for small ε.*

Proof. Since all coefficients of $q_k(p)$ are positive by Lemma 1, we have $q_k''(p) > 0$ for all $p > 1$ and $k > 1$, thus q_k is convex. Now, there is some *minimal* value $k^*_{\max}$ of k such that for $k = k^*_{\max}, \dots, n-1$, we have $q_k(p) > p$ for all $p > 1$. The q_k intersect the diagonal $q = p$ for $k < k^*_{\max}$ in descending order of k: first $q_{k^*_{\max}-1}$, then $q_{k^*_{\max}-2}$, ... Whenever some q_k intersects the diagonal at some $p = p_k$, the optimal value of k for $p = q > p_k$ drops from $k+1$ to k. However, since $q_1(p) = (n-2+p)/(n-1)$, the function q_1 does not intersect the diagonal any more, so $k = 1$ will never be optimal. □

We now investigate how $k^*_{\max}$ grows with n.

Proposition 3. *We have*

$$\sqrt{2n+\frac{9}{4}} - \frac{3}{2} \le k^*_{\max} < \sqrt{2n}.$$

Proof. The maximum $k^*_{\max}$ satisfies the following condition on the derivatives of q_k at $p = 1$:

$$q'_{k^*_{\max}}(1) \ge 1 > q'_{k^*_{\max}-1}(1). \tag{3}$$

We calculate $q_k'(p)$ by using the logarithmic derivative for the first term in (2):

$$q_k'(1) = n\left(\sum_{i=n-k}^{n} \frac{1}{i} - 1\right) + n - k - 1 = n\sum_{i=n-k}^{n} \frac{1}{i} - k - 1.$$

Now (3) is equivalent to

$$\begin{aligned} & n \sum_{i=n-k^*_{\max}}^{n} \frac{1}{i} - k^*_{\max} - 1 \geq 1 > n \sum_{i=n-k^*_{\max}+1}^{n} \frac{1}{i} - k^*_{\max} \\ \Longleftrightarrow \quad & n \sum_{i=n-k^*_{\max}}^{n} \left(\frac{1}{i} - \frac{1}{n}\right) \geq 1 > n \sum_{i=n-k^*_{\max}+1}^{n} \left(\frac{1}{i} - \frac{1}{n}\right) \\ \Longleftrightarrow \quad & \sum_{j=1}^{k^*_{\max}} \frac{j}{n-j} \geq 1 > \sum_{j=1}^{k^*_{\max}-1} \frac{j}{n-j}. \end{aligned}$$

Since the function $\frac{x}{n-x}$ is convex for $0 < x < n$, Jensen's inequality implies:

$$1 > \sum_{j=1}^{k^*_{\max}-1} \frac{j}{n-j} \geq (k^*_{\max} - 1) \cdot \frac{k^*_{\max}/2}{n - k^*_{\max}/2}, \quad \text{thus} \quad k^*_{\max} < \sqrt{2n}.$$

On the other hand, we have

$$1 \leq \sum_{j=1}^{k^*_{\max}} \frac{j}{n - k^*_{\max}} = \frac{k^*_{\max}(k^*_{\max}+1)}{2(n - k^*_{\max})}, \quad \text{thus} \quad k^*_{\max} \geq \sqrt{2n + \frac{9}{4}} - \frac{3}{2}.$$

□

5 Conclusion and Open Questions

The shortlisting valleys found in [1,3] do not appear in this urn model. The optimal value of the shortlist size k is quite easy to find and behaves in a simple way.

However, the model we considered above is a very restricted one. First, all but one of the balls are homogeneous. The situation of inhomogeneous balls would be interesting to study, and a start has already been made in [2].

Second, Alice and Bob had the same "precision parameter", namely the weight p. However, the proof of Prop. 1 and Fig. 1 already deal with a different "subjective weight" q of ball 1 for Bob.

Finally, the order of Alice and Bob is fixed. It may be advantageous to let Bob do the first shortlisting step if Alice and Bob differ in precision, either in the case of $n - 1$ homogeneous balls or in the general case. However, the situation in this case is far from straightforward: already for $n = 4$ and the simple model used above, there are examples where putting the more precise expert (i.e., the expert with the larger parameter) in the first position is best (e.g., $p = 4$, $q = 3$) and other examples where putting him in the second position is best (e.g., $p = 4$, $q = 2$). This seems to result from the condition that k be an integer: we conjecture that if the value of k that maximizes the hitting ratio (1) is allowed to take any real value, placing the better expert in the second position is always better than placing him in the first position.

References

1. Kolassa S. (2004a) An Anomaly in Two-Step Shortlisting by Imperfect Experts. Submitted to Mathematics and Computers in Simulation
2. Kolassa S. (2004b) Multi-Step Shortlisting by Imperfect Experts. Submitted as dissertation at the Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität Jena, Germany.
3. Kolassa S., Schwarz S. (2003) Two-Step Shortlisting by Imperfect Experts. Technical Report 03-19. Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität Jena, Germany. `http://www.minet.uni-jena.de/preprints/kolassa_03/two_agent_shortlisting.pdf`
4. Schmidt R. (2004) Multiple-Choice-Optimierung am Beispiel des Urnenmodells. Unpublished manuscript. Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität Jena, Germany

Stephan Kolassa is grateful for support from the DFG-Graduiertenkolleg "Approximation und algorithmische Verfahren."

Total Reward Variance in Discrete and Continuous Time Markov Chains

Karel Sladký[1] and Nico M. van Dijk[2]

[1] Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4, 182 08 Praha 8, Czech Republic
e-mail: sladky@utia.cas.cz
[2] University of Amsterdam, Department of Economic Sciences and Econometrics,
Roetersstrat 11, 1018 WB Amsterdam, The Netherlands
e-mail: nivd@fee.uva.nl

Abstract. This note studies the variance of total cumulative rewards for Markov reward chains in both discrete and continuous time. It is shown that parallel results can be obtained for both cases.

First, explicit formulae are presented for the variance within finite time. Next, the infinite time horizon is considered. Most notably, it is concluded that the variance has a linear growth rate. Explicit expressions are provided, related to the standard average reward case, to compute this growth rate.

1 Introduction

The usual optimization criteria examined in the literature on optimization of Markov reward processes, e.g. total discounted or mean reward, may be quite insufficient to characterize the problem from the point of the decision maker. To this end it is necessary to select more sophisticated criteria that also reflect variability-risk features of the problem, most notably by taking into account the variance of the cumulative rewards. For a detailed discussion of such approaches see the review paper by White [8].

In this paper therefore we aim to establish results for the variance of the cumulative rewards. More precisely, it will be shown that similar results can be obtained for the discrete time case (as partly already obtained in the literature [1], [2], [5], [6]) and the continuous time case, which seems to be new. In addition to reward rates, also transition rewards are herein included.

Particularly, for the average (or mean) reward case, it can be concluded that the variance of the total reward has an asymptotic linear growth rate in time. Relations for this growth rate to be computed are provided.

2 Formulation

In both the discrete time and continuous time case consider a Markov reward chain with finite state space $\mathcal{S} = \{1, 2, \ldots, N\}$, denoted by the Markov chains:

$$\{X^{\mathrm{d}}(n) | n = 0, 1, 2, \ldots\} \quad \text{in discrete time}$$
$$\{X^{\mathrm{c}}(t) | t \geq 0\} \quad \text{in continuous time}$$

Throughout we use the superindex d and c to indicate whether the discrete (d) and continuous (c) time case is in order.

Discrete time case

The discrete time Markov reward chain is characterized by

p_{ij} : the one step transition probabilities for a transition from $i \to j$,

r_{ij} : the one-step reward accrued to a transition from $i \to j$,

$\tilde{r}_i = \sum_j p_{ij} r_{ij}$: the expected one-step reward in state i.

Let

$P = [p_{ij}]$ be the transition probability matrix, and

$\Pi^{\mathrm{d}} = \lim\limits_{m \to \infty} \frac{1}{m} \sum_{n=0}^{m-1} P^n$ the limiting matrix.

(Note that $\Pi^{\mathrm{d}} = \lim\limits_{n \to \infty} P^n$ if P is aperiodic.)

Continuous time case

The continuous time Markov reward chain is characterized by

q_{ij} : transition rate for a transition from $i \to j$ $(j \neq i)$ with $q_{ii} = -\sum_{j, j \neq i} q_{ij}$,

r_{ij} : instantaneous transition reward when a transition from $i \to j$ takes place,

r_i : reward rate per unit of time incurred in state i,

$\tilde{r}_i = r_i + \sum_{j, j \neq i} q_{ij} r_{ij}$: total reward rate per unit of time incurred in state i.

Let

$Q = [q_{ij}]$ be the generator matrix. Then

$P^{\mathrm{c}}(t) = e^{tQ}$ is the transition matrix over time t, and

$\Pi^{\mathrm{c}} = \lim\limits_{t \to \infty} P^{\mathrm{c}}(t)$ $(\Leftrightarrow \Pi^{\mathrm{c}} Q = Q \Pi^{\mathrm{c}} = 0)$ the limiting matrix.

Remark. (Transition rewards.)

1. Note that transition rewards r_{ij} are less natural as instantaneous rewards in a discrete-time that in a continuous-time setting. Furthermore, as far as transition rewards are concerned, they can be included as an total reward rate $\tilde{r}_i = r_i + \sum_{j \neq i} q_{ij} r_{ij}$ the expected reward rate in state i.

2. However, the actual state transition of the state process itself and its corresponding reward consequences, in contrast, will be of influence on the second moment and the variance of the total reward. The instantaneous transition rewards therefore cannot be included in the reward rate for the continuous time case. This will also become apparent later on by the expressions that will be derived.

Cumulative rewards

Let $\xi^{\mathrm{d}}(n)$ and $\xi^{\mathrm{c}}(t)$ denote the (random) cumulative rewards for the discrete and continuous time Markov reward chain, i.e.:

$$\xi^{\mathrm{d}}(n) := \sum_{k=0}^{n-1} r_{X_k, X_{k+1}},$$

$$\xi^{\mathrm{c}}(t) := \int_0^t r_{X_s} \mathrm{d}s + \sum_{k=0}^{N(t)} r_{X(\tau_k^-), X(\tau_k^+)}$$

where $N(t)$ denotes the (total) number of jumps before time t and where $X(\tau_k^-)$ and $X(\tau_k^+)$ denote the state found just before and after the k-th jump. We will be interested in the variance of those quantitites. To this end, denote by

$$R_i^{\mathrm{d}}(n),\ R_i^{\mathrm{c}}(t) := \text{the expected cumulative reward, and}$$
$$V_i^{\mathrm{d}}(n),\ V_i^{\mathrm{c}}(t) := \text{the variance of the cumulative reward}$$

for the discrete (d) and continuous (c) time model up to time n or t respectively when starting in state i at time 0. Then by standard (dynamic programming) steps the reward vector $R^{\mathrm{d}}(n)$ and $R^{\mathrm{c}}(t)$ are directly obtained by ($\tilde{r}^{\mathrm{d}}$, $\tilde{r}^{\mathrm{c}}$ denotes one-step reward vector and total reward rate vector respectively)

$$R^{\mathrm{d}}(n) = \mathsf{E}[\xi^{\mathrm{d}}(n) | X^{\mathrm{d}}(0)] = \sum_{k=0}^{n-1} P^k \, \tilde{r}^{\mathrm{d}} \tag{1}$$

$$R^{\mathrm{c}}(t) = \mathsf{E}[\xi^{\mathrm{c}}(t) | X^{\mathrm{c}}(0)] = \int_0^t P(s) \tilde{r}^{\mathrm{c}} \mathrm{d}s = \int_0^t [\tilde{r}^{\mathrm{c}} + Q R^{\mathrm{c}}(s)] \mathrm{d}s. \tag{2}$$

3 Preliminaries

Discrete time

As is well-known (cf. e.g. [3,4]) for the discrete time case we have

$$R^{\mathrm{d}}(n) = \sum_{k=0}^{n-1} P^k \tilde{r}^{\mathrm{d}} = \tilde{r}^{\mathrm{d}} + PR^{\mathrm{d}}(n-1), \quad g^{\mathrm{d}} := \lim_{n\to\infty} n^{-1} R^{\mathrm{d}}(n) = \Pi^{\mathrm{d}} \tilde{r}^{\mathrm{d}} \quad (3)$$

is the vector of the long run average expected rewards. When P^{d} is unichain g^{d} is a *constant vector.*

Furthermore (cf. e.g. [3,4]) there exists a unique solution $w^{\mathrm{d}} = [w_i^{\mathrm{d}}]$ for the set of linear equations

$$w^{\mathrm{d}} + g^{\mathrm{d}} = \tilde{r}^{\mathrm{d}} + P^n w^{\mathrm{d}}, \qquad \Pi^{\mathrm{d}} w^{\mathrm{d}} = 0. \tag{4}$$

Then

$$R^{\mathrm{d}}(n) = n g^{\mathrm{d}} + w^{\mathrm{d}} - P^n w^{\mathrm{d}} \tag{5}$$

and when P is aperiodic,

$$\lim_{n\to\infty} [R^{\mathrm{d}}(n) - n g^{\mathrm{d}}] = w^{\mathrm{d}}$$

with the convergence geometric.

Continuous time case

For the continuous time case we similarly have

$$R^{\mathrm{c}}(t) = \int_0^t P(s)\, \tilde{r}^{\mathrm{c}}\, \mathrm{d}s = \int_0^t [\tilde{r}^{\mathrm{c}} + QR^{\mathrm{c}}(s)]\, \mathrm{d}s, \quad g^{\mathrm{c}} := \lim_{t\to\infty} t^{-1} R^{\mathrm{c}}(t) = \Pi^{\mathrm{c}} \tilde{r}^{\mathrm{c}}. \tag{6}$$

When Q is unichain, the rows of Π^{c} are identical and g^{c} is a *constant vector.* Again (cf. [3,4]) that there exists a unique solution for the set of linear equations

$$g^{\mathrm{c}} = r^{\mathrm{c}} + Q w^{\mathrm{c}}, \qquad \Pi^{\mathrm{c}} w^{\mathrm{c}} = 0 \tag{7}$$

and

$$R^{\mathrm{c}}(t) = g^{\mathrm{c}} t + w^{\mathrm{c}} - P(t) w^{\mathrm{c}} \tag{8}$$

with the exponential convergence

$$\lim_{t\to\infty} [R^{\mathrm{c}}(t) - t g^{\mathrm{c}}] = w^{\mathrm{c}}.$$

4 Variance of the Cumulative Rewards

For both the discrete time and continuous time case by standard conditioning we obtain the relation:

$$\begin{aligned}\mathsf{E}_i\{[\xi(t)]^2\} = \mathsf{E}_i[\xi(s)]^2 + 2\mathsf{E}_i\{\xi(s)\sum_{j\in\mathcal{S}} \mathsf{E}_j[\xi(t-s)]\mathsf{P}(X(s)=j)\}\\ + [\mathsf{E}_i\{\sum_{j\in\mathcal{S}} \mathsf{E}_j[\xi(t-s)]^2\mathsf{P}(X(s)=j)\}].\end{aligned} \tag{9}$$

By using this relation and after a number of manipulations the relations below for the variance can be obtained (see [5] and [7])

$$V_i^{\mathrm{d}}(n+1) = \sum_{j=1}^{N} p_{ij}[r_{ij} + R_j^{\mathrm{d}}(n)]^2 - [R_i^{\mathrm{d}}(n+1)]^2 + \sum_{j=1}^{N} p_{ij}V_j^{\mathrm{d}}(n). \tag{10}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}V_i^{\mathrm{c}}(t) = \sum_{j\in\mathcal{S}} q_{ij}V_j^{\mathrm{c}}(t) + \sum_{j\in\mathcal{S},j\neq i} q_{ij}[r_{ij} + R_j^{\mathrm{c}}(t)]^2 +\\ q_{ii}[R_i^{\mathrm{c}}(t)]^2 - 2R_i^{\mathrm{c}}(t)\{\sum_{j\in\mathcal{S},j\neq i} q_{ij}r_{ij} + \sum_{j\in\mathcal{S}} q_{ij}R_j^{\mathrm{c}}(t)\}.\end{aligned} \tag{11}$$

5 Transient Case

For unichain Markov processes (with P aperiodic in the discrete time case), under conditions $g^{\mathrm{d}} = 0$, $g^{\mathrm{c}} = 0$ on letting $n \to \infty$, resp. $t \to \infty$, from (5), resp. (8), we get $\lim_{n\to\infty} R_i^{\mathrm{d}}(n) = R_i^{\mathrm{d}} = w_i^{\mathrm{d}}$, resp. $\lim_{t\to\infty} R_i^{\mathrm{c}}(t) = R_i^{\mathrm{c}} = w_i^{\mathrm{c}}$. Similarly, if also $\lim_{n\to\infty} V_i^{\mathrm{d}}(n) = V_i^{\mathrm{d}}$, or $\lim_{t\to\infty} V_i^{\mathrm{c}}(t) = V_i^{\mathrm{c}}$ and then (10) and (11) take on the form

$$V_i^{\mathrm{d}} = \sum_{j\in\mathcal{S}} p_{ij}[r_{ij} + R_j^{\mathrm{d}}]^2 - [R_i^{\mathrm{d}}]^2 + \sum_{j\in\mathcal{S}} p_{ij}V_j^{\mathrm{d}} \tag{12}$$

$$\begin{aligned}0 = \sum_{j\in\mathcal{S}} q_{ij}V_j^{\mathrm{c}} + \sum_{j\in\mathcal{S},j\neq i} q_{ij}[r_{ij} + R_j^{\mathrm{c}}]^2 +\\ q_{ii}[R_i^{\mathrm{c}}]^2 - 2R_i^{\mathrm{c}}\{\sum_{j\in\mathcal{S},j\neq i} q_{ij}r_{ij} + \sum_{j\in\mathcal{S}} q_{ij}R_j^{\mathrm{c}}\}.\end{aligned} \tag{13}$$

Note that (12), (13) are always fulfilled in models with discounting (even with P periodic in the discrete time case). For details and further results see [5] and [7].

6 Infinite Horizon – Average Case

6.1 Discrete time case

By using the relations (4), (5) and (10) in a number of steps the following relations can be obtained for the average case

$$V_i^{\mathrm{d}}(n+1) = \sum_{j=1}^{N} p_{ij}[r_{ij} - g^{\mathrm{d}} + w_j^{\mathrm{d}}]^2 - [w_i^{\mathrm{d}}]^2 + \sum_{j=1}^{N} p_{ij} V_j^{\mathrm{d}}(n) + o(n). \qquad (14)$$

Hence with the vector s^{d} defined by

$$s_i^{\mathrm{d}} = \sum_{j=1}^{N} p_{ij}[r_{ij} - g^{\mathrm{d}} + w_j^{\mathrm{d}}]^2 - [w_i^{\mathrm{d}}]^2$$

in matrix form we have

$$V^{\mathrm{d}}(n+1) = s^{\mathrm{d}} + PV^{\mathrm{d}}(n) + \varepsilon(n) \qquad (15)$$

where all elements of $\varepsilon(n)$ converge to the null vector geometrically, that is $\|\varepsilon(n)\| \le c\varepsilon^n$ for some numbers $c > 0$, $\varepsilon \in (0,1)$.

Since $\varepsilon(n)$ converges to the null vector geometrically, in analogy with (3) and by iteration we can also show that the growth rate

$$\gamma^{\mathrm{d}} = \lim_{n\to\infty} n^{-1} V^{\mathrm{d}}(n) \quad \text{exists}$$

and in virtue of (4), (5) this growth rate can be calculated as the (unique) solution of

$$\gamma^{\mathrm{d}} + \bar{w}^{\mathrm{d}} = s^{\mathrm{d}} + P\bar{w}^{\mathrm{d}}, \quad \Pi^{\mathrm{d}}\bar{w}^{\mathrm{d}}. \qquad (16)$$

Since $\gamma^{\mathrm{d}} = \Pi^{\mathrm{d}} s^{\mathrm{d}}$, it also holds that $\gamma^{\mathrm{d}} = \Pi^{\mathrm{d}} s^{\mathrm{d},\ell}$ for $\ell = 1,2,3$, where the vectors $s^{\mathrm{d},\ell}$ are defined by

$$s_i^{\mathrm{d},1} = \textstyle\sum_{j=1}^N p_{ij}\{[r_{ij} - g^{\mathrm{d}}]^2 + 2r_{ij}w_j^{\mathrm{d}}\}, \quad s_i^{\mathrm{d},2} = \textstyle\sum_{j=1}^N p_{ij} r_{ij}[r_{ij} + 2w_j^{\mathrm{d}}] - [g^{\mathrm{d}}]^2,$$

$$s_i^{\mathrm{d},3} = \textstyle\sum_{j=1}^N p_{ij}\{[r_{ij} - g^{\mathrm{d}}]^2 + 2[r_{ij} - g^{\mathrm{d}}]w_j^{\mathrm{d}}\}.$$

Note that the last expression can be applied also if in (4) the normalizing condition $\Pi^{\mathrm{d}} w^{\mathrm{d}} = 0$ is violated, see the early paper [2].

Hence calculating the growth rate γ^{d} in (16) we can replace the term s^{d} by $s^{\mathrm{d},\ell}$, $\ell = 1,2,3$ defined above. For details and further results see [5].

6.2 Continuous time case

With the $(N \times 1)$-vector s^{c} defined by

$$s_i^{\mathrm{c}} = \sum_{j \in \mathcal{S}, j \neq i} q_{ij}\{[r_{ij} + w_j^{\mathrm{c}}]^2 - [w_i^{\mathrm{c}}]^2\} + 2[r_i - g^{\mathrm{c}}]w_i^{\mathrm{c}}$$

it can be shown that

$$\frac{\mathrm{d}}{\mathrm{d}t} V_i^{\mathrm{c}}(t) = s_i^{\mathrm{c}} + \sum_{j \in \mathcal{S}} q_{ij} V_j^{\mathrm{c}}(t) + o(t) \tag{17}$$

or in matrix form

$$\frac{\mathrm{d}}{\mathrm{d}t} V^{\mathrm{c}}(t) = s^{\mathrm{c}} + QV^{\mathrm{c}}(t) + \varepsilon(t) \tag{18}$$

where all elements of the $(N \times 1)$-vector $\varepsilon(t)$ converge to zero exponentially. That is $\|\varepsilon(t)\| \le c e^{-\delta t}$ for some numbers $c > 0$, $\delta > 0$.

In virtue of the above exponential convergence, furthermore, we can conclude that

$$\lim_{t \to \infty} \frac{1}{t} V^{\mathrm{c}}(t) = \gamma^{\mathrm{c}},$$

where γ^{c} is the unique solution to

$$\gamma^{\mathrm{c}} = s^{\mathrm{c}} + Q\bar{w}^{\mathrm{c}}, \quad \Pi^{\mathrm{d}}\bar{w}^{\mathrm{c}} = 0. \tag{19}$$

Since $\gamma^{\mathrm{c}} = \Pi^{\mathrm{c}} s^{\mathrm{c}}$, it also holds that $\gamma^{\mathrm{c}} = \Pi^{\mathrm{c}} s^{\mathrm{c},\ell}$ for $\ell = 1, 2$, where the vectors $s^{\mathrm{c},\ell}$ are defined by

$$s_i^{\mathrm{c},1} = \textstyle\sum_{j \in \mathcal{S}, j \neq i} q_{ij}\{[r_{ij} + w_j^{\mathrm{c}}]^2 - [w_i^{\mathrm{c}}]^2\} + 2r_i w_i^{\mathrm{c}},$$
$$s_i^{\mathrm{c},2} = \textstyle\sum_{j \in \mathcal{S}, j \neq i} q_{ij}\{[r_{ij}]^2 + 2r_{ij}w_j^{\mathrm{c}}\} + 2r_i w_i^{\mathrm{c}}.$$

For details and further results see [7].

Acknowledgements. This research was supported by the Grant Agency of the Czech Republic under Grants 402/02/1015 and 402/04/1294.

References

1. Benito, F. (1982): Calculating the variance in Markov processes with random reward. Trabajos de Estadistica y de Investigacion Operativa, **33**, 73–85
2. Mandl, P. (1971): On the variance in controlled Markov chains. Kybernetika, **7**, 1–12
3. Puterman, M. L. (1994): Markov Decision Processes – Discrete Stochastic Dynamic Programming. Wiley, New York

4. Ross, S. M. (1970): Applied Probability Models with Optimization Applications. Holden–Day, San Francisco, CA
5. Sladký, K., Sitař, M. (2004): Optimal solutions for undiscounted variance penalized Markov decision chains. In: Dynamic Stochastic Optimization (Marti, K., Ermoliev, Y., Pflug, G., Eds.), LNEMS, Vol. 532, Springer, Berlin, pp. 43–66
6. Sobel, M. J. (1982): The variance of discounted Markov decision processes. J. Appl. Probab., **19**, 794–802
7. van Dijk, N. M., Sladký, K. (2004): On total reward variance in continuous-time Markov reward chains. Manuscript
8. White, D. J. (1988): Mean, variance and probability criteria in finite Markov decision processes: A review. J. Optim. Theory Appl., **56**, 1–29

An Efficient Conjugate Directions Method Without Linear Searches

Edouard Boudinov[1] and Arkadiy I. Manevich[2]

[1] Information Bank, FORTIS Bank, P.O. Box 243, NL-1000AE Amsterdam, The Netherlands
[2] Dniepropetrovsk National University, Dniepropetrovsk, Ukraine

Abstract. New conjugate directions algorithms are proposed, which are based on an orthogonalization procedure and do not perform line searches. The orthogonalization procedure prevents the conjugate vectors set from the degeneracy, eliminates high sensitivity to computation errors pertinent to methods of conjugate directions, and thus enable us to solve large-scale minimization problems without preconditioning. Numerical experiments have confirmed high efficiency of the algorithms for minimizing large-scale quadratic functions.

1 INTRODUCTION

Conjugate directions (CD) methods are widely used for solving large-scale linear algebra and optimization problems because of minimum storage requirements. But their theoretical property of quadratic termination holds only in precise arithmetic. If the number of variables N is large then even for quadratic objective functions the number of iterations can exceed $N + 1$ by several times because of high sensitivity to computational errors. The cause is due to the usual procedure of conjugate vector construction based on the assumption that exact linear minima have been found on all preceding iterations (only under this premise the conjugacy conditions with respect to all conjugate vectors are satisfied). But even very small errors in locating the linear minima are accumulated yielding to violation of the conjugacy conditions and gradual derangement of the conjugate vector set.

In last years a clear trend to using inexact linear search is manifested aiming to reduce the number of functions or/and gradients calls and running time. Moreover, already in 70–80's there were proposed Quasi-Newton and conjugate directions algorithms without linear searches [4], [5], [8], [11]. They generate conjugate directions (or/and sequence of matrices) without accurate line minimization, retaining the quadratic termination property. However until now these algorithms were not able to solve large-scale minimization problems.

In this work a new CD - method without linear minimizations is proposed, which employs another scheme of construction of conjugate vectors based on an orthogonalization procedure: in constructing the next conjugate vector the component of the function gradient is used that is orthogonal to the subspace of preceding conjugate vectors (instead of the gradient itself). Such an approach allows attaining very high accuracy in constructing conjugate directions and thus eliminates high sensitivity to computation errors pertinent to known CD-methods. As this procedure does not require the current point to be the minimum point along a given conjugate vector, any function evaluations for linear minimization on each iteration can be excluded ensuring considerable decrease of running time.

At each iteration only one step is done, which includes displacements along the last two conjugate vectors, so the movement occurs in the hyperplane determined by these conjugate vectors. Along each conjugate vector two successive steps are made: a preliminary step on k-th iteration and the Newton-like step on the next iteration.

The proposed algorithm and its modification have been realized in JAVA programming language. They have been tested on several quadratic test-functions with the number of variables N up to 1'000'000. The numerical experiments have confirmed that these algorithms locate the minimum for quadratic functions of very many variables in at most $N+1$-th steps, without perceptible losses in precision. The numerical experiments have shown that the running time for solving all test - problems was less by several times in comparison with known efficient algorithms for large-scale problems.

2 DESCRIPTION OF THE METHOD

2.1 Basic algorithm

Consider the basic algorithm as applied to minimization of quadratic functions (we do not discuss here modifications needed for non-quadratic functions).

At the first iteration $k = 1$ vectors d_1, n_1, and the step are as follows:

$$d_1 = n_1 = -\frac{g_1}{\|g_1\|}, \qquad x_2 = x_1 + \delta_1^{(1)} d_1, \tag{1}$$

where $g_k = g(x_k)$ is the gradient of the objective function $f(x)$, and $\delta_1^{(1)}$ is an arbitrary step.

Let, after $k-1$-th step, $\{n_i\}$ $(i = 1, \dots, k-1)$ be an orthonormalized vector set in the subspace Ω_{k-1}, which is determined by the conjugate vectors $\{d_i\}$ $(i = 1, \dots, k-1)$. Vector n_k (a normalized vector, orthogonal to the subspace Ω_{k-1}), is

$$n_k = \frac{n_k^*}{\|n_k^*\|}, \qquad n_k^* = -g_k + \gamma_{k-1}^{(k)} n_{k-1}, \qquad \gamma_{k-1}^{(k)} = (g_k, n_{k-1}) \tag{2}$$

((x,y) denotes the scalar product of vectors x and y). A new conjugate vector d_k is sought as a linear combination of the preceding conjugate vector d_{k-1} and the vector n_k $(\Delta g_{k-1} = g_k - g_{k-1})$:

$$d_k = \frac{d_k^*}{\|d_k^*\|}, \qquad d_k^* = n_k + \beta_{k-1}^{(k)} d_{k-1}, \qquad \beta_{k-1}^{k} = -\frac{(n_k, \Delta g_{k-1})}{(d_{k-1}, \Delta g_{k-1})} \tag{3}$$

The step on the k-th iteration is as follows:

$$x_{k+1} = x_k + \alpha_{k-1}^{(k)} d_{k-1} + \delta_k^{(k)} d_k, \tag{4}$$

$$\alpha_{k-1}^{(k)} = -\frac{\delta_{k-1}^{(k-1)}(g_k, d_{k-1})}{(\Delta g_{k-1}, d_{k-1})}, \qquad \delta_k^{(k)} = \beta_{k-1}^{(k)} \cdot (\delta_{k-1}^{(k-1)} + \alpha_{k-1}^{(k)}) \tag{5}$$

The algorithm stops when, for a given ϵ, the condition $\|g_k\| \leq \epsilon$ is satisfied. So the motion on the k-th iteration occurs in the hyper-plane of the last two conjugate vectors. The component of this step along the direction d_k is a preliminary step $\delta_k^{(k)}$, which can be an arbitrary one for quadratic functions. The component of the step along the direction d_{k-1} is the Newton-like step obtained from the gradient projections on d_{k-1} at the last two points x_{k-1} and x_k. Thus, instead of linear searches, two steps are done along each conjugate direction: a given step and, on the next iteration, the Newton-like step. The result is that for quadratic functions the minimum along d_{k-1} is reached on the k-th iteration without any function evaluations.

It can be proved that the above procedure (Eqs. (1)- (5)) does lead for quadratic functions to the construction of an orthonormalized vector set $\{n_i\}$ $(i = 1, \ldots, k)$ and an **A**-conjugate vector set $\{d_i\}$ $(i = 1, \ldots, k)$ on the k-th iteration.

2.2 Modified algorithm

The accuracy of determining the normal vector by formulae (2) can be insufficient for large N in the case when the vector g_k makes a very small angle to the subspace Ω_{k-1}. Even very small rounding errors accumulated at all steps lead to disturbance and derangement of the conjugate vector set and the normal vector set. Errors of order 10^{-15} at each iteration can result in failing the search for large scale problems.

An efficient way to suppress the rounding errors and to rise the accuracy is to orthogonalize the vector n_k^* not only with respect to n_{k-1} but with respect to all preceding n_i $(i = 1, \ldots, k-1)$. This idea is used in the "modified algorithm", which determines the normal vector as follows (instead of Eq. (2)):

$$n_k^* = -g_k + \sum_{i=1}^{k-1} \gamma_i^{(k)} n_i, \qquad \gamma_i^{(k)} = (g_k, n_i) \tag{6}$$

But this algorithm requires storage of all found normal vectors, in distinction from the basic algorithm.

3 NUMERICAL EXPERIMENTS

The proposed algorithms have been realized in JAVA programming language. The algorithms has been tested on several quadratic test–functions with the number of variables N up to 1'000'000. We present the results for the following quadratic functions of N variables.

$$F_1(x_1,\dots,x_N)=\sum_{i=1}^{N}\frac{x_i^2}{i}+\lambda\sum_{i=1}^{N}\sum_{j=i+1}^{N}\frac{x_i\,x_j}{ij}, \qquad (7)$$

$$F_s(x_1,\dots,x_{10000})=\sum_{i=1}^{10000}\frac{x_i^2}{i^s}, \qquad s=1,\dots,5 \qquad (8)$$

(in all cases the start point was $x^{(0)}=(1,\dots,1)$ and minimum $F(0,\dots,0)=0$).

Functions (7) were used to examine the proposed algorithms at solving very large scale quadratic problems (up to 1 million variables) with the diagonal matrices (if $\lambda=0$), as well as with the dense matrices. Functions (8) are used to examine the performance of the algorithms on quadratic functions with Hessian that is very close to degenerative one.

For comparison we have chosen the implementations of the following three groups of methods: the conjugate gradients methods from $CG+$ package [9], the quasi-Newton BFGS method (Broyden, Fletcher, Goldfarb, and Shanno [1],[2], [3], [6], [12]) and the trust-region method that combines the conjugate gradients and quasi-Newton methods in one algorithm [7], [10], [12].

All computations have been performed on PC Pentium 2.66 GHz with RAM of 785.9 MB in double precision. The gradient was calculated analytically.

Note the proposed algorithm has only two parameters that should be supplied by a user: the initial step $\delta_1^{(1)}$, and the termination tolerance on the function gradient value ϵ (all runs were stopped when $\|g_k\|\leq\epsilon$). The $\delta_1^{(1)}$ value was always equal to 0.5. The ϵ value varied in the range of $10^{-12}-10^{-20}$ depending on the accuracy needed for a particular function minimum.

The same range of the termination tolerance on the function gradient was used for the conjugate gradients methods from the CG+ package. For the quasi-Newton and trust-region methods the termination tolerance on the function value was taken equal to 10^{-20}.

Table 1 presents the results obtained for the function F_1 with $\lambda=1$. The number of variables N varies from 1000 to 20'000. The "Basic Algorithm" relates to the algorithm that uses in Eq. (2) only one normal vector (found on the preceding iteration). The "Modified Algorithm" relates to the algorithm where all already known normal vectors are used in reconstruction of vector n_k in Eq. (6). In all tables N is the number of variables, ϵ_g, ϵ_f and ϵ_x is the accuracy in gradient, function and arguments, respectively, k_g and k_f are the numbers of the gradient and function evaluations, respectively; t is the running time (CPU consumption) in seconds; n_n^{max} is the number of stored normal vectors.

Table 1. Results for quadratic function $F_1(x_1, \dots, x_N) = \sum_{i=1}^{N} \frac{x_i^2}{i} + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{x_i\, x_j}{ij}$

N	ϵ_g	ϵ_f	ϵ_x	iter	k_g	k_f	n_n^{max}	t (sec)
Modified Algorithm								
4'000	$< 10^{-12}$	$< 10^{-22}$	$< 10^{-10}$	150	151	1	146	39
10'000	$< 10^{-12}$	$< 10^{-21}$	$< 10^{-10}$	201	202	1	195	317
20'000	$< 10^{-12}$	$< 10^{-21}$	$< 10^{-10}$	258	259	1	246	1593
Basic Algorithm								
4'000	$< 10^{-12}$	$< 10^{-22}$	$< 10^{-10}$	305	306	1	1	73
10'000	$< 10^{-12}$	$< 10^{-21}$	$< 10^{-10}$	473	474	1	1	702
20'000	$< 10^{-12}$	$< 10^{-21}$	$< 10^{-10}$	668	669	1	1	3926
Fletcher-Reeves								
4'000	$< 10^{-12}$	$< 10^{-22}$	$< 10^{-10}$	851	1709	1709		1586
10'000	$< 10^{-12}$	$< 10^{-22}$	$< 10^{-10}$	1310	2628	2628		15226
Polak-Ribiere								
4'000	$< 10^{-12}$	$< 10^{-22}$	$< 10^{-10}$	817	1635	1635		1517
10'000	$< 10^{-12}$	$< 10^{-22}$	$< 10^{-10}$	992	1996	1996		11565
Quasi-Newton								
4'000	$< 10^{-9}$	$< 10^{-17}$	$< 10^{-7}$	315	316	2024		1062
Trust-Region								
1'000	$< 10^{-11}$	$< 10^{-24}$	$< 10^{-11}$	34	11011	11011		3461

For all values of N the minima have been obtained by the both proposed algorithms in less than $N + 1$ steps with very high accuracy in function and parameters x_i, and this accuracy can be secured for very large N. The Modified version of the proposed algorithm appears to be most efficient in terms of number of iterations, functions and gradient calls as well as CPU time; The basic algorithm requires approximately twice as much iterations and time, but the number of iterations remains much less than the number of variables.

The Fletcher-Reeves and Polak-Ribiere algorithms demand the number of iterations approximately by 10 times as much as the modified algorithm. The line minimization used in the conjugate gradients algorithms clearly consumes a lot of CPU through function and gradient calculations, but it does improve the convergence rate. The quasi-Newton method requires the memory allocation proportional to N^2, this results in 'Out-of-memory'-problem when N is larger than approximately 4000 (for the computer used). For $N < 4000$ the running time is slightly lower for the quasi-Newton method than for the conjugate gradient algorithms. The trust-region method clearly turns out to be less efficient for this test-function for all N values and requires a lots of CPU for $N > 1000$.

In order to check the algorithm for even higher N values we have considered the quadratic function $F_1(x_1,\dots,x_N)$ with $\lambda = 0$, which has a diagonal Hessian. Note that the both proposed algorithms are independent of whether Hessian is dense or diagonal. But the running time for a function with diagonal Hessian is essentially lower (the search includes at most $N+1$ determinations of steps, every step determination takes the number of basic operations proportional to N; each iteration includes one gradient evaluation, every gradient evaluation for a quadratic function takes the number of basic operations proportional to N^2 in the case of dense Hessian, and to N in the case of diagonal Hessian).

The results for N values up to 1'000'000 are presented in Table 2. Two proposed algorithms are compared with the Fletcher-Reeves algorithm. One can see that for very large numbers of variables the basic algorithm becomes preferable. It enables us to solve the problem up to $N = 1'000'000$ performing only about 5000 steps. The storage requirements do not permit the modified algorithm be used for $N > 100'000$. The Fletcher-Reeves algorithm demands at least twice the number of iterations in comparison with the basic algorithm.

Table 2. Results for quadratic function $F_1(x_1,\dots,x_N) = \sum_{i=1}^{N} \frac{x_i^2}{i}$

N	ϵ_g	ϵ_f	ϵ_x	iter	k_g	k_f	n_n^{max}	t (sec)
Modified Algorithm								
20'000	$< 10^{-12}$	$< 10^{-21}$	$< 10^{-9}$	241	242	1	240	16
50'000	$< 10^{-12}$	$< 10^{-20}$	$< 10^{-9}$	324	325	1	324	118
100'000	$< 10^{-12}$	$< 10^{-20}$	$< 10^{-9}$	406	407	1	405	356
Basic Algorithm								
20'000	$< 10^{-12}$	$< 10^{-21}$	$< 10^{-9}$	652	653	1	1	8
50'000	$< 10^{-12}$	$< 10^{-21}$	$< 10^{-9}$	1021	1022	1	1	33
100'000	$< 10^{-12}$	$< 10^{-21}$	$< 10^{-9}$	1446	1447	1	1	86
1'000'000	$< 10^{-12}$	$< 10^{-20}$	$< 10^{-9}$	4557	4558	1	1	2490
Fletcher-Reeves								
20'000	$< 10^{-12}$	$< 10^{-22}$	$< 10^{-10}$	1520	3054	3054		28
50'000	$< 10^{-12}$	$< 10^{-22}$	$< 10^{-10}$	2616	5245	5245		125
100'000	$< 10^{-12}$	$< 10^{-22}$	$< 10^{-10}$	3786	7591	7591		361

Function $F_s(x_1,\dots,x_N)$ Eq. (8) is designed to investigate how the convergence of minimization procedures depends on the rate of decrease of the eigenvalues, which is governed by changing the parameter s. We compare the results obtained for this function at various s by the proposed methods and three CG - algorithms in Table 3, for $N = 10'000$.

The modified algorithm has effectively solved all problems up to s=5 with very high accuracy; the number of steps for $N = 10'000$ much less than the number of variables. The basic algorithm as well as the CG-methods can solve

Table 3. Results for the quadratic function $F_s(x_1, \ldots, x_{10000}) = \sum_{i=1}^{10000} \frac{x_i^2}{i^s}$

s	ϵ_g	ϵ_f	ϵ_x	iter	k_g	k_f	n_n^{max}	t (sec)
Modified Algorithm								
s=1	$< 10^{-20}$	$< 10^{-37}$	$< 10^{-19}$	396	397	1	276	9
s=2	$< 10^{-20}$	$< 10^{-34}$	$< 10^{-14}$	974	975	1	737	55
s=3	$< 10^{-20}$	$< 10^{-30}$	$< 10^{-10}$	1590	1591	1	1231	153
s=4	$< 10^{-20}$	$< 10^{-27}$	$< 10^{-6}$	2278	2279	1	1627	288
s=5	$< 10^{-25}$	$< 10^{-32}$	$< 10^{-8}$	3338	3339	1	2298	620
Basic Algorithm								
s=1	$< 10^{-12}$	$< 10^{-21}$	$< 10^{-9}$	463	464	1	1	3
s=2	$< 10^{-12}$	$< 10^{-18}$	$< 10^{-6}$	19413	19414	1	1	128
Fletcher-Reeves								
s=1	$< 10^{-12}$	$< 10^{-22}$	$< 10^{-10}$	1024	2054	2054		9
s=2	$< 10^{-12}$	$< 10^{-21}$	$< 10^{-7}$	100012	201008	201008		1129
Polak-Ribiere								
s=1	$< 10^{-12}$	$< 10^{-22}$	$< 10^{-10}$	764	1538	1538		6
s=2	$< 10^{-12}$	$< 10^{-19}$	$< 10^{-6}$	110013	220521	220521		1237
Positive Polak-Ribiere ($\beta = \max \beta, 0$)								
s=1	$< 10^{-12}$	$< 10^{-22}$	$< 10^{-10}$	764	1538	1538		7
s=2	$< 10^{-12}$	$< 10^{-20}$	$< 10^{-7}$	100012	200512	200512		1113

problems only for $s = 1$ and $s = 2$. These algorithms have approximately the same convergence at $s = 1$. For $s = 2$ the considered CG-algorithms require the number of gradient evaluations about ten times as much as the proposed basic algorithm.

4 CONCLUSIONS

A new algorithm of conjugate directions and its modification are proposed that are based on the use of an orthogonalization procedure. The basic algorithm uses the only preceding "normal" vector and requires storage of only one conjugate vector and corresponding normal vector. The modified algorithm employs all previously found normal vectors in the orthogonalization procedure, Eq. (6). Both algorithms do not include linear searches, replacing them by successive two steps along each conjugate direction (the preliminary step and the Newton-like step). Such an approach ensures significant reduction in the number of gradient and function evaluations and the running time. The proposed orthogonalization procedure prevents the conjugate vectors set from degenerating.

The proposed algorithms have been tested on various quadratic test–functions of very large number of variables N, up to 1'000'000. They turn out to be superior in terms of the number of steps and the running time for quadratic functions in the whole range of N considered, in comparison with the CG–algorithms, quasi-Newton method and trust–region method. In all the problems considered the both algorithms successfully found minima with very high accuracy, and the number of steps was considerably less than the number of variables.

The basic algorithm is preferable at solving very large dimension problems, but it is less efficient (in comparison with the modified algorithm) at minimization of functions with sharply changing eigenvalues. The modified algorithm is found to be the best at solving complicated minimization problems, but in the case of limited memory allocation one would need either to limit the number of normal vectors used in the orthogonalization procedure or to restart the procedure after a certain number of iterations.

References

1. Broyden C.G. The Convergence of a Class of Double-rank Minimization Algorithms. *J. Inst. Maths. Applics.* 1970; **6**: 76–90.
2. Goldfarb D. A Family of Variable Metric Updates Derived by Variational Means. *Mathematics of Computing* 1970; **24**: 23-26.
3. Shanno D.F. Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computing* 1970; **24**: 647–656.
4. Dixon L.C.W. Conjugate Directions without Linear Searches. *J. Inst. Maths. Applic.* 1973; **11**: 317–328.
5. Nazareth L. A Conjugate Direction Algorithm without Line Searches. *Journal of Optimization Theory and Applications* 1977; **23**: 373–387.
6. Fletcher R. *Practical Methods of Optimization.* Vol. 1. J. Wiley: 1980.
7. Byrd R.H., Schnabel R.B. and Shultz G.A. Approximate Solution of the Trust Region Problem By Minimization over Two-Dimensional Subspaces. *Mathematical Programming* 1988; **40**: 247–263.
8. Manevich A.I. and Polyanchikov P.I. Single-step method of conjugate directions. *Izvestija of USSR Academy of Sciences, Technical Cybernetics* 1984; **6**: 41-47 (in Russian).
9. Gilbert J.C. and Nocedal J. Global Convergence Properties of Conjugate Gradient Methods for Optimization. *SIAM Journal on Optimization* 1992; **2**: 21–42.
10. Branch M.A., Coleman T.F. and Li Y. A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained minimization Problems. *SIAM Journal on Scientific Computing* 1999; **21**(1): 1-23.
11. Manevich A.I. and Boudinov E. An efficient conjugate directions method without linear minimization. *Nuclear Instruments and Methods in Physics Research* 2000; **A 455**: 698–705.
12. *Optimization Toolbox for Use with MATLAB. User's Guide. Version 2.* The MathWorks Inc., 2000.

The Robust Shortest Path Problem by Means of Robust Linear Optimization*

D.Chaerani[1], C.Roos[1], and A.Aman[2]

[1] Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Sciences
Department of Software Technology, Algorithms Group
Mekelweg 4, 2628 CD Delft
The Netherlands
E-mail: {d.chaerani, c.roos}@ewi.tudelft.nl
[2] Department Mathematics Institut Pertanian Bogor Indonesia

Abstract. We investigate the robust shortest path problem using the robust linear optimization methodology as proposed by Ben-Tal and Nemirovski. We discuss two types of uncertainty, namely, box uncertainty and ellipsoidal uncertainty. In case of box uncertainty, the robust counterpart is simple. It is a shortest path problem with the original arc lengths replaced by their upper bounds. When dealing with ellipsoidal uncertainty, we obtain a conic quadratic optimization problem with binary variables. We present an example to show that a subpath of a robust shortest path is not necessarily a robust shortest path.

1 Introduction

The shortest path problem (SPP) is one of the simplest and most well-studied combinatorial optimization problems. For a given network $G = (\mathcal{V}, \mathcal{A})$ with node set $\mathcal{V}$ and arc set $\mathcal{A}$, an instance of the SPP is finding a directed path from a single node s to a single node t with a minimal total length, with respect to a given length function $\mathbf{c} : \mathcal{A} \to \mathbb{R}_+$. A binary optimization model for the shortest path problem is

$$\min\{c^T x \ : \ Ax = b, x \in \{0,1\}^{\mathcal{A}}\}, \tag{1}$$

where A is the node-arc incident matrix, b is the vector in $\mathbf{R}^{\mathcal{V}}$ with entries $b_s = 1$, $b_t = -1$ and $b_j = 0$ for all $j \in \mathcal{V} \setminus \{s,t\}$, and c is the vector of arc lengths. It is well known that the constraint matrix A is a totally unimodular and hence we can equally consider the linear optimization problem

$$\min\{c^T x \ : \ Ax = b, x \in [0,1]^{\mathcal{A}}\}.$$

Recently, some authors considered the SPP for the case where the arc lengths are not certain. The aim is then to find a so called robust shortest

* Financial support was provided by the Royal Netherlands Academy of Arts and Sciences in the framework of the Scientific Programme Indonesia - Netherlands (SPIN).

path, i.e., a path that behaves relatively stable under all possible realizations of the uncertain arc lengths. For instance, when a communication network is used to send packages from a source to a sink, an uncertain delay time can occur. Treating the uncertain delay time between two nodes as an uncertain arc length, we obtain an uncertain SPP.

Karasan et al. [4] and Bertsimas et al. [5] considered the uncertain shortest path problem. They model data uncertainty by assuming that each arc length belongs to an interval. We will refer to this case as box uncertainty. Karasan et al. define the robust path as a path with minimum deviation, i.e., the minimal difference between the worst case path length and the shortest path length for all possible realizations. A variant of this approach is proposed by Bertsimas et al. who assume that the uncertainty occurs only on a fixed number of the arcs.

In this paper we apply the Robust Linear Optimization (RLO) methodology as proposed by Ben-Tal and Nemirovski (see [1–3]). This methodology applies to general LO problem. We apply it in this paper to the SPP.

The paper is organized as follows. Section 2 briefly introduces the theory of Robust Linear Optimization. Section 3 is devoted to the RSPP and gives an illustrative example. Conclusions can be found in Section 4.

2 Robust Linear Optimization

We briefly recall some definitions and some main results from [3]. Consider a linear optimization problem

$$\min_x \{c^T x \ : \ Ax \geq b\}, \tag{$\mathcal{LO}$}$$

and let $\mathcal{U}$ be the set of all possible realizations of (A, b, c). So the set $\mathcal{U}$ models the uncertainty in the data, A, b and c, of $(\mathcal{LO})$. We call $\mathcal{U}$ the uncertainty set. As a consequence, we have a whole family of $\mathcal{LO}$ problems, for each $(c, A, b) \in \mathcal{U}$, one $\mathcal{LO}$ problem. This family is given by

$$\left\{\min_x \{c^T x \ : \ Ax \geq b\} \ : \ (c, A, b) \in \mathcal{U}\right\}. \tag{2}$$

Instead we consider the so-called robust counterpart of (2), namely

$$\min \left\{\ell \ : \ \ell \geq c^T x, Ax - b \geq 0, \forall (c, A, b) \in \mathcal{U}\right\}. \tag{3}$$

The formulation (3) is a linear optimization problem with usually infinitely many constraints, depending on the uncertainty set $\mathcal{U}$. Hence, in general this problem may be very hard to solve. Special cases for $\mathcal{U}$ make (3) computational tractable. In [1], it has been shown that the robust counterpart (3) is equivalent to an explicit computationally tractable problem provided that $\mathcal{U}$ has a simple structure. This becomes clear from the following theorem which presents three examples of computational tractable uncertainty sets.

Theorem 1 (cf. [3]). *Assume that the uncertainty set $\mathcal{U}$ in (2) is given as the affine image of a bounded set $\mathcal{Z} = \{\zeta\} \subset \mathbb{R}^N$, and $\mathcal{Z}$ is given either*

1. *by a system of linear inequalities $P\zeta \leq p$, or*
2. *by a system of conic quadratic inequalities $\|P_i\zeta - p_i\|_2 \leq q_i^T\zeta - r_i, i = 1, \dots, M$, or*
3. *by a system of linear matrix inequalities $P_0 + \sum_{i=1}^{dim\zeta} \zeta_i P_i \succeq 0$.*

In the cases 2 and 3 assume also that the system of constraints defining $\mathcal{U}$ is strictly feasible. Then the robust counterpart (3) of (2) is equivalent to a linear optimization problem in case 1, a conic quadratic problem in case 2, and a semidefinite problem in case 3. In all cases, the data of the resulting robust counterpart problem are readily given by M, N and the data specifying the uncertainty set. Moreover, the size of the resulting problem is polynomial in the size of the data specifying the uncertainty set.

In the following we restrict ourselves to the SPP and we consider only box and ellipsoidal uncertainty.

3 The Robust Shortest Path Problem (RSPP)

3.1 Problem definition

In case of the uncertain SPP, A and b are fixed. The uncertainty is only in the vector c, so $\mathcal{U}$ will denote the set of all possible realizations of c. Thus, the family of uncertain SPP's is given by

$$\left\{\min_x\{c^T x \ : \ Ax = b, x \in \{0,1\}^{\mathcal{A}}, c \in \mathcal{U}\right\}, \tag{4}$$

and the corresponding robust counterpart is

$$\min\{\ell \ : \ \ell \geq c^T x, Ax = b, x \in \{0,1\}^{\mathcal{A}}, \forall c \in \mathcal{U}\}. \tag{5}$$

Using the set

$$\mathcal{P} = \left\{x : Ax = b, x \in \{0,1\}^{\mathcal{A}}\right\},$$

which consists of the incidence vectors of all $s - t$ paths, we can reformulate (5) as

$$\ell^* = \min_{x \in \mathcal{P}} \left\{\ell \ : \ \ell \geq c^T x, \forall c \in \mathcal{U}\right\}. \tag{6}$$

Given $x \in \mathcal{P}$, the worst case length of x is given by $\max_{c \in \mathcal{U}} c^T x$. Hence, the robust counterpart admits the following description:

$$\ell^* = \min_{x \in \mathcal{P}} \left\{\max_{c \in \mathcal{U}} c^T x\right\}. \tag{7}$$

A path with worst case length ℓ^* is called a robust shortest path (RSP) and the problem of finding a RSP is called a robust shortest path problem (RSPP).

3.2 The robust counterpart of SPP with box uncertainty

In this subsection we assume that the uncertainty set $\mathcal{U}$ is a box, i.e., $\mathcal{U}$ is defined as follows

$$\mathcal{U} = \{c \,:\, l \leq c \leq u\}, \tag{8}$$

where l and u are two vectors in $\mathbf{R}^{\mathcal{A}}$ with $l \leq u$. Now, obviously,

$$\max_{c\in\mathcal{U}} c^T x = u^T x.$$

Hence, by (7) we have

$$\ell^* = \min_{x\in\mathcal{P}} u^T x. \tag{9}$$

This shows us that the robust counterpart of the uncertain SPP with box uncertainty is a SPP with the original arc lengths replaced by their upper bounds. An immediate consequence is the following result

Theorem 2. *In case of the SPP with box uncertainty set, every sub path of the robust shortest path is a robust shortest path.*

A special case of box uncertainty occurs when the uncertainty is relative to the nominal values, i.e., when the uncertainty set has the form

$$\mathcal{U} = \{c \,:\, |c - c^n| \leq \gamma c^n\}, \tag{10}$$

for some $\gamma \geq 0$. Then we have for each $x \in \mathcal{P}$:

$$\max_{c\in\mathcal{U}} c^T x = (1+\gamma)(c^n)^T x. \tag{11}$$

Hence, the RSP is just a (usual) shortest path (with respect to c^n), but its length is multiplied by $(1+\gamma)$.

3.3 The robust counterpart of SPP with ellipsoidal uncertainty

In this section we consider an ellipsoid uncertainty set centered at c^n, the vector of nominal arc lengths. We assume that $\mathcal{U}$ has the form

$$\mathcal{U} = \{c : c = c^n + Qw, \|w\| \leq 1\}, \tag{12}$$

where Q is a fixed matrix of size $|\mathcal{A}| \times p$ and $w \in \mathbf{R}^p$ for some p.

Given $x \in \mathcal{P}$, one may easily verify that the worst case length $c^T x, c \in \mathcal{U}$, is attained when

$$w = \frac{Q^T x}{\|Q^T x\|}. \tag{13}$$

Hence, denoting the worst-case length of x as ℓ_x, we have

$$\ell_x = \max_{c\in\mathcal{U}} c^T x = (c^n)^T x + \|Q^T x\|. \tag{14}$$

This implies that (7) becomes

$$\ell^* = \min_{x\in\mathcal{P}} \ell_x = \min_{x\in\mathcal{P}}\{(c^n)^T x + \left\|Q^T x\right\|\}. \tag{15}$$

This can be formulated as

$$\min\left\{\ell_x \;:\; \ell_x \geq (c^n)^T x + \left\|Q^T x\right\|, Ax = b, x \in \{0,1\}^{\mathcal{A}}\right\}. \tag{16}$$

Since $\left\|Q^T x\right\| \leq \ell_x - (c^n)^T x$ is a conic quadratic constraint, this is a conic quadratic optimization (CQO) problem with binary variables. This becomes apparent if we rewrite (16) as follows

$$\begin{aligned} \min\ & \ell_x \\ & \|z\| \leq \alpha, \\ & \ell_x - c^T x - \alpha = 0, \\ & Q^T x - z = 0, \\ & Ax = b, x \in \{0,1\}^{\mathcal{A}}. \end{aligned} \tag{17}$$

The first constraint is a conic quadratic constraint, followed by three linear constraints and a binary constraint. Due to the binary constraint, its solution requires a branch and bound scheme. An example of RSPP is presented in the next section.

3.4 An example

Consider the network of Figure 1, where the nominal arc lengths are as indicated.

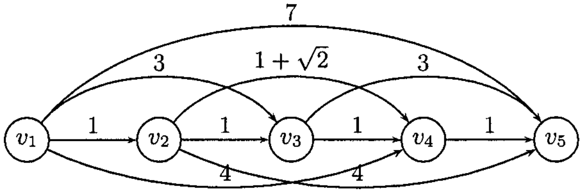


Fig. 1. A simple network

Assume that $Q = rI$, where I is the identity matrix and $r \geq 0$, then the uncertainty set is given by

$$\mathcal{U} = \{c \;:\; c = c^n + rw, \|w\| \leq 1\}, r \geq 0. \tag{18}$$

Due to (13), for a given $x \in \mathcal{P}$ the worst-case lengths is attained when $w = \frac{x}{\|x\|}$, and according to (14) we have

$$\ell_x(r) = (c^n)^T x + r\,\|x\|\,, r \geq 0. \tag{19}$$

Table 1. All paths from v_1 to v_5 and their worst-case lengths.

i	x_i	$\ell_{x_i}(r)$
1	$v_1 - v_5$	$7 + r$
2	$v_1 - v_2 - v_5$	$5 + r\sqrt{2}$
3	$v_1 - v_3 - v_5$	$6 + r\sqrt{2}$
4	$v_1 - v_4 - v_5$	$5 + r\sqrt{2}$
5	$v_1 - v_2 - v_3 - v_5$	$5 + r\sqrt{3}$
6	$v_1 - v_2 - v_4 - v_5$	$3 + \sqrt{2} + r\sqrt{3}$
7	$v_1 - v_3 - v_4 - v_5$	$5 + r\sqrt{3}$
8	$v_1 - v_2 - v_3 - v_4 - v_5$	$4 + 2r$

All paths from $s = v_1$ to $t = v_5$ and their worst-case lengths are presented in Table 1. We conclude that the shortest worst case length is given by

$$\begin{aligned}\ell^*(r) &= \min\{7 + r, 5 + r\sqrt{2}, 6 + r\sqrt{2}, 5 + r\sqrt{3}, 3 + \sqrt{2} + r\sqrt{3}, 4 + 2r\}.\\ &= \min\{7 + r, 5 + r\sqrt{2}, 3 + \sqrt{2} + r\sqrt{3}, 4 + 2r\}.\end{aligned}$$

We see from Figure 2 that the robust shortest path length is a piecewise linear concave function of r with four different intervals where $\ell^*(r)$ depends linearly on r. On each interval we have a different RSP (see Table 2).

Table 2. RSP's from v_1 to v_5 and their worst-case lengths.

Interval	RSP's	$\ell_x(r)$
$0 \leq r \leq 1.5458$	$v_1 - v_2 - v_3 - v_4 - v_5$	4+2r
$1.5458 \leq r \leq 1.8431$	$v_1 - v_2 - v_4 - v_5$	$3 + \sqrt{2} + r\sqrt{3}$
$1.8431 \leq r \leq 4.8284$	$v_1 - v_2 - v_5$, $v_1 - v_4 - v_5$	$5 + r\sqrt{2}$
$r \geq 4.8284$	$v_1 - v_5$	$7 + r$

We generalize the result of above example in the following theorem, where we introduce a scaling parameter r in the definition of the matrix $\mathcal{Q}$ in (12).

Theorem 3. *Let $\mathcal{U}$ be the ellipsoidal uncertainty set given by*

$$\mathcal{U} = \{c \ : \ c = c^n + (r\mathcal{Q})w, \|w\| \leq 1\}, r \geq 0 \tag{20}$$

where r is a scaling parameter. Then the robust shortest path length is a piecewise linear concave function of r.

Proof. For a given path $x \in \mathcal{P}$, the worst case length of x depends linearly on r:

$$\ell_x(r) = (c^n)^T x + r \left\|\mathcal{Q}^T x\right\|, r \geq 0. \tag{21}$$

By (15), the robust shortest path length is a piecewise linear concave function of r. □

In the next section, we use the same example as before to show that a subpath of a robust shortest path is not necessarily a robust shortest path.

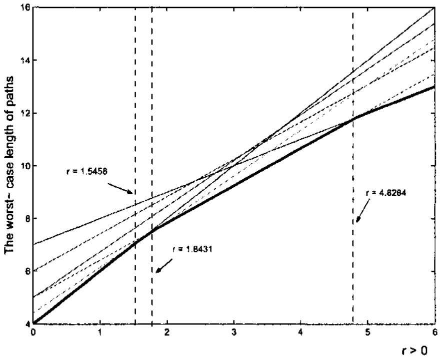


Fig. 2. The piecewise linear concave function $\ell^*(r)$.

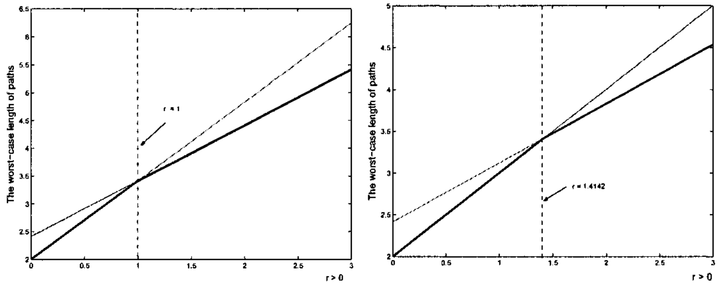
3.5 Subpath of RSP's

It is interesting to ask if a subpath of a robust shortest path is a robust shortest path. The answer to this question is negative. This is shown below by using the example of section 3.4.

In Table 2, we see that if $0 \leq r \leq 1.5458$, the subpath of the RSP $v_1 - v_2 - v_3 - v_4 - v_5$ from v_2 to v_4 is $v_2 - v_3 - v_4$. The worst case length of this path is $2 + r\sqrt{2}$. There is one other path, namely $v_2 - v_4$, whose worst case length is $1 + \sqrt{2} + r$. Therefore, the RSP from v_2 to v_4 has length

$$\min(2 + r\sqrt{2}, 1 + \sqrt{2} + r),$$

as shown in Figure 3.*a*. Hence we conclude that for $0 \leq r \leq 1$ the subpath $v_2 - v_3 - v_4$ is a RSP, but if $1 \leq r \leq 1.5458$ then it is not a RSP from v_2 to v_4.



(a)

(b)

Fig. 3. $\ell_x(r)$ from v_2 to v_4

An alternative approach might be as follows. Fix the lengths of the arcs outside the subpath $v_2 - v_3 - v_4$ to their worst case value, and then find the shortest worst case length path from v_2 to v_4.

See Figure 4. We fix $c_{12} = c_{45} = 1 + \frac{r}{2}$ such that the perturbation vector w from v_2 to v_4 has $w_{12} = w_{45} = \frac{1}{2}$. Then the worst case length of $v_2 - v_3 - v_4$ becomes $2 + r$ and the worst case length $v_2 - v_4$ becomes $1 + \sqrt{2} + \frac{r}{\sqrt{2}}$. The RSP has length

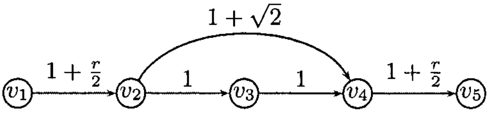


Fig. 4. The subpaths from v_2 to v_4

$$\min(1 + \sqrt{2} + \frac{r}{\sqrt{2}}, 2 + r),$$

as shown in Figure 3.*b*. Hence we conclude that for $0 \leq r \leq \sqrt{2}$ the subpath $v_2 - v_3 - v_4$ is a RSP, but if $\sqrt{2} \leq r \leq 1.5458$ then it is not a RSP from v_2 to v_4. From this example we may conclude that a subpath of a robust shortest path is not necessarily a robust shortest path.

4 Conclusion

The RSPP is simple in case of box uncertainty. It then is a SPP with all arc lengths at their upper bounds. In case of ellipsoidal uncertainty the RSPP becomes a conic quadratic problem with binary constraints. In the first case, subpaths of robust shortest path are itself robust shortest paths, but this property does not hold in the second case, where it deserves further investigation.

References

1. A. Ben-Tal and A. Nemirovski. (1999) Robust solutions of uncertain linear programs. Oper. Res. Lett., **25(1)**1–13.
2. A. Ben-Tal and A. Nemirovski. (2001) Lectures on Modern Convex Optimization. Analysis, Algorithms and Engineering Applications. Volume 1 of MPS/SIAM Series on Optimization. SIAM, Philadelphia,USA.
3. A. Ben-Tal and A. Nemirovski. (2002) Robust optimization—methodology and applications. Math. Program, **92**(3, Ser. B): 453–480.
4. O.E. Karasan, M.C. Pinar, H.Yaman. (2001) The Robust Shortest Path Problem with Interval Data. Optimization Online. $http : //www.optimization - online.org/DB_HTML/2001/08/361.html.$
5. D.Bertsimas, M. Sim. (2003) Robust Discrete Optimization and Network Flows, Math. Program., Ser. B. **98**: 49-71.

Approximation Algorithms for Finding a Maximum-Weight Spanning Connected Subgraph with given Vertex Degrees *

Alexey E. Baburin and Edward Kh. Gimadi

Sobolev Institute of Mathematics, prospekt Akademika Koptyuga 4, 630090
Novosibirsk, Russia

Abstract. In the paper a problem of finding a maximum-weight spanning connected subgraph with given vertex degrees is considered. The problem is MAX SNP-hard, because it is a generalization of a well-known Traveling Salesman Problem. Approximation algorithms are constructed for deterministic and random instances. Performance bounds of these algorithms are presented.

1 Introduction

Let $G(V, E)$ be a complete n-vertex undirected graph without loops with a non-negative weight function w of edges. There are known integers d_i $(i = 1, \ldots, n), 1 \leq d_i \leq n$.

In [6] the problem of finding a realization of a set of integers as degrees of the vertices in a subgraph G' of G is formulated. (This set is called a graphical partition of a number p, where $p = \sum_{i=1}^{n} d_i$). It is clear that for every such realization the number p is even and $d_i \leq n-1$ for each $i = 1, \ldots, n$. These conditions are not sufficient. For example, a set $D = (3, 3, 3, 1)$ is not a graphical partition. A constructive criterion of realizability of a set of integers is implied from the following statement presented in [7]:

Theorem 1. *Let $n > d_1 \geq d_2 \geq \ldots \geq d_n$ and $\sum_{i=1}^{n} d_i$ is even. Then a partition $D = (d_1, \ldots, d_n)$ is graphical if and only if a modified partition $D' = (d_2 - 1, d_3 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n)$ is graphical.*

An optimization appearance of the problem was described in [2] by means of reduction to the problem of finding a maximum-weight matching.

The problem of finding a maximum-weight spanning connected subgraph with given vertex degrees appeared in [4] and was denoted by *CSSDP*. Denote by $W^*(G)$ the weight of an optimal solution for an instance G of the problem, by $W_A(G)$ – the weight of a solution presented by approximation algorithm $\bar{A}$ for the same instance.

* This research was supported by the Russian Foundation for Basic Research (grant 02-01-01153), program of supporting of leading science schools of Russia (project "Nauchnaya Shkola - 313.2003.1"), and INTAS (grant 00-217)

$\Delta_A = \min_G \frac{W_A(G)}{W^*(G)}$ is called *a performance ratio* of algorithm A.

An approximation algorithm was presented in [4] for the metric case of $CSSDP$. The algorithm only worked for the case when all integers d_i are even. The performance ratio of the algorithm exceeded $1 - \frac{1}{d(d+1)}$, where $d = min\{d_i | i = 1, \dots, n\}$. In this paper we present an approximation algorithm and show its performance ratios for different classes of the problem $CSSDP$.

An algorithm for random instances of the problem was also presented in [4]. We improve the performance estimates by presenting a modification of the algorithm and show its asymptotical optimality for a wider class of input instances.

Different applications of the problem and its variations appeared in [1].

2 Approximation algorithm A' for deterministic instances of $CSSDP$

Below we consider a complete n-vertex undirected graph $G = (V, E)$ without loops with a non-negative weight functions w of edges. Integers d_i $(i = 1, \dots, n), 2 \le d_i \le n$ are given.

2.1 Algorithm A' description

We propose an algorithm A' for solving $CSSDP$ on the graph. First, it constructs an optimal solution for a relaxation of the problem (without the requirement of connectivity). Then the components of connectivity are patched into one component that is a feasible solution for the problem.

Stage 1. Using Gabow's algorithm [3] a spanning subgraph G' of G with given vertex degrees is found.

Stage 2. Components of connectivity $C_1, \dots, C_\mu$ are found. The components are reordered so that the minimum edge of G' belongs to C_1. If $\mu = 1$ then the subgraph G' is the output of algorithm A', else the stage 3 is performed.

Stage 3. For every component C_i $(i = 1, \dots, \mu)$ a set S_i of edges is formed. The set S_1 consists of all edges in C_1. For $i > 1$, S_i consists of all edges in C_i that do not form a component of 2-connectivity (that are not *bridges*). In every set S_i an edge $e_i = (u_i, v_i)$ of minimum weight is found.

Stage 4. Let $i = 1$. Put $p_{\mu+1} = q_1 = u_1$, $p_1 = v_1$.

Stage 5. If $w(q_i, u_{i+1}) \ge w(q_i, v_{i+1})$ put $q_{i+1} = v_{i+1}$, $p_{i+1} = u_{i+1}$. Otherwise put $q_{i+1} = u_{i+1}$, $p_{i+1} = v_{i+1}$.

Stage 6. Put $i = i + 1$. If $i < \mu - 1$ the stage 5 is performed, else the stage 7 is performed.

Stage 7. If $w(q_{\mu-1}, u_\mu) + w(u_\mu, p_{\mu+1}) \ge w(q_{\mu-1}, v_\mu) + w(v_\mu, p_{\mu+1})$ put $q_\mu = v_\mu$, $p_\mu = u_\mu$. Otherwise put $q_\mu = u_\mu$, $p_\mu = v_\mu$.

Stage 8. Edges $e_1, \dots, e_\mu$ are removed from G'.

Stage 9. Edges $(q_1, p_2), (q_2, p_3), \dots, (q_\mu, p_{\mu+1})$ are added to G'.

End of the algorithm A'.
Subgraph G' is the output of algorithm A'.

2.2 Algorithm A' analysis

Lemma 1. *Algorithm A' is finite, deterministic and produces a feasible solution of CSSDP.*

Proof. Algorithm A' is finite because the only loop of it (on stages 5–6) repeats for exactly $\mu - 2$ times and $\mu \leq n$. On stage 8 a set of edges is removed from G' that lowers the degrees of two different vertices of each component of connectivity by one. On stage 8 a set of edges is added to G' that raises the degrees of the same vertices by one. This proves that the degrees of the vertices are decided on stage 1, where they are made equal to the input values of $d_1, \ldots, d_n$. On stage 9 all components of connectivity are connected. This makes a loop between the components (or maybe a chain in case a bridge edge was chosen as e_1). This shows that the constructed subgraph is connected. The proof is complete.

Lemma 2. *Algorithm A' performs in $O(n^3)$ running time.*

Proof. The running time of the algorithm is determined on stage 1. Gabow's algorithm performs in $O(n^3)$ running time. The proof is complete.

Let $G(V, E)$ be a complete n-vertex undirected graph without loops. A non-negative weight function w of edges is called *metric* if it satisfies the triangular inequality.

Theorem 2. *Algorithm A' guarantees the performance ratio for a solution of CSSDP such that*

$$\Delta_{A'} \geq 1 - \frac{2}{d(d+1)},$$

where $d = min\{d_i \mid i = 1, \ldots, n\}$.

Proof. Let W' be the summary weight of the edges of G'. $W' \geq W^*(G)$ because G' is the optimal solution for a relaxation of the problem.

$$W_{A'}(G) = W' - \sum_{i=1}^{\mu} w(e_i) + \sum_{i=1}^{\mu} w(q_i, p_{i+1}).$$

Each component of connectivity C_i has at least $d + 1$ vertices. Thus it has at least $d(d-1)/2$ edges. In case it has a bridge (more than one component of 2-connectivity) it has more than $d(d-1)/2$ edges that are not bridges. So each set S_i consists of at least $d(d-1)/2$ edges. Edge e_i has the minimum weight in the set S_i.

$$\sum_{i=1}^{\mu} w(e_i) \leq \frac{2}{d(d+1)} W'.$$

We conclude that

$$\frac{W_{A'}(G)}{W^*(G)} \geq \frac{W_{A'}(G)}{W'} \geq 1 - \frac{2}{d(d+1)}.$$

The proof is complete.

Theorem 3. *For metric CSSDP, algorithm A' guarantees the performance ratio such that*

$$\Delta_{A'} \geq 1 - \frac{1}{d(d+1)},$$

where $d = min\{d_i \mid i = 1, \ldots, n\}$.

Proof. Because of the triangular inequality and choice of vertices p_i, q_i we get

$$w(e_i) \leq 2w(q_{i-1}, p_i)$$

for each $i = 2, \ldots, \mu - 1$.

Each component of connectivity C_i has at least $d+1$ vertices. Thus it has at least $d(d-1)/2$ edges. In case it has a bridge (more than one component of 2-connectivity) it has more than $d(d-1)/2$ edges that are not bridges. So each set S_i consists of at least $d(d-1)/2$ edges.

$$\sum_{i=1}^{\mu} w(e_i) \leq \frac{2}{d(d+1)} W'.$$

At the same time the edge e_1 has the least weight of all edges in G' before stage 8. Thus

$$w(e_1) \leq w(e_\mu).$$

The choice of p_μ, q_μ and the triangular inequality yields

$$w(e_\mu) \leq w(q_{\mu-1}, p_\mu) + w(q_\mu, p_{\mu+1}).$$

Recall that W is the summary weight of the edges of G'. $W' \geq W^*(G)$ because G' is the optimal solution for a relaxation of the problem. We continue

$$W_{A'}(G) = W' - \sum_{i=1}^{\mu} w(e_i) + \sum_{i=1}^{\mu} w(q_i, p_{i+1})$$

$$= W' - \sum_{i=2}^{\mu-1} (w(e_i) - w(q_{i-1}, p_i)) - w(e_1) + w(q_{\mu-1}, p_\mu) + w(q_\mu, p_{\mu+1}) - w(e_\mu)$$

$$\leq W' - \frac{1}{2} \sum_{i=2}^{\mu-1} w(e_i) - w(e_1) \leq W' - \frac{1}{2} \sum_{i=1}^{\mu} w(e_i).$$

We conclude that

$$\frac{W_{A'}(G)}{W^*(G)} \geq \frac{W_{A'}(G)}{W'} \geq 1 - \frac{1}{d(d+1)}.$$

The proof is complete.

3 Approximation algorithms for random instances of regular *CSSDP*

In this section we consider *CSSDP* with random weights of edges of an undirected graph G and $d_1 = d_2 = \ldots = d_n = d$.

Let us use the notations presented in [5] to construct algorithms with performance estimates $(\varepsilon_n, \delta_n)$. Suppose algorithm A for solving an optimization problem. By $F_A(I)$, $F^*(I)$ we denote the values of the objective function when solving an individual problem I by algorithm A and precisely respectively.

Algorithm A *has performance estimates* $(\varepsilon_n, \delta_n)$ in the class K_n of n-dimensional maximization problems if for all n the following inequality is satisfied:

$$\mathbf{Pr}\{1 - \frac{F_A(I)}{F^*(I)} > \varepsilon_n\} \le \delta_n.$$

Here ε_n is called *a relative error*, δ_n is *a fault probability* of algorithm A, $\mathbf{Pr}\{J\}$ is the probability of a corresponding event J.

An algorithm is called *asymptotically optimal* in the problem class $K = \cup\{K_n | n = 1, 2, \ldots\}$ if there exist its performance estimates ε_n, δ_n such that $\varepsilon_n \longrightarrow 0$, $\delta_n \longrightarrow 0$ as $n \longrightarrow \infty$.

Without loss of generality suppose $n = m(d-1)$, where m is an integer number. Split the matrix w_{ij} of size $n \times n$ into $(d-1)^2$ square matrices

$$W_{rs} = \{(w_{ij}) \mid r - 1 < \frac{i}{m} \le r, \; s - 1 < \frac{j}{m} \le s\},$$

where $1 \le r, s < d$.

We modify the algorithm $\widetilde{A}$ presented in [4] as follows:

Stage 1. Find approximate solutions for the maximum Traveling Salesman Problem $\mathcal{TSP}_r$ on matrices W_{rr}, $1 \le r < d$ using "Farthest City" heuristic.

Stage 2. Find approximate solutions for the generalized maximum Assignment Problem $\mathcal{AP}_r, 1 \le r < d-1$. Here a problem $\mathcal{AP}_r$ stands for choosing r elements in each row 1 in each column on a block of matrices $W_{d-r-1,s}$, $d - r \le s < d$. The solution is found by selecting r maximum elements in each row (one by one in each column).

Stage 3. Take the chosen elements of the matrix (w_{ij}) as the output of the algorithm.

End of algorithm $\widetilde{A}$.

Note that the algorithm constructs a connected spanning regular subgraph G' in $O(n^2)$ running time.

3.1 Probabilistic analysis of algorithm $\widetilde{A}$ for uniform distribution

The result of the section relies on Petrov's theorem [8]:

Theorem 4. *Consider independent random variables $X_1, \ldots, X_n$ and let $S = \sum_{k=1}^{n} X_k$. Let there be positive constants $g_1, \ldots, g_n$ and T such that*

$$\mathbf{E}e^{tX_k} \leq e^{\frac{1}{2}g_k t^2} (k = 1, \ldots, n)$$

for all $0 \leq t \leq T$. Let $U = \sum_{k=1}^{n} g_k$. Then

$$\mathbf{Pr}\{S > x\} \leq e^{-x^2/2U}, 0 \leq x \leq UT,$$

$$\mathbf{Pr}\{S > x\} \leq e^{-Tx/2}, x \geq UT.$$

We will also need the result of the following lemma:

Lemma 3. *Let ξ_k be a minimum between k random variables with identical distribution function $\mathbf{Pr}\{\xi < x\} = x$, $0 \leq x \leq 1$. Then for an arbitrary $y \geq \frac{\pi^2}{3} - 2$ and a sum $S = \sum_{k=1}^{m} \tilde{\xi}_k$ of variables $\tilde{\xi}_k = \xi_k - \mathbf{E}\xi_k$, $k = 1, \ldots, m$ the following inequality holds:*

$$\mathbf{Pr}\{S \geq y\} \leq e^{-y/2}. \tag{1}$$

Proof. We need some auxiliary facts which are valid for a distribution function of ξ_k :

$$F_{\xi_k}(x) = 1 - (1 - x)^k.$$

(a) For each integer $k \geq 1$, $\mathbf{E}\xi_k = 1/(k+1)$.

(b) For each integer $k \geq 1$,

$$\mathbf{E}e^{t\xi_k} = k \sum_{i=0}^{\infty} \frac{t^i}{k(k+1)\cdots(k+i)}.$$

(c) For each $t \leq 1$,

$$\frac{t}{k+1} \leq \frac{1}{2}.$$

(d) For any $0 \leq \alpha \leq 1/2$,

$$\frac{e^{-\alpha}}{1 - \alpha} \leq 1 + \alpha^2.$$

Facts (a) and (b) are used in the following conclusions:

$$\mathbf{E}e^{t\tilde{\xi}_k} = e^{-t/(k+1)}\mathbf{E}e^{t\xi_k} \leq \frac{e^{-t/(k+1)}}{1 - t/(k+1)}.$$

Using the facts (c) and (d) we get

$$\mathbf{E}e^{t\tilde{\xi}_k} \leq 1 + \left(\frac{t}{k+1}\right)^2 \leq e^{g_k t^2/2},$$

where $g_k = 2(k+1)^{-2}$.

Let us denote the sum of all g_k, $1 \le k \le m$, by U. Then

$$U = 2\sum_{i=2}^{m+1} \frac{1}{k^2} \le 2\sum_{i=1}^{\infty} \frac{1}{k^2} - 2 \le \frac{\pi^2}{3} - 2.$$

For this g_k, U, and $T = 1$ random variables $\widetilde{\xi}_k$ satisfy Petrov's Theorem and thus we conclude

$$\mathbf{Pr}\{S \ge y\} \le e^{-y/2}.$$

The proof is complete.

Let us describe a class $\mathcal{K} = \{\mathcal{K}_n\}$ of the problem with the distance matrix (w_{ij}), whose entries are selected independently from the segment $[a_n, b_n]$, $a_n \ge 0$, with the same uniform distribution function $P(x) = \mathbf{Pr}\{\xi < x\}$ of random variable $\xi = (c_{ij} - a_n)/(b_n - a_n)$, $0 \le \xi \le 1$.

Theorem 5. *Let $y \ge \frac{\pi^2}{3} - 2$. Algorithm $\widetilde{A}$ constructs an approximate d-regular solution of CSSDP with performance estimates ε_n, δ_n for the class $\mathcal{K}$ where*

$$\varepsilon_n = \left(y + \ln(m+1) + \frac{2}{d}\right)\frac{b_n - a_n}{mb_n},$$

$$\delta_n = e^{-y/2}.$$

Proof. Let S_r denote the summary weight of all edges chosen on stage 1 for the problem $\mathcal{TSP}_r$. Then $S_r = ma_n + (b_n - a_n)(\sum_{i=1}^{m-1} \Psi_i^r + \xi^r)$, where Ψ_k^r is a maximum between k independent random variables with identical distribution function $P(x) = x$, $0 \le x \le 1$, and ξ^r is another random variable with this distribution.

Let Q_r denote the summary weight of all edges chosen on stage 2 for the problem $\mathcal{AP}_r$. Then

$$Q_r = mra_n + (b_n - a_n)\sum_{i=1}^{m} \zeta^r_{r,(m-1)r},$$

where $\zeta^r_{j,k}$ is a sum of j maximum values in a set of k independent random variables with identical distribution function $P(x) = x$, $0 \le x \le 1$.

For the value of the optimal solution $F^*(I)$ we have

$$F^*(I) \le \frac{ndb_n}{2}.$$

The value $F_{\widetilde{A}}(I)$ of the solution, obtained by algorithm $\widetilde{A}$ on instance I, is equal to

$$F_{\widetilde{A}}(I) = \sum_{r=1}^{d-1} S_r + \sum_{r=1}^{d-2} Q_r.$$

To get the performance estimates of the algorithm we conclude

$$\mathbf{Pr}\left\{1 - \frac{F_{\widetilde{A}}(I)}{F^*(I)} > \varepsilon_n\right\} \leq \mathbf{Pr}\left\{\sum_{i=1}^{m} \xi_i \geq \frac{\varepsilon_n m b_n}{b_n - a_n} - \frac{2}{d}\right\}.$$

For

$$y = \frac{\varepsilon_n m b_n}{b_n - a_n} - \frac{2}{d} - \ln(m+1)$$

we get

$$\mathbf{Pr}\left\{1 - \frac{F_{\widetilde{A}}(I)}{F^*(I)} > \varepsilon_n\right\} \leq \mathbf{Pr}\left\{\sum_{i=1}^{m} \tilde{\xi}_i \geq y\right\}.$$

We use the result of Lemma 3 and get the performance estimates ε_n and δ_n for $y \geq \frac{\pi^2}{3} - 2$, where

$$\varepsilon_n = \left(y + \ln(m+1) + \frac{2}{d}\right)\frac{b_n - a_n}{m b_n}, \quad \delta_n = e^{-y/2}.$$

The proof is complete.

Theorem 6. *Algorithm $\widetilde{A}$ is asymptotically optimal for the problem of finding d-regular solution of CSSDP for the class $\mathcal{K}$ if $d = o(n)$.*

Proof. Put $y = 2\ln(m+1)$. The previous statement guarantees performance estimates ε_n, δ_n for algorithm $\widetilde{A}$ where

$$\varepsilon_n = \left(3\ln(m+1) + \frac{2}{d}\right)\frac{b_n - a_n}{m b_n} \leq \frac{3\ln(m+1) + 1}{m},$$

$$\delta_n = \frac{1}{m+1}.$$

Since $d = o(n)$ we have $m = n/(d-1) \to \infty$ as $n \to \infty$. Hence we conclude that $\varepsilon_n \to 0$, $\delta_n \to 0$ as $n \to \infty$. The proof is complete.

3.2 The case of a minorizing type distribution function

By $\mathcal{K}^0$ denote the class $\mathcal{K}$ with the distribution function $P(x) \leq x$, $0 \leq x \leq 1$.

Theorem 7. *Algorithm $\widetilde{A}$ is asymptotically optimal for the problem of finding d-regular solution of CSSDP for the class $\mathcal{K}^0$ if $d = o(n)$.*

Proof. Denote by $F^0_{\widetilde{A}}$ the objective value of the solution obtained by algorithm $\widetilde{A}$ for instance of the problem from $\mathcal{K}^0$.

Note that the objective obtained by algorithm $\widetilde{A}$ is a linear function of variables Ψ_i^r, Ψ_i^{rs} and ξ^r from Theorem 4. Any of this variables (say ψ for class $\mathcal{K}$ and ϕ for class $\mathcal{K}^0$) is a maximum of a certain number of independent random variables with the same distribution function $P(x)$. Then for arbitrary $y \geq 0$ the following inequality holds:

$$\mathbf{Pr}\{\phi \geq y\} \geq \mathbf{Pr}\{\psi \geq y\}.$$

Thus it can be shown that for arbitrary $y \geq 0$ the following inequality is satisfied:

$$\mathbf{Pr}\{F^0_{\tilde{A}} \geq y\} \geq \mathbf{Pr}\{F_{\tilde{A}} \geq y\}.$$

Thus the results of Theorem 4 stand for the class $\mathcal{K}^0$ also, i.e. the algorithm $\tilde{A}$ is asymptotically optimal. The proof is complete.

References

1. R. G. Busacker, T. L. Saaty: Finite Graphs and Networks: an Introduction with Applications, McGraw Hill, NewYork (1965).
2. J. Edmonds, E. L. Johnson: Matchings: a well solvable class of integer linear programs, *Combinatorial Structures and their Applications*, Gordon and Breach, New York (1970), 89–92.
3. H. N. Gabow: An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems, *Proceedings of the 15th annual ACM symposium on theory of computing, Boston, Apr. 25–27*, ACM, New York (1983), 448–456.
4. E. Kh. Gimadi, A. I. Serdukov: A problem of finding the maximal spanning connected subgraph with given vertex degrees, *Oper. Res. Proc. 2000*, Springer Verlag (2001), 55–59.
5. E. Kh. Gimadi, N. I. Glebov, V. A. Perepelitsa: Algorithms with estimates for discrete optimization problems, *Problemy Kibernetiki*, Nauka, Moskva, vol.31 (1976), 35–42, in Russian.
6. F. Harary: Graph Theory, *Addison-Wesley, Reading,* Massachusetts (1969).
7. V. Havel: A note to question of existance of finite graphs, *Casopis Pest Mat.,* **80** (1955), 477–480.
8. V. V. Petrov: Limit Theorems of Probability Theory. Oxford Univ. Press (1995).

Multiprocessor Scheduling Problem with Stepwise Model of Job Value Change

Adam Janiak and Tomasz Krysiak

Institute of Engineering Cybernetics, Wroclaw University of Technology,
Janiszewskiego 11/17, 50-372 Wrocław, Poland
Adam.Janiak@pwr.wroc.pl, Tomasz.Krysiak@pwr.wroc.pl

Abstract. The paper deals with a scheduling problem on the parallel processors, in which the sum of values of all the jobs is maximized. The value of job is characterized by a stepwise non-increasing function. Establishing an order of processing of datagrams which are sent by multiprocessor router is a practical example of application of this problem. It was already proved that its single processor case is NP-hard, thus the problem is also NP-hard. Therefore, two pseudo-polynomial time algorithms for the problem with common moments of job value change and a polynomial time algorithm for the case with identical processing times are constructed. It is also constructed and experimentally tested a number of heuristic algorithms which solve the general version of the problem.

1 Introduction

The paper deals with a scheduling problem on the parallel processors, in which the sum of values of all the jobs is maximized and a job value is described by a non-increasing stepwise function. Establishing an order of processing of datagrams which are sent by router is an application example of the problem, if a multiprocessor router is given. The precise description of this situation (but not for multiprocessor case) is given in [2], since a single processor case of this problem was considered there. It was proved in the mentioned paper, that the problem is NP-hard. Moreover, a pseudo-polynomial time algorithm for some special case, and a number of heuristic algorithms for general single-processor version of the problem, were constructed. In this paper, based on the results presented in [2], we construct two pseudo-polynomial time algorithms and some heuristics for parallel processor cases of the problem. Thus, this paper is an extension of [2].

The remaining part of the paper is organized as follows. In the next section we formulate the problem and next we prove polynomially solvable cases of the problem. Section 4 deals with the pseudo-polynomial time algorithms constructed for the cases with common moments of job value change and in Section 5 we present and experimentally compare some heuristic algorithms constructed to solve the general version of the problem. Some concluding remarks are given in Section 6.

2 Problem Formulation

There are given a set of m identical processors $M = \{M_1, \ldots, M_m\}$ and a set of n independent and non-preemptive jobs $J = \{J_1, \ldots, J_n\}$, immediately available for processing at time 0. Each processor can execute at the most one job at a time. Whereas, each job i is characterized by its processing time $p_i > 0$, its value $v_i(t)$ dependent on the time t and the moments $d_{ij} > 0$, $j = 1 \ldots k-1$ at which a change of job value occur. The model of job value is given by non-increasing stepwise function defined as follows

$$v_i(t) = \begin{cases} w_{i1}, & 0 < t \leq d_{i1} \\ w_{i2}, & d_{i1} < t \leq d_{i2} \\ \vdots & \\ w_{ik}, & d_{ik-1} < t \end{cases},$$

where $w_{i1} > w_{i2} > \ldots > w_{ik}$.

Solution of the problem is represented by a set of permutations $\Pi = \{\pi_1, \ldots, \pi_l, \ldots, \pi_m\}$, where π_l denotes a schedule of jobs assigned to execute on a processor M_l. Let n_l denotes a number of jobs processed on M_l (then $\sum_{l=1}^{m} n_l = n$). The objective is to find such an assignment of jobs to the processors and such a schedule of jobs on the processors $\Pi = \{\pi_1, \ldots, \pi_m\}$, for which the sum of job values, calculated at their completion times $C_{\pi_l(i)}$ (if a job is processed on the i^{th} position of the permutation π_l), is *maximal*:

$$\sum_{l=1}^{m} \sum_{i=1}^{n_l} v_{\pi_l(i)}(C_{\pi_l(i)}) \Rightarrow \max.$$

Therefore, using the three-field notation $\alpha \mid \beta \mid \gamma$ for scheduling problems [1], the problem considered in the paper is given by

$$P \left| v_i(C_i) = \begin{cases} w_{i1}, & 0 < C_i \leq d_{i1} \\ w_{i2}, & d_{i1} < C_i \leq d_{i2} \\ \vdots & \\ w_{ik}, & d_{ik-1} < C_i \end{cases} \right| \sum v_i(C_i).$$

3 Polynomially Solvable Cases

Property 1 *Problem $P\,|p_i = p|\sum v_i(C_i)$, where $v_i(C_i)$ are some arbitrary functions, can be solved optimally in $O(n^3)$ time.*

Proof. Since $p_i = p$, thus there exist exactly n moments $t_j = \lceil \frac{j}{m} \rceil \cdot p$ $(j = 1, \ldots, n)$, at which the jobs complete. Therefore, the problem $P\,|p_i{=}p|\sum v_i(C_i)$ is solved by finding an optimal assignment of the jobs to these moments. This amounts to formulating and solving an $n \times n$ usual maximization assignment problem with cost coefficients $c_{ij} = v_i(t_j)$. It is known that the assignment problem can be solved in $O(n^3)$ steps ([3]). □

In similar way we can solve the problem with uniform parallel processors $Q\left|p_i = p\right|\sum v_i(C_i)$, where processing time of job i executed on a processor l is given as $p_{il} = p/s_l$ (s_l denotes a given *speed* of processor M_l). The only difference is to determine the moments $t_1, \dots, t_n$: for each processor M_l, $l = 1, \dots, m$, initialize $b_l = p/s_l$; then for $j = 1, \dots, n$: find a processor l^* such that $b_{l^*} = \min_{l=1,\dots,m}\{b_l\}$, set $t_j := b_{l^*}$ and then $b_{l^*} := b_{l^*} + p/s_{l^*}$. This operation can be done in $O(n \log m)$ time ([4]), thus the problem is solvable also in $O(n^3)$ steps.

4 Pseudo-Polynomial Time Algorithms

In this section, we present two pseudo-polynomial time algorithms solving two following problems:

$$P\left|v_i(C_i) = \begin{cases} w_{i1}, \ 0 < C_i \le d_1 \\ w_{i2}, \ d_1 < C_i \le d_2 \\ 0, \qquad d_2 < C_i \end{cases}\right|\sum v_i(C_i), \tag{1}$$

$$P\left|v_i(C_i) = \begin{cases} w_{i1}, \ 0 < C_i \le d_1 \\ w_{i2}, \ d_1 < C_i \le d_2 \\ w_{i3}, \ d_2 < C_i \le d_3 \\ w_{i4}, \quad d_3 < C_i \end{cases}\right|\sum v_i(C_i). \tag{2}$$

Let t_i' and t_i'' denote, for both problems, sum of processing times of jobs completed, on the processor i, before d_1 and between d_1 and d_2, respectively. Let t_i''' denote also sum of processing times of jobs completed, on the processor i, between d_2 and d_3, only for the problem (2). Assume also, the jobs in both problems are numbered according to LPT rule, i.e., $p_1 \ge p_2 \ge, \dots, \ge p_n$. Thus, we will consider the jobs in natural order $1, 2, \dots, n$. For the problem (1), we can observe that there are three possible choices of schedule for each job j ($j = 1, \dots, n$) on any processor i ($i = 1, \dots, m$): (i) job j completes before d_1 (then t_i' increases by p_j and the criterion value increases by w_{j1}); (ii) job j completes after d_1 but before d_2 (then t_i'' increases by p_j and the criterion value increases by w_{j2}); (iii) job j completes after d_2 (then t_i', t_i'' and the criterion value do not increase). Thus, for the problem (1) let $F_j\left((t_1', t_1''), \dots, (t_m', t_m'')\right)$ denote maximum sum of job values (i.e., criterion value) of subproblem involving the jobs $1, \dots, j$, where the sum of processing times of jobs completed, on the processor i, before d_1 and between d_1 and d_2, are equal to t_i' and t_i'', respectively ($i = 1, \dots, m$). Now, we can calculate recurrently $F_j\left((t_1', t_1''), \dots, (t_m', t_m'')\right)$ for all possible $(t_1', t_1''), \dots, (t_m', t_m'')$ (i.e., for $t_i' = 0, \dots, d_1$ and $t_i'' = 0, \dots, d_2 - t_i'$, $i = 1, \dots, m$) for $j = 1, \dots, n$. Hence, if we calculate $F_n\left((t_1', t_1''), \dots, (t_m', t_m'')\right)$ for all possible $(t_1', t_1''), \dots, (t_m', t_m'')$, then we obtain the optimal criterion value F^* (as the maximum from among these ones). The formal description of the pseudo-polynomial time algorithm

(PPolynl) for the problem (1) is given as follows.

Algorithm PPolynl

Step 1: *Renumber jobs according to LPT rule, i.e.,* $p_1 \geq p_2 \geq, \ldots, \geq p_n$. *Set*

$$F_j\left((t_1', t_1''), \ldots, (t_m', t_m'')\right) := \begin{cases} 0 & \textit{if } j, t_1', t_1'', \ldots, t_m', t_m'' = 0, \\ -\infty & \textit{otherwise}, \end{cases}$$

for $j = 0, \ldots, n$, $t_i' = 0, \ldots, d_1$ *and* $t_i'' = 0, \ldots, d_2 - t_i'$, $i = 1, \ldots, m$.
Then set $j := 1$.

Step 2: *For each* $t_i' = 0, \ldots, d_1$ *and* $t_i'' = 0, \ldots, d_2 - t_i'$ $(i = 1, \ldots, m)$ *calculate:*

$$F_j\left((t_1', t_1''), \ldots, (t_m', t_m'')\right) = \max_{1 \leq i \leq m} \begin{cases} F_{j-1}\left((t_1', t_1''), \ldots, (t_i' - p_j, t_i''), \ldots, (t_m', t_m'')\right) + w_{j1}, \\ \qquad \text{if } t_i' \geq p_j \\ F_{j-1}\left((t_1', t_1''), \ldots, (t_i', t_i'' - p_j), \ldots, (t_m', t_m'')\right) + w_{j2}, \\ \qquad \text{if } t_i'' \geq p_j \text{ and } t_i'' < (d_2 - d_1) + p_j \\ F_{j-1}\left((t_1', t_1''), \ldots, (t_i', t_i''), \ldots, (t_m', t_m'')\right) \end{cases}.$$

If $j = n$, *then go to Step 3, otherwise set* $j := j+1$ *and repeat Step 2.*

Step 3: *Find the optimal schedule by backtracking.*

Property 2 *The problem* $P \left| v_i(C_i) = \begin{cases} w_{i1}, & 0 < C_i \leq d_1 \\ w_{i2}, & d_1 < C_i \leq d_2 \\ 0, & d_2 < C_i \end{cases} \right| \sum v_i(C_i)$ *can be solved optimally in pseudo-polynomial time* $O\left(n\left(d_1 d_2\right)^m\right)$ *by the algorithm PPolynl.*

Proof. Since in Step 2 of PPolynl we check all possible choices of schedule for each job j, thus, we obtain all possible values of criterion. Thus, the algorithm finds optimal solution of the problem. Now, it is enough to calculate only a computational complexity of the algorithm. Since there are given n jobs and m processors, and we have d_1 different values of t_i' and at most d_2 different values of t_i'' $(i = 1, \ldots, m)$, thus Step 1 and Step 2 require $O\left(n\left(d_1 d_2\right)^m\right)$ operations and Step 3 requires n operations. Thus, the whole algorithm PPolynl requires $O\left(n\left(d_1 d_2\right)^m\right)$ time. □

In the problem (2), similarly to the problem (1), there are four possible choices of schedule for each job j on processor i $(i = 1, \ldots, m)$: (i) job j completes before d_1 (then t_i' increases by p_j and the criterion value increases by w_{j1}); (ii) job j completes between d_1 and d_2 (then t_i'' increases by p_j and the criterion value increases by w_{j2}); (iii) job j completes between d_2 and d_3 (then t_i''' increases by p_j and the criterion value increases by w_{j3}); (iv) job j completes after d_3 (then t_i', t_i'' and t_i''' do not increase and the criterion value increases by w_{j4}). Then we can define as $F_j^\gamma\left((t'_1,t''_1,t'''_1),\dots,(t'_m,t''_m,t'''_m)\right)$ maximum sum of values of jobs $1,\dots,j$, where the sum of processing times of jobs completed, on the processor i, before d_1, between d_1 and d_2 and between d_2 and d_3 are equal to t'_i, t''_i and t'''_i, respectively ($i=1,\dots,m$). Additionally, $\gamma=\{\gamma_1,\dots,\gamma_i,\dots,\gamma_n\}$ is a set of jobs, in which γ_i denotes a job scheduled as the first from among these ones on a processor i, which are completed between d_1 and d_2. The optimal criterion value F^* we obtain by calculating $F_n^\gamma\left((t'_1,t''_1,t'''_1),\dots,(t'_m,t''_m,t'''_m)\right)$ for all possible combinations of γ (there are $\binom{n}{m}$ such combinations) and all possible $(t'_1,t''_1,t'''_1),\dots,(t'_m,t''_m,t'''_m)$ ($t'_i=0,\dots,d_1$, $t''_i=0,\dots,d_2-t'_i$ and $t'''_i=0,\dots,d_3-(t'_i+t''_i)$, $i=1,\dots,m$). The formal description of the pseudo-polynomial time algorithm (PPolyn2) for the problem (2) is given as follows.

Algorithm PPolyn2

Step 1: *Renumber the jobs according to LPT rule, i.e.,* $p_1\ge p_2\ge,\dots,\ge p_n$.

Set $F_j^\gamma((t'_1,t''_1,t'''_1),\dots,(t'_m,t''_m,t'''_m)):=\begin{cases} w_{\gamma_i 2} \ \textit{if}\ j,t'_i,t'''_i=0,t''_i=p_{\gamma_i}, \\ \qquad i=1,\dots,m,p_{\gamma_i}\le d_2 \\ -\infty \ \textit{otherwise}\end{cases}$,

for all combinations of γ *and* $t'_i=0,1,\dots,d_1$; $t''_i=0,1,\dots,\min\{d_2-t'_i,d_2-d_1+p_{\gamma_i}\}$; $t'''_i=0,1,\dots,d_3-(t'_i+t''_i)$, $i=1,\dots,m$. *Then set* $j:=1$.

Step 2: *For all combinations of* γ *and for each* $t'_i=0,1,\dots,d_1$; $t''_i=0,1,\dots,$ $\min\{d_2-t'_i,d_2-d_1+p_{\gamma_i}\}$; $t'''_i=0,1,\dots,d_3-(t'_i+t''_i)$ $(i=1,\dots,m)$ *calculate if* $t''_i+t'''_i<d_3-d_1+p_{\gamma_i}$:

$$F_j^\gamma\left((t'_1,t''_1,t'''_1),\dots,(t'_m,t''_m,t'''_m)\right)=$$

$$\max_{1\le i\le m}\begin{cases} F_{j-1}^\gamma((t'_1,t''_1,t'''_1),\dots,(t'_i-p_j,t''_i,t'''_i),\dots,(t'_m,t''_m,t'''_m))+w_{j1}, \\ \qquad \text{if } t'_i\ge p_j \text{ and } j\ne\gamma_i \\ F_{j-1}^\gamma((t'_1,t''_1,t'''_1),\dots,(t'_i,t''_i-p_j,t'''_i),\dots,(t'_m,t''_m,t'''_m))+w_{j2}, \\ \qquad \text{if } t''_i\ge p_j \text{ and } j\ne\gamma_i \\ F_{j-1}^\gamma((t'_1,t''_1,t'''_1),\dots,(t'_i,t''_i,t'''_i-p_j),\dots,(t'_m,t''_m,t'''_m))+w_{j3}, \\ \qquad \text{if } t'''_i\ge p_j \text{ and } t'''_i<(d_3-d_2)+p_j \text{ and } j\ne\gamma_i \\ F_{j-1}^\gamma((t'_1,t''_1,t'''_1),\dots,(t'_i,t''_i,t'''_i),\dots,(t'_m,t''_m,t'''_m))+w_{j4}\end{cases}$$

If $j=n$, *then go to Step 3, otherwise set* $j:=j+1$ *and repeat Step 2.*

Step 3: *Find the optimal schedule by backtracking.*

Property 3 *The problem* $P\left|v_i(C_i)=\begin{cases} w_{i1}, & 0<C_i\le d_1\\ w_{i2}, & d_1<C_i\le d_2\\ w_{i3}, & d_2<C_i\le d_3\\ w_{i4}, & d_3<C_i\end{cases}\right|\sum v_i(C_i)$ *can be solved optimally in pseudo-polynomial time* $O\left(n\binom{n}{m}(d_1d_2d_3)^m\right)$ *by the algorithm PPolyn2.*

Proof can be done similarly to the proof of Property 2.

5 Heuristic Algorithms for the General Case of the Problem

For the general version of the problem we constructed and experimentally tested a number of heuristic algorithms, which deliver approximate solutions in sensible (i.e., polynomial) time. General scheme of the examined algorithms consists of the following two steps.

Step 1: Construct initial list of jobs L by sequencing the jobs according to one of the following three rules:

$L1$ – in non-increasing order of w_{i1};
$L2$ – in non-decreasing order of p_i;
$L3$ – in non-decreasing order of p_i/w_{i1}.

Step 2: Take the jobs in sequence from the list L and assign them to the processors, according to one of the following two rules:

$P1$ – put the current job to the earliest available processor;
$P2$ – put the current job to the processor, on which the value of this job is maximal.

Now, the algorithms can be described by giving only the rules Li and Pj, $i=1,2,3$, $j=1,2$ (e.g., algorithm $L2P1$ puts jobs in the list L in non-decreasing order of p_i and then assigns jobs from this list in sequence to the earliest available processor). Thus, we obtain six algorithms, which computational complexity is equal to $O(n\log n)$.

The algorithms were tested for $n=9$, 50, 100 and 500 jobs and for $m=2$, 4, 10 and 25 processors. 500 randomly instances were generated for each n and m (except $n=9$ and $m=10$ and 25, for which tests are pointless). The problem parameters were randomly generated according to the uniform distribution from the following intervals: $p_i\in(0,90)$; $w_{i1}\in[1,100)$; $w_{ij}\in(0,w_{ij-1})$ for $j=2,\dots,k$; $d_{ij}\in[d_{ij-1}+100,d_{ij-1}+200)$ for $j=1,\dots,k$ and $i=1,\dots,n$. For $n=9$, for each generated instance the optimal solution was found by explicit enumeration (let OPT denote the criterion value for this solution). Then, for each algorithm we calculated a performance ratio: $(OPT/ALG\ -\ 1)\cdot 100\%$, where ALG denotes an objective function value obtained by the considered algorithm. For $n > 9$, for each generated instance of the considered problem the algorithm which gave a solution with the largest criterion value was found. Let A_{BEST} denote the largest criterion value. Then, the following performance ratio $(A_{BEST}/ALG - 1) \cdot 100\%$ was calculated for the other algorithms. The average values of the performances defined above, obtained for all 500 generated instances of the problem, with a given n and m, are shown in Table 1.

Table 1. Average performance ratios for the considered algorithms

	$L1P1$	$L2P1$	$L3P1$	$L1P2$	$L2P2$	$L3P2$
n			$m = 2$			
9	5.74	6.51	5.03	2.68	5.38	**2.46**
50	34.76	22.58	8.17	33.16	13.26	**3.09**
100	49.73	19.79	7.86	43.71	13.73	**4.98**
500	159.66	19.37	12.85	137.33	13.68	**6.93**
			$m = 4$			
9	0.89	1.89	0.66	**0.03**	1.23	0.41
50	18.19	20.16	11.12	8.69	12.45	**0.12**
100	39.62	18.72	11.82	23.47	9.16	**4.45**
500	156.19	28.16	11.96	130.55	17.27	**2.25**
			$m = 10$			
50	6.17	21.03	5.16	**0.26**	10.63	0.32
100	19.99	22.09	12.47	5.88	11.97	**0.27**
500	63.77	26.45	12.08	43.42	14.75	**0.98**
			$m = 25$			
50	0.22	0.91	0.20	**0.00**	**0.00**	**0.00**
100	6.77	10.41	7.09	**0.07**	3.71	0.19
500	36.16	23.29	12.05	18.19	11.06	**0.36**

It is worth noticing at first that, for $n = 9$ jobs, all algorithms deliver solutions very close to the optimum (generally up to 6 percent worse than optimal ones, but often even 1 percent or less). Moreover, constructing list of jobs in non-decreasing order of p_i/w_{i1} (rule $L3$) gives better solutions than two other methods (i.e., $L1$ and $L2$). Similarly, it is better to put the jobs to the processors, on which the value of these jobs are maximal (rule $P2$) than to the earliest available processors (rule $P1$). Thus, as you can see in the tabel above, the best method is $L3P2$. It is easy to notice also that the algorithms deliver better solutions if there are given more processors.

6 Conclusions

In this paper, we considered a problem of scheduling jobs on parallel processors in order to maximize the sum of job values. The job value is characterized by a stepwise non-increasing function. The problem is NP-hard since its single processor case is proved to be NP-hard. Thus, it is highly unlikely to construct an optimal algorithm solving the considered problem in polynomial time. Hence, based on the results obtained for single processor case, we constructed two pseudo-polynomial time algorithms for the problems with two and three common moments of job value change and also a number of heuristic algorithms for the general version of the problem. The computational complexity of the pseudo-polynomial time algorithms is $O\left(n\left(d_1d_2\right)^m\right)$ and $O\left(n\binom{n}{m}\left(d_1d_2d_3\right)^m\right)$ (where d_i, $i \in \{1,2,3\}$, denote the moments of job value change). As you can see, the problem with two moments of job value change is significantly easier to solve than the problem with three moments. However, the both algorithms are more efficient for the instances with small number of processors (m) and for small values of d_i. For the general version of the problem we constructed six algorithms with computational complexity equal to $O(n \log n)$, which derive the solutions up to 6 percent worse than the optimal ones.

References

1. Graham R. L., Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G. (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics **5**, 287-326
2. Janiak A., Kasperski A., Krysiak T. (2004) Scheduling jobs with a stepwise function of change of their values. In Ahr D., Fahrion R., Oswald M., Reinelt G. (Eds.), Operations Research Proceedings 2003, Selected Papers of the International Conference on Operations Research (OR 2003), Heidelberg, September 3-5, 2003, 363-370
3. Lawler E. L. (1976) Combinatorial optimization: Networks and Matroids. Holt, Rinehart & Winston, New York
4. Lawler E. L. and Lenstra J. K. and Rinooy Kan A. H. G. and Shmoys D. B. (1993) Sequencing and scheduling: Algorithms and complexity. In Graves S. C., Rinnooy Kan A. H. G., and Zipkin P. H. (Eds.), Handbooks in Operations Research and Management Science, **4**, 445-522

The Prize Collecting Connected Subgraph Problem - A New NP-Hard Problem arising in Snow Removal Routing

P O Lindberg[1] and Gholamreza Razmara[1]

Linköping University, Linköping, SE-58183, Sweden

Abstract. We consider a new NP-hard optimization problem, the Prize Collecting Connected Subgraph Problem, which appears as a subproblem in routing of snow plows during snow fall. In this problem we have a set of edges in an undirected network, with edge costs and edge times. Moreover there is a time budget. The problem is to find a connected subset (corresponding to a snowplow tour) of minimal cost subject to the budget constraint. This problem can be modeled using flow constraints or introducing valid inequalities. We exemplify computations for the classical Sioux Falls Network

1 Introduction

In an *undirected* network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with *edge costs* c_e, nonnegative (i.e. expenditures) or positive (i.e. incomes), and *edge times* t_e, the *Prize Collecting Connected Subgraph Problem (PCCSP)* is to find to find the connected subgraph that maximizes income minus expenditures subject to a time budget. This problem arises as a column generation subproblem in periodic snow removal routing problems, [4].

In this paper we introduce and study the PCCSP. In particular we show that it is NP-hard, by reducing the *Prize Collecting Steiner Tree Problem* to it. We give various formulations of the PCCSP, based on flow constraints or valid inequalities. We further suggest a computational procedure for its solution, and demonstrate it on the Sioux Falls network.

The terminology in the Prize Collecting (PC) area is not very standardized. As to PC Steiner Trees (e.g [3] and the references therein) it is not customary to introduce a budget constraint. For PC TSPs though it seems customary to have some buget/minimal income constraint. Feillet et al [1] term the budget-free PC TSP a *profitable tour* problem, and suggest a terminology for this class of problems. In this terminology the PCCSP would be something like the *Profitable Connected Subgraph Problem with One Additional Constraint.*

2 Background: Snow Removal Routing under Snowfall

Suppose we are given a *directed* network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ with *arcs* $a \in \mathcal{A}$ corresponding to road segments to be plowed by snow clearing vehicles. For the case during (a long) snowfall, we want to design a set of cyclic snow clearing routes that can be run perpetually during the snowfall, and that cover all arcs in $\mathcal{A}_p$, the set of arcs to be plowed. To avoid plowing too often, some arcs in some routes may be used for transport. Further, the arcs in $\mathcal{A}_p$ have *minimal time spans* $\underline{t}_a$ between successive plowings, smaller for more important arcs (i.e. road segments).

To be more specific, associate with all arcs plowing costs c_a^P and times t_a^P as well as transportation costs c_a^T and times t_a^T. To streamline the model we will duplicate each arc into one plowing arc and one transportation arc, with corresponding costs and times c_a and t_a.

Let a *route* r be a directed cyclic path in the network, and let the *duration* of r be $t_r =_{df} \sum_{a\in\mathcal{A}} t_a$ and its cost $c_r =_{df} \sum_{a\in\mathcal{A}} c_a$. To be consistent with the minimal time spans, each route will be assigned a *period* $p_r \in \underline{T} = \{\underline{t}_a\}_{a\in\mathcal{A}}$, which is how often it is repeated. To be *feasible* the route must fulfill on the one hand $t_r \leq p_r$, and on the other hand $\underline{t}_a \geq p_r, \forall a \in r$.

Let $\mathcal{R}$ be the set of feasible routes and let $A_{ar} = 1$ if a is plowed in r, and 0 else. Then the problem to find the minimal cost set of routes covering all arcs can be stated as

$$(P)\{\min_x \sum_{r\in\mathcal{R}} c_r x_r \text{ s.t. } \{\sum_{r\in\mathcal{R}} A_{ar} x_r \geq 1, a \in \mathcal{A}_p; x_r \in \{0,1\}, r \in \mathcal{R}$$

Remark. We have used a set covering rather than a set partitioning formulation, since if an arc is covered twice, say, one of the plowings can be changed into a transport, giving a cheaper route.

Since $\mathcal{R}$ typically is too large to be enumerated efficiently, we will have to use column generation. To this end, let the rows in (P) get multipliers λ. (For notational simplicity we let transport arcs have $\lambda_a = 0$.) Thus the reduced cost for route r is $\bar{c}_r = c_r - \sum_a A_{ar}\lambda_a = \sum_{a\in r} c_a - \sum_{a\in r} \lambda_a = \sum_{a\in r} \bar{c}_a$, where $\bar{c}_a = c_a - \lambda_a$ is the reduced cost of a. Thus we have the following

Column Generation Subproblem:

(P_{CG}) Find feasible cyclic route with minimal reduced cost $\bar{c}_r =_{df} \sum_{a\in r} \bar{c}_a$.

3 Subproblems as PCCSPs

In practical operations, all road segments are cleared by the same vehicle in both directions. Thus, we will assume the same. To further simplify the subproblem, we will assume that the same is true also about transports.

Assumption 1. In any route, all road segments plowed or used as transport, are plowed or transported in both directions.

Remark. These assumptions lead to an essentially undirected network. In practice there are small one-way segments. These have to be modeled by undirected edges. The ensuing errors can be made small, though, [4]. These simplifications, however, allow more efficient solution of the simplified subproblems below.

The simplifications just made allow simple representation of cyclic routes.

Lemma 1. Under Assumption 1, cyclic routes correspond one-to-one to connected subgraphs in the undirected network.
Proof. This and other proofs are omitted for space reason. •
Remark. Strictly speaking, Assumption 1 allows for "silly" routes, e.g. using the same edge for transport twice in each direction. The one-to-one correspondance of the lemma, however, concerns non silly routes, where edges are used for transport at most once in each direction (and of course are plowed at most once also)
Corollary. We may assume that each route plows each edge at most once and uses it for transport at most once in each direction.

4 Modeling the Subproblems

As seen in the previous section the column generation subproblem can be stated as a problem in an undirected network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where each *edge* $e \in \mathcal{E}$ corresponds to two directed arcs in the directed network. In the subproblem the "reduced costs" $\bar{c}_a$ of the arcs are fixed. Thus, let the *edge cost* c_e and the *edge time* t_e of e be the sum of the reduced costs and times, respectively, of the arcs corresponding to e. Let the minimal time spans $\underline{t}_e$ be equal to those of the arcs (assumed equal). Thus the column generation subproblem is
($PCCSP$) Find a connected subgraph $\mathcal{S}$ minimizing the cost $c_{\mathcal{S}} =_{df} \sum_{e \in \mathcal{S}} c_e$ subject to the time budget $t_{\mathcal{S}} =_{df} \sum_{e \in \mathcal{S}} t_e \leq p$.
Here p is the period of the route we are going to generate. (We will tacitly assume that when we generate a route of period p, then only edges with $\underline{t}_e \geq p$ will be eligible.)
Generating columns it is more convenient to solve the *rooted* version ($PCCSP_r$) of $PCCSP$, where a *fixed edge* e_{fix} has to belong to $\mathcal{S}$.

$$(PCCSP_r) \qquad \min_{\mathcal{S} \subset \mathcal{E}} c_{\mathcal{S}} =_{df} \sum_{e \in \mathcal{S}} c_e \tag{1}$$

$$\mathcal{S} \ \ connected$$

$$t_{\mathcal{S}} =_{df} \sum_{e \in \mathcal{S}} t_e \leq p$$

$$e_{fix} \quad \in \mathcal{S}$$

In what follows, modeling and solution techniques are given for $PCCSP_r$ unless otherwise stated. There are several ways to state $PCCSP_r$ as a mathematical programming problem. The most straightforward probably is by introducing single commodity flows to enforce connectedness.
Let $x_e = 1$ if e is in $\mathcal{S}$ and 0 else. Hence, if $x_{\bar{e}} > 0$, $\bar{e}$ needs to be connected to e_{fix}. Therefore, for each edge $\bar{e}$ in $\mathcal{E}$ we send the amount $x_{\bar{e}}$ of flow from e_{fix}

to $\bar{e}$, assuming edges have capacities x_e. To this end introduce an arbitrary direction for each e, and let f_e be the flow in that direction. Let s_e and d_e be start and end nodes of e, for this orientation, and $F_n = \{e \in \mathcal{E} | s_e = n\}$ and $B_n = \{e \in \mathcal{E} | d_e = n\}$, the corresponding forward and backward stars. We further locate the demand of e to its start node s_e. Let the *support* of $x \in \mathcal{R}^{\mathcal{E}}$ be $\mathcal{S}_x =_{df} \{e \in \mathcal{E} | x_e \neq 0\}$, and let $x^{\mathcal{S}} = (x_e^{\mathcal{S}})_{e \in \mathcal{E}}$ with $x_e = 1$ if $e \in \mathcal{S}$ and 0 else, be the *indicator vector* of $\mathcal{S}$.

In solving $(PCCSP_r)$ we will use LP relaxations. The straightforward single-commodity formulation will cause the corresponding bounds to be quite weak, and the LP solutions to be quite fractional. For this reason we will use the following non-simultaneous *multi-terminal flow* formulation. For each edge $\bar{e}$ we introduce a single commodity flow $f^{\bar{e}}$ with demand $x_{\bar{e}}$, from e_{fix} to $\bar{e}$. These flows all (non-simultaneously) use the edges e of capacities x_e.

$$(P_{MTF}) \qquad \min_{x,f} \sum_{e \in \mathcal{E}} c_e x_e \tag{2}$$

$$\sum_{e \in F_n} f_e^{\bar{e}} - \sum_{e \in B_n} f_e^{\bar{e}} = \left\{ \begin{array}{ll} x_{\bar{e}} & n = s_{e_{fix}} \\ -x_{\bar{e}} & n = s_{\bar{e}} \\ 0 & otherwise \end{array} \right\}, \quad \bar{e} \in \mathcal{E} \setminus \{e_{fix}\}$$

$$-x_e \leq f_e^{\bar{e}} \leq x_e$$

Lemma 3. x solves (P_{MTF}) if and only if $\mathcal{S}_x$ solves $(PCCSP_r)$. •

(P_{MTF}) gives better bounds and less fractional LP solutions than the single commodity formulation. It might, however, be quite cumbersome to solve. Therefore we will consider a 3^{rd} formulation.

For two sets of nodes, $\mathcal{X}$ and $\mathcal{Y}$, we let $(\mathcal{X}, \mathcal{Y}) =_{df} \{e \in \mathcal{E} | s_e \in \mathcal{X}, d_e \in \mathcal{Y}$ or $s_e \in \mathcal{Y}, d_e \in \mathcal{X}\}$ be the set of edges between $\mathcal{X}$ and $\mathcal{Y}$. Further we let $\tilde{\mathcal{X}}$ denote the complement $\mathcal{N} \setminus \mathcal{X}$ of $\mathcal{X}$. A *cutset* (or more commonly just a cut) is a set of the form $(\mathcal{X}, \tilde{\mathcal{X}})$. We say that the cutset *separates* the edges e and $\tilde{e}$ if $e \in (\mathcal{X}, \mathcal{X})$ and $\tilde{e} \in (\tilde{\mathcal{X}}, \tilde{\mathcal{X}})$, or vice versa. We further let $\mathcal{C}^{tot}(e, \tilde{e})$ denote the set of all cutsets separating e and $\tilde{e}$.

Let $\mathcal{X}$ be a subset of $\{0,1\}^{\mathcal{E}}$. An inequality $a^T x \leq \beta$ is a *valid inequality* for $\mathcal{X}$ if it is satisfied by all $x \in \mathcal{X}$. If (P) is an optimization problem with feasible set F in $\{0,1\}^{\mathcal{E}}$, then a valid inequality for (P) is one for F. Let $\mathcal{X}^{\mathcal{C}} =_{df} \{x \in \{0,1\}^{\mathcal{E}} | \mathcal{S}_x$ connected$\}$ be the set of indicator vectors of connected subgraphs of $\mathcal{G}$, and $\mathcal{X}_r^{\mathcal{C}}$ the corresponding set for subgraphs rooted in e_{fix}.

Lemma 4. For any pair of edges $\bar{e}, \tilde{e} \in \mathcal{E}$, and any cutset $\mathcal{C} \in \mathcal{C}^{tot}(\bar{e}, \tilde{e})$

$$\sum_{e \in \mathcal{C}} x_e \geq x_{\bar{e}} + x_{\tilde{e}} - 1 \tag{3}$$

is a valid inequality for $\mathcal{X}^{\mathcal{C}}$ (and a fortiori for $PCCSP$). •
Corollary. For any edge $\bar{e} \in \mathcal{E}$, and any cutset $\mathcal{C} \in \mathcal{C}^{tot}(e_{fix}, \bar{e})$,

$$\sum_{e \in \mathcal{C}} x_e \geq x_{\bar{e}} \tag{4}$$

is a valid inequality for $\mathcal{X}_r^{\mathcal{C}}$ (and a fortiori for $PCCSP_r$). •
Let us denote the inequality (4) by $C^{\#}(\mathcal{C}, \bar{e})$. In fact the inequalities (3) are not only valid for $\mathcal{X}^{\mathcal{C}}$, they describe $\mathcal{X}^{\mathcal{C}}$:
Proposition 1. $\mathcal{X}^{\mathcal{C}}$ is equal to
$\{x \in \{0,1\}^{\mathcal{E}} | x \text{ satisfies (3) } \forall \bar{e}, \tilde{e} \in \mathcal{E}, \forall \mathcal{C} \in \mathcal{C}^{tot}(\bar{e}, \tilde{e})\}$. •
Corollary. $\mathcal{X}_r^{\mathcal{C}}$ is equal to
$\{x \in \{0,1\}^{\mathcal{E}} | x \text{ satisfies } C^{\#}(\mathcal{C}, \bar{e}) \; \forall \bar{e} \neq e_{fix}, \forall \mathcal{C} \in \mathcal{C}^{tot}(e_{fix}, \bar{e})\}$. •
In view of this Corollary $PCCSP_r$ can be stated

$$(P_{VI}) \qquad \min_x \sum_{e \in \mathcal{E}} c_e x_e \tag{5}$$

$$\sum_{e \in \mathcal{C}} x_e \geq x_{\bar{e}} \qquad \bar{e} \in \mathcal{E} \setminus \{e_{fix}\}, \mathcal{C} \in \mathcal{C}^{tot}(e_{fix}, \bar{e})$$

$$x_e \in \{0,1\}$$

5 NP-hardness of PCCSP

A special case of $PCCSP$ is the *pure* or budget-free version. This can be viewed as a version where p is so large that the budget never is restraining. We prove the NP-hardness of $PCCSP$ by showing that the Prize Collecting Steiner Tree Problem, $PCSTP$, can be reduced to a pure $PCCSP$.
In the $PCSTP$ we have edge costs $c_e \geq 0$ and node penalties $\pi_n \geq 0$. The problem is to find a tree T minimizing the sum of the edge costs in T plus the sum of the penalties of the nodes not in T. The NP hardness of $PCSTP$ follows e.g. from that it can emulate the STP, which is known to be NP-hard. Thus let there be given a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with edge costs and node penalties, and consider the $PCSTP$ in $\mathcal{G}$. To model the $PCSTP$ by a $PCCSP$, we enlarge $\mathcal{G}$ by introducing for each $n \in \mathcal{N}$ a new node $n' \in \mathcal{N}'$, and a new edge $e_n \in \mathcal{E}'$ between n and n' of cost $c_{e_n} = -\pi_n$. We claim that the $PCSTP$ can be solved by solving the (pure) $PCCSP$ in $\mathcal{G}' = (\mathcal{N} \cup \mathcal{N}', \mathcal{E} \cup \mathcal{E}')$.
Proposition 2. The $PCCSP$ is NP-hard. •

6 Solving the *PCCSP*

We will solve $PCCSP$ by first solving the LP relaxation of (P_{VI}) and then use Branch&Bound or heuristics. (P_{VI}) has an exponential number of constraints, since the number of cutsets $(\mathcal{X}, \tilde{\mathcal{X}})$ is $\mathcal{O}(2^{|\mathcal{N}|})$ and for each of these we have $\mathcal{O}(|\mathcal{A}|)$ choices of edges in the valid inequality (4). Therefore, we will generate these valid inequalities, called *cuts* in this section, as needed. In choosing cuts we take some advice from [3]. According to their experience, one should try to generate several violated cuts (in their case subtour elimination constraints) at the same time. Moreover these cuts should be independent, i.e. have disjoint sets of edges. Further they found that simple *a priori cuts* can be very effective. We will try to use both these experiences.

Independent inequalities

Each of the cuts (3) or (4) can be viewed as a max-flow min-cut inequality for one commodity in the multi-terminal flow formulation (P_{MTF}). Gomory and Hu [2] show how to determine all min-cuts in such a situation, by solving $|\mathcal{N}| - 1$ max-flow problems. Since we do not have a code to generate all min-cuts, we will use standard max-flow min-cut computations to find violated cuts or verify that none exists.

Let $(\underline{P}_{VI})$ be the relaxation of (P_{VI}) where we have generated a subset of the cuts, and let $(\underline{LP}_{VI})$ be the LP relaxation of $(\underline{P}_{VI})$. Let $\bar{x}$ be the optimal solution to $(\underline{LP}_{VI})$. For a given $\bar{e} \in \mathcal{E}$ (typically with $\bar{x}_{\bar{e}} > 0$), let $(P^{MF}_{\bar{x},\bar{e}})$ be the problem to send maximal flow from e_{fix} to $\bar{e}$, assuming the edges has capacities $\bar{x}_e$.

$$(P^{MF}_{\bar{x},\bar{e}}) \qquad \max_{f,F} F \tag{6}$$

$$\sum_{e \in F_n} f_e - \sum_{e \in B_n} f_e = \begin{cases} F & n = s_{e_{fix}} \\ -F & n = s_{\bar{e}} \\ 0 & otherwise \end{cases}$$

$$-x_{\bar{e}} \leq f_e \leq x_{\bar{e}}$$

We then have the rather obvious result.

Proposition 3. Let $\bar{x}$ be the optimal solution to $(\underline{LP}_{VI})$, and let $\bar{F}_{\bar{e}}$ be the maximal flow F in $(P^{MF}_{\bar{x},\bar{e}})$ and $\bar{C}$ the corresponding min-cut.

If $\bar{F}_{\bar{e}} < \bar{x}_{\bar{e}}$, then the cut $C^{\#}(\bar{C}, \bar{e})$ is violated by $\bar{x}$.

If $\bar{F}_{\bar{e}} \geq \bar{x}_{\bar{e}}$, then all cuts $C^{\#}(C, \bar{e})$ for cutsets C separating e_{fix} and $\bar{e}$ are satisfied by $\bar{x}$. •

In view of this proposition we will term the cuts $C^{\#}(C, \bar{e})$ *max-flow cuts.*

Corollary. If $\bar{F}_{\bar{e}} \geq \bar{x}_{\bar{e}}$ for all $\bar{e}$ with $\bar{x}_{\bar{e}} > 0$, then there is no violated cut for $\bar{x}$, whence $\bar{x}$ is an optimal solution to (LP_{VI}).

Remark. When $\mathcal{S}_{\bar{x}}$ is not connected, violated cuts can of course be identified more simply, by identifying the components of $\mathcal{S}_{\bar{x}}$ through labeling. This is preferably done starting from components not containing e_{fix}, in this way yielding more independent cuts. (In a similar way max-flow cuts can be generated from edges e with $x_e > 0$.)

A Priori Cuts

To speed up the cut generation, we will in the beginning of the process add some simple a priori cuts:

1. For each edge $\bar{e}$ with $c_{\bar{e}} \leq 0$, we adjoin the cut $C^{\#}(\mathcal{C}, \bar{e})$, where $\mathcal{C} = (\mathcal{X}, \tilde{\mathcal{X}})$ with $\mathcal{X} = \{s_{\bar{e}}, d_{\bar{e}}\}$.
2. For each edge $\bar{e}$ with $c_{\bar{e}} > 0$, we adjoin the cuts $C^{\#}(\mathcal{C}, \bar{e})$, where $\mathcal{C} = (\mathcal{X}, \tilde{\mathcal{X}}) \setminus \bar{e}$ with $\mathcal{X} = \{s_{\bar{e}}\}$ or $\{d_{\bar{e}}\}$.

Solution Procedure

The procedure for solving $PCCSP_r$ can now be summarized as follows.
1. Let $(\underline{LP}_{VI})$ be the version with only a priori cuts and let $\bar{x}$ be its solution.
2. While $\mathcal{S}_{\bar{x}}$ is not connected identify violated cuts by labeling, adjoin them to $(\underline{LP}_{VI})$ and let $\bar{x}$ be the new solution.
3. While there are violated max-flow cuts, adjoin them to $(\underline{LP}_{VI})$ and let $\bar{x}$ be the new solution.
4. We have now solved (LP_{VI}), the LP relaxation of (P_{VI}). Solve (P_{VI}) using this information using exact or heuristic methods.

Note that if we branch on the x_e variables, when we solve (P_{VI}) by Branch-and-Bound, the candidate problems will be problems of the same type.

7 Computational Exemplification

For space reasons we limit our experiments to the rather small Sioux Falls (SF) network with 24 nodes and 38 edges, classical in Traffic Assignment. In [4] results are given for the larger real world Eskilstuna network.

In figure 1NW (North West) we display the SF network, where the fixed edge is (14,23). The numbers at the fat edges are the nonzero $\bar{x}_e$ values of the first LP solution for a given problem instance. Note that all $\bar{x}_e$ are 0 or 1, due to that the budget is not binding. We also display the first cuts (dashed), found by repeated connectivity labeling from the components of $S_{\bar{x}}$. (These cuts could also have been found by one application of the Gomory-Hu procedure.) Figure 1NE displays the situation after the introduction of the cuts in Fig. 1NW. Here dashed edges are transport edges. Note that we have fractional $\bar{x}_e$-values since the budget is now binding. Now $S_{\bar{x}}$ is connected, and we add max-flow cuts, displayed.

In Fig. 1SW we display the final LP solution (after the introduction of the cuts in Fig. 1NE). Now no max-flow cuts are violated, and we have the optimal solution to LP_{VI}. The heuristic integer solution is given in Fig. 1SE (achieved e.g. by forcing $x_{(14,15)}$ to 1.)

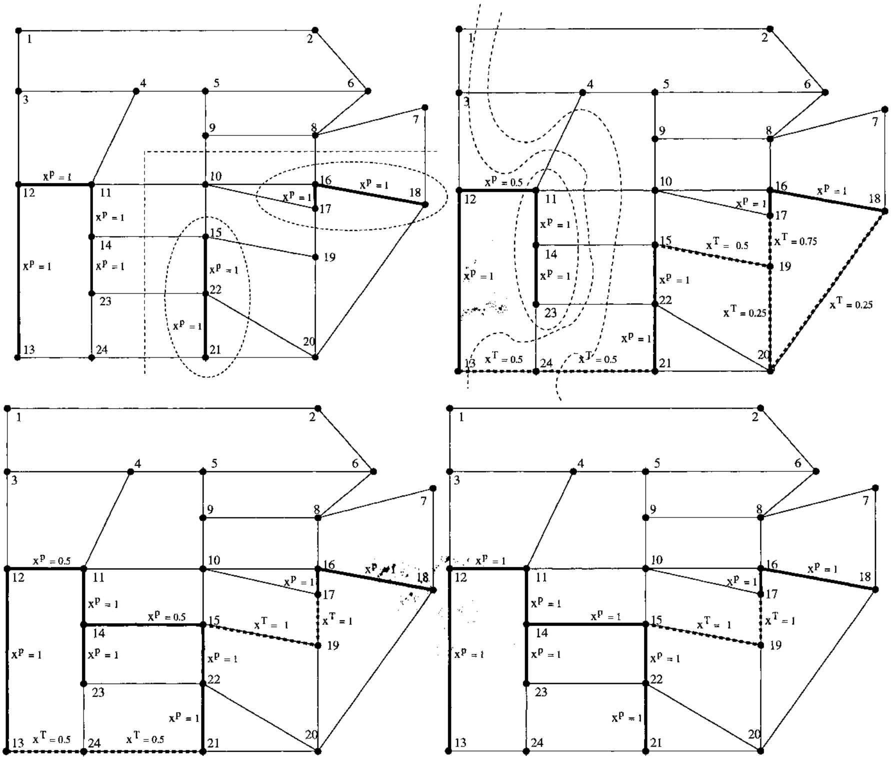


Fig. 1. LP solutions, cuts, and integer solution.

References

1. D. Feillet, P. Dejax, and M. Gendreaux (2004) Travelling Salesman Problems with Profits: an Overview. Laboratoire Informatique d'Avignon, to appear in *Transp. Sc.*
2. R.E.Gomory, and T.C. Hu (1961) Multi-terminal network flows. *J. Soc. Indust. Appl. Math.* **9** 551–570.
3. A. Lucena, and M. Resende (2004) Strong lower bounds for the prize collecting Steiner problem in graphs. *Discrete Applied Mathematics* **141** 277-294.
4. G. Razmara (2004) *Snow Removal Routing – Theory and Applications.* Linköping Studies in Science and Technology No. 888, Ph.D. Thesis, Dept of Mathematics, Linköping University.

A Less Flexibility First Based Algorithm for the Container Loading Problem

Yuen-Ting Wu[1], Yu-Liang Wu[2]

Department of Computer Science and Engineering, the Chinese University of Hong Kong, Hong Kong.

Abstract.

This paper presents a Less Flexibility First (LFF) based algorithm for solving container loading problems in which boxes of given sizes are to be packed into a single container. The objective is to maximize volume utilization. LFF, firstly introduced in [An effective quasi-human heuristic for solving the rectangle packing problem, European Journal of Operations Research 141 (2002) 341], is an effective deterministic heuristic applied to 2D packing problems and generated up to 99% packing densities. Its usage is now extended to the container loading problem. Objects are packed according to their flexibilities. Less flexible objects are packed to less flexible positions of the container. Pseudo-packing procedures enable improvements on volume utilization. Encouraging packing results with up to 93% volume utilization are obtained in experiments running on benchmark cases from other authors.

1. Introduction

Profit is a major concern in running business. One way to maximize profit is to minimize the cost. Transportation costs may greatly affect the overall costs. Effective management of transportation by maximizing the utilization of containers can contribute to an obvious reduction of costs. Various algorithms for container loading address this problem. A good algorithm can achieve high volume utilization of containers and in turn reduce the number of containers being used.

Many researchers have studied several variants of this problem. Classification of 3D packing problems can be done by types of boxes and by objective. The former classifies the problem into three types. In homogeneous cases, a single type of boxes (all with the same dimensions) is to be packed. A weakly heterogeneous box set refers to a few number of box types with a lot of individual boxes

[1] Email: ytwu@cse.cuhk.edu.hk

[2] Email: ylw@cse.cuhk.edu.hk

in each type. A strongly heterogeneous box set refers to a large number of box types with a few individual boxes in each type [1]. For objective classification, strip packing, bin packing and knapsack loading [6] are considered. The emphasis of this paper is on knapsack loading which is usually applied to the loading of cargoes into a container.

In knapsack loading problem, boxes are to be packed into a single container with fixed dimensions. A subset of these boxes is packed in an arrangement which maximizes a pre-defined profit. Some boxes can be left unpacked. When the profit is set to be the utilized volume, the objective will be minimization of wasted space. Volume utilization is calculated by: *Volume of occupied space / Total volume of the container* (V_o / V_t). Given a set of n rectangular-shaped boxes $\{b_1, b_2, b_3, \dots, b_n\}$, with known dimensions l_i x h_i x d_i for the i^{th} box, and a single rectangular-shaped container B with fixed dimensions L x H x D, a subset of the boxes should be packed without violating the following criteria:

- All edges of the packed boxes should be parallel to the container edges.
- No overlapping of boxes is allowed.
- All packed boxes must be completely stowed inside the container
- Orientation Constraint: Orientation defines horizontal or vertical placement of the box's surfaces. Orientation constraints states which sides of the box cannot be placed vertically. This restricts the rotation of the box and reduces the number of possible packing positions.
- Stability Constraint: Stability constraints require every packed box to be supported in a stable and balanced manner. Support can be provided by other packed boxes or by filling all empty space with foam rubber [6]. The former should ensure that *Area supported by underneath layer / Total base area >= a predefined value.* The latter method saves time for computing the area ratio and increases the number of possible packing positions. As the volume utilization is quite high, the space to be filled is not much. This approach is practical and is adopted in this paper for handling stability constraints.

Container loading problem is proven to be NP-hard [2]. No existing algorithms are able to give optimal solutions in polynomial time. Heuristics are mostly adopted in this kind of problems. Several heuristics using layering approach are developed. The advantage is mainly on balancing of load inside the container. [5] presents several ranking rules for selection of the most promising layer depths. [1] introduces a hybrid genetic algorithm based on layering. Genetic algorithms on non-layering methods are presented in [7]. Other methods include an integer programming model [4] and a parallel tabu search algorithm which increases diversity by exchanging local solutions [2].

We present a Less Flexibility First algorithm for this problem. Boxes are packed according to their flexibilities. Less flexible boxes are packed to the less flexible position of the container. By a series of pseudo-packing and greedy-packing process, a box is always packed to a position with the highest Fitness Cost Function Value (FFV). Final packing result is promising.

This paper is organized as follows. Section 2 describes the LFF principle. Section 3 presents the implementation of LFF on knapsack loading. Section 4 shows some experimental results on benchmark cases. Section 5 concludes this paper.

2. Rules of the Less Flexibility First Principle (LFFP)

2.1 Flexibility and Corners

The core idea of LFF algorithm is the concept of flexibility. The order of packing is determined by the flexibility of container space and flexibility of boxes.

- Flexibility of empty space follows this order: flexibility of a corner < flexibility of a side < flexibility of a central void area. This order is determined according to the freedom of move, i.e. the number of valid movements. In Fig. 1, black arrows show valid movements. Dashed arrows indicate invalid movements after which the packed object will change its original situation (e.g., an object at a corner leaves the corner).

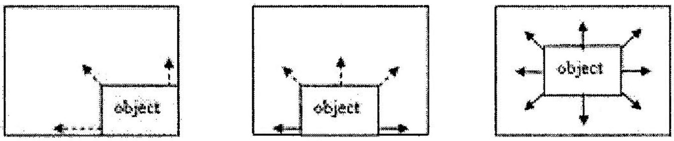


Fig. 1. Flexibility of a corner, a side and centre void area

- Flexibility of boxes depends on size and shape. Fig. 2 shows that there are fewer positions to accommodate a larger box. The smaller box Y can be packed into I, II or III while the bigger box X can only be packed into II.

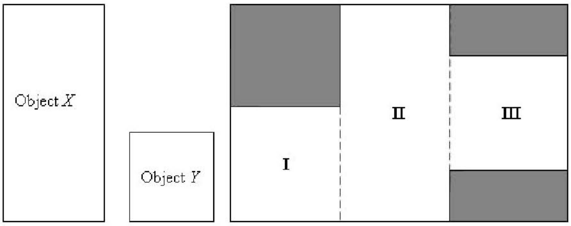


Fig. 2. Flexibility of boxes (Top view in 2D)

In the original LFF [12], the longest sides of the objects are compared to measure flexibility. In 3D LFF, the lengths of three sides are considered. For two boxes b_1 and b_2 with dimensions 100m x 80m x 30m and 90m x 70m x 60m respectively, if the longest side is used to determine flexibility, b_1 is less flexible. In fact, its shortest side is 30m while that of b_2 is 60m. Without orientation constraint, it is possible for b_1 to be packed to a narrow gaps of width less than 60m, but not for b_2. In most cases, the shortest sides contribute more to the flexibility than the longest. However, when the difference between shortest sides of two boxes is little, we should take the longest side into account.

Orientation constraint is another factor affecting flexibility. For two boxes of same dimensions, b_1 with orientation constraints and b_2 without orientation constraints, b_2 is obviously more flexible.

The top view of a corner *C* is shown in Fig.3. Draw a cross on the corner point to form four regions I, II, II and IV. The shaded regions are occupied by packed boxes while the white region is empty space. A corner must:

- have any three out of the four regions occupied by other boxes (or the boundary of the container). Four types of corners are shown.
- be in touch with the container base/top or lower/upper surface of another box
- be bounded by other three boxes in shaded regions in the *height*-dimension.

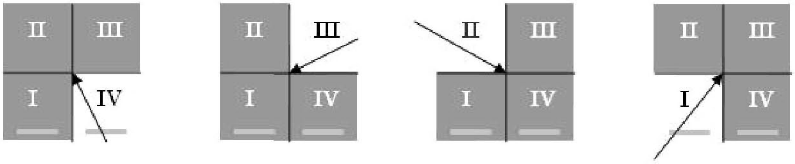


Fig. 3. Four different corners (Top-view)

2.2 The Less Flexible Principle (LFFP)

Chinese masons who perform packing task using the rule “Golden are the corners; silvery are the sides, and strawy are the voids”. That means empty corners should be filled first, which is followed by boundary sides and void areas come last. We derived our LFFP from this ancient packing strategy. It is thus regarded as *quasi-human* heuristic [12]. Corners are defined as the least flexible space for packing while larger boxes are less flexible boxes. LFFP packs less flexible boxes to the less flexible empty space. In other words, large boxes are packed first to the empty corners. Packing in a void area should be avoided as this creates empty space surrounding the packed box b_p. Such space may not be large enough for accommodating other boxes in later stages. b_p becomes an obstacle in this case.

There are corners in any packing configuration unless 100% volume utilization is achieved. In LFF, corners are the only candidates of space for packing a box. A packed box must occupy at least one corner in the packing configuration.

3. Implementation of LFF in Container Loading Problems

3.1 Generation of Corner Occupying Packing Move (COPM)

There can be more than one candidate corners for an unpacked box. The representation of a packing relationship between a box and a corner is defined as a corner occupying packing move (COPM), in the form of a tuple: <longest side, medium side, shortest side, orientation, x_l, y_l, z_l> where longest side, medium side and shortest side are the dimensions of the box being packed. Orientation states which sides of the box are placed along *x*, *y* and *z*-dimensions. It can be set to 0-5 (0 and 1 with shortest side placed vertically; 2 and 3 with medium side placed vertically; 4 and 5 with longest side placed vertically). x_l, y_l, z_l are the coordinates of the box's lower-front-left corner in a 3D coordinate system representing the container.

Given a current packing configuration in which all boxes are packed at fixed locations, the first step of LFF algorithm is to generate a COPM list representing all valid candidate packing positions for all unpacked boxes. When packing a box to a corner, without orientation constraints, six orientations are possible. For a packing configuration with six corners, a maximum of 6 x 6 = 36 COPMs can be generated. This COPM list is sorted in ascending order of flexibility in the way discussed in Section 2. As this list is processed from top to bottom, less flexible boxes are packed first, which obeys the principle of LFF.

For a box b_0 with dimensions 14m x 8m x 6m, orientation constraint states that the 14m side cannot be place vertically. The COPM at corner (0, 0, 0) will be <14, 8, 6, *ori*, 0, 0, 0> where *ori* = 0, 1, 2 or 3 (*ori* = 4 or 5 are invalid). COPMs at other corners are generated in the same manner.

3.2 Pseudo-Packing and the Greedy Approach

Each COPM for an unpacked box is then evaluated to choose the best as the final packing position. COPMs that cause overlapping of adjacent boxes or exceeding container boundary are deleted before evaluation takes place.

"Pseudo-Packing" means placing a box temporarily to a location specified by a COPM. In each pseudo-packing process, the least flexible unpacked box b_i is pseudo-packed to one COPM in the list. A Fitness Cost Function Value (FFV) is associated with every COPM of b_i for assessing the suitability of that COPM. The boxes left unpacked are "pseudo-packed" to the first available corner in the current packing configuration greedily, without violating the five criteria stated in Section 1, until no valid corners can be found for any of the unpacked boxes. The volume utilization (V_o / V_t) is the FFV of that COPM of b_i. Then all "pseudo-packed" boxes, including b_i, are removed from the container and pseudo-packing continues to evaluate other COPMs of b_i. An example is illustrated in Section 3.3.

3.3 Update of Corner List

Coordinates of corners are stored in a list for generation of COPMs. After "pseudo-packing" a box, at least one existing corner is occupied and some new corners are produced. The corner list must be updated before pseudo-packing the remaining boxes. Occupied corners are deleted while new corners are inserted.

Three boxes b_0, 14m x 8m x 6m; b_1, 12m x 6m x 5m and b_2, 9m x 9m x 4m are being packed into a container 20m x 15m x 10m. Some COPMs of b_0 is listed in Section 3.1. The FFV of COPM <14, 8, 6, 0, 0, 0, 0> is calculated by pseudo-packing b_0 at (0, 0, 0) with 14m side along x-dimension, 8m along y-dimension and 6m along z-dimension[3]. Corner at (0, 0, 0) is deleted from corner list. Corners at (14, 0, 0), (0, 8, 0) and (0, 0, 6) are inserted. b_1 can neither be packed at (0, 0, 6) nor (0, 0, 10). The first available corner for pseudo-packing b_1 greedily without overlapping is (0, 8, 0). New corners are (0, 8, 5), (0, 15, 5), (12, 8, 0) and (12, 15, 0). Fig. 4 shows the packing configuration.

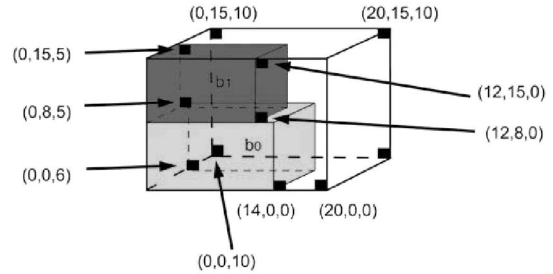


Fig. 4. Packing configuration after b_0 and b_1 are pseudo-packed (all corners are shown)

If there is no orientation constraints for b_2, it can be placed at (0, 0, 6) with 4m side along z-dimension. FFV is total volume of b_0, b_1 and b_2. All boxes are then removed. b_0 is pseudo-packed to another COPM till all COPMs are evaluated.

[3] x-dimension refers to the length of the container, y-dimension refers to the height of the container, z-dimension refers to the depth of the container

3.4 Real-Packing

After all COPMs of a box are evaluated, the box will be "real-pack" to the COPM with the highest FFV. The corner list is updated and the COPMs of the next unpacked box will undergo pseudo-packing. The process continues until no boxes can be packed to any corners. The LFF algorithm is shown in Fig. 5.

```
LFF algorithm for container loading:
Starting with a empty container
1.  Based on the current packing configuration, find all possible COPMs for each unpacked
    box; represent each COPM by a tuple <longest side, medium side, shortest side, orientation,
    $x_1, y_1, z_1$>.
2.  Sort all of these tuples according to their flexibility.
3.  For each candidate COPM, do 3.1 to 3.3 to find its fitness function value (FFV).
    3.1  Pseudo-pack this COPM.
    3.2  Pseudo-pack all the remaining boxes based on the current COPM list and with a
         greedy approach, until no more COPM can be packed.
    3.3  Calculate FFV of this candidate COPM as the occupied volume.
    Note: Before the pseudo-packing for the next candidate COPM is evaluated, the previously
          pseudo-packed COPMs must be removed.
4.  Pick the candidate COPM with the highest FFV and really pack the corresponding box ac-
    cording to the COPM. The corner list is updated for later packing procedures.
```

Fig. 5 LFF algorithm for container loading problems

4. Experimental Results

Benchmarks from Bischoff and Ratcliff [8] and Loh and Nee [11] are used.

Test case (no. of box types, mean no. of boxes per type)	Average volume utilization (%) achieved by LFF	Best volume utilization (%) achieved by LFF
BR1 (3, 50.1)	87.19	93.42
BR2 (5, 27.3)	88.02	93.92
BR3 (8, 16.8)	88.37	93.84
BR4 (10, 13.3)	88.07	92.45
BR5 (12, 11.1)	88.05	92.09
BR6 (15, 8.8)	88.15	92.43
BR7 (20, 6.5)	87.62	90.90
Average	87.92	92.72

Table 1 Numerical results obtained by LFF on 700 problems of Bischoff and Ratcliff

In Bischoff and Ratcliff test cases, there are seven sets with 100 individual cases in each set. The average volume utilization of the seven sets is 87.92%. The best volume utilization in every set is over 90%. It is very satisfactory. The heterogeneity increases from BR1 to BR7. The average volume utilization for weakly heterogeneous set BR1 is slightly lower than that of strongly heterogeneous ones but the difference is very small. This shows that the performance of LFF is stable for all kinds of cases.

Test cases	Ngoi et.al. [3]	Bischoff et al. [9]	Bischoff and Ratcliff [8]	Gehring and Bordfeldt [10]	Bortfeldt and Gehring [1]	LFF
Mean Vol. Util.(%)	69.0	69.5	68.6	70.0	70.1	70.1

Table2 Numerical results obtained by six algorithms on 15 problems of Loh and Nee

Table 2 shows results obtained by six algorithms on 15 Loh and Nee problems. Using LFF, no boxes are left unstowed in 13 out of 15 Loh and Nee problems. The average volume utilization is 70.1%, which is better than four other methods.

The experiments of LFF are repeated with different criteria for measuring flexibility of boxes. We try to consider the longest side or the shortest side only. Improvement of results become noticeable when combining both. If the difference between the longest sides of two boxes is very large while that between their shortest sides is very small, flexibility will largely depend on the longest side, and vice versa. The results in Table 1 and 2 are obtained in this way.

5. Conclusion

This paper presents the LFF algorithm that achieves a satisfactory volume utilization in container loading problems. Yet there is still room for improvements. Firstly, the evaluation of object flexibility greatly affects the result. We believe that more can be done to sort the flexibility in a way leading to higher volume utilization. Another direction of improvement is the choice of packing position of boxes based on the FFV of their COPMs. There are often many COPMs having the same FFV. The first one is always selected without further assessment. If a mechanism is introduced to solve this tie, a further raise in volume utilization is possible. Future research will focus on these two issues.

Acknowledgement

This work was partially supported by Hong Kong RGC Grant CUHK4229/03E, and NSFC/RGC N_CUHK405/02

Reference

1. A. Bortfeldt, H. Gehring, A hybrid genetic algorithm for the container loading problem, European Journal of Operational Research 131(2001) 143–161
2. A. Bortfeldt, H. Gehring, D. Mack, A parallel tabu search algorithmfor solving the container loading problem, Parallel Computing 29 (2003) 641–662
3. B. K. A. Ngoi, M. L, Tay, E. S. Chua, Applying spatial representation techniques to the container packing problem, International Journal of Production Research 1(1994) 59–73
4. C. S. Chen, S. M. Lee, Q. S. Shen, An analytical model for the Container Loading Problem, European Journal of Operational Research 80(1995) 68–76
5. D. Pisinger, Heuristic for the container loading problem, European Journal of Operational Research 141 (2002) 982 – 392
6. D. Pisinger. The container loading problem, in Proceedins NOAS' 97, 1997.
7. D. Y. He, J. A. Cha, Research on solution to complex container loading problem based on genetic algorithm, Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing (2002)
8. E. E. Bischoff, B. S. W. Ratcliff, Issues in the development of approaches to container loading, Omega 23 (1995) 377 – 390
9. E. E. Bischoff, F. Janetz, M. S. W. Ratcliff, Loading pallets with non-identical items, European Journal of Operational Research 84(1995) 681–692
10. H. Gehring, A. Bortfeldt, A genetic algorithm for solving the container loading problem, International Transactions in Operational Research 4 (1997) 401-418
11. T. H. Loh, A. Y. C. Nee, A packing algorithm for hexahedral boxes, Proceedings of the Conference of Industrial Automation, Singapore(1992) 115-126
12. Y. L. Wu, W. Q. Huang, S. C. Lau, C. K. Wong and G. H. Young, An effective quasi-human heuristic for solving the rectangle packing problem, European Journal of Operational Research 141 (2002) 341 – 358

Scheduling with Fuzzy Methods

Wolfgang Anthony Eiden

Darmstadt University of Technology, Department of Computer Science,
Algorithmics Group, 64289 Darmstadt, Germany, research@wolfgang-eiden.de

Abstract. Nowadays, manufacturing industries – driven by fierce competition and rising customer requirements – are forced to produce a broader range of individual products of rising quality at the same (or preferably lower) cost. Meeting these demands implies an even more complex production process and thus also an appropriately increasing request to its scheduling. Aggravatingly, vagueness of scheduling parameters – such as times and conditions – are often inherent in the production process. In addition, the search for an optimal schedule normally leads to very difficult problems (NP-hard problems in the complexity theoretical sense), which cannot be solved efficiently.
With the intent to minimize these problems, the introduced heuristic method combines standard scheduling methods with fuzzy methods to get a nearly optimal schedule within an appropriate time considering vagueness adequately.

1 Introduction

Scheduling is a fundamental part of production planning and control. The task of scheduling is the allocation of activities over time to limited resources, where a number of conditions must be preserved. Resources represent objects, which can be allocated by activities. Using ordered sequences of activities, basic production flows can be specified. These sequences, which are mainly predetermined by technical or organizational requirements, are specified by jobs. Jobs can also specify supplementary conditions, as for example deadlines. In manufacturing, a job usually models an order.

2 Basic Approaches

Before the method presented in this paper is sketched, the fundamental basic approaches are considered which led to its development.
At first, reasons for conceptualizing the method as heuristic are given (Section 2.1). After showing that the consideration and processing of vague data, conditions and objectives are necessary (Section 2.2), it is shown how the integration of such information can by achieved in a common way (Section 2.3). Using this potential, it is also possible to integrate varying, partly vague conditions into the scheduling process (Section 2.4). Based on this possibility of integration, allocation recommendations can be derived (Section 2.5) which can be finally transferred into an allocation decision (Section 3.6).

2.1 Usage of a heuristic method

Nowadays, many manufacturing industries are confronted with large variety in jobs and activities whose processing can be very complex to coordinate or schedule. In practice, the determination of optimal schedules normally leads to NP-hard problems in the complexity theoretical sense, for which no efficient algorithms are known [3]. Nevertheless the theoretically optimal schedule has mostly only a short time of validity [5].

Considering the cost-benefit calculation, it is advisable to use a heuristic method generating an approximately optimal schedule in appropriate time.

2.2 The necessity for integration of vagueness

The considered input variables and parameters in scheduling – such as times, lengths of times, quantities and restrictions – usually possess an inherent vagueness [5]. Often, sharpened data is not available or can only be expensively acquired. In addition, dependences between relevant variables are only known approximately [2].

As a rule, there is a continuous transition between permissible and non-permissible conditions [9]. Frequently this fact is ignored [6]. Instead vague data or conditions are often sharpened artificially. However, artificial sharpening of data or conditions should usually be advised against. An artificial sharpening leads sometimes to a distorted image of the reality. In the worst case this leads even to a complete loss of reality [7]. Since it is closer to reality, the consideration of vague information is better than the consideration of artificial sharpened information [6].

As logical consequence, it is necessary to integrate the vagueness of naturally vague information into the scheduling process.

2.3 Integration of vagueness with fuzzy methods

Naturally vague information conveys in their basic form (but also in a nearly basic form) a more exact conceivability of its accuracy than in an artificially sharpened form. It usually can be assumed that the scheduling results will be more realistic when using information with a form as close to its basic form as possible [9], [6].

A computer-aided interpretation and processing is only attainable if the underlying modeling and processing are both well defined and equally suitable for sharp and for vague information. Vagueness must be processed precisely. For this reason, both the modeling language and the kind of processing must be from a strictly mathematical nature. As a premise, both high comprehensibility and transparency of decision must be ensured [5].

In this context, the fuzzy set theory is particularly suitable. With the fuzzy set theory it is possible to map and precisely process both sharp information and not exact quantifiable information (and vague information respectively) in a uniform way [5].

2.4 Integration of varying, partly vague, basic conditions and objectives

Scheduling processes are subjected to varying, partly vague conditions and objectives. It concerns production internal conditions as well as production external conditions. For instance, job- and resource-specific conditions (production internal conditions) are regarded as well as politically and strategically characterized conditions (production external conditions).

A fundamental approach of the presented method is to provide a possibility to integrate the varying, partly vague, basic conditions directly into the scheduling process and therefore to minimize manual intervention.

For this purpose, it is necessary to interlink the context relevant conditions adequately and to use them as a decision basis whenever decisions must be made in the scheduling process.

The fuzzy theory offers the possibility to put that into practice using fuzzy approximate reasoning methods – as for instance the fuzzy decision support system of *Rommelfanger* and *Eickemeier* [10]. With this method, it is possible to map and precisely process sharp information and not exact quantifiable information (and vague information respectively) – such as data, conditions and assessments – in a uniform way [5].

Human decision-making processes can also be integrated into the scheduling process. The feature of human decision-making processes is to get a good solution even if the decision circumstances are complex or poorly structured [11]. That applies also if the underlying information is incomplete, vague or even contradictory [12].

On account of these possibilities, varying, partly vague, basic conditions and objectives can be uniformly integrated into the scheduling process independently of their degree of vagueness. In this way, it is guaranteed that their substantial influence also appears in the scheduling process.

2.5 Usage of resource-specific and resource-comprehensive recommendations for allocations

The allocation of a job is performed by allocation of all its activities to resources – and thereby, the conditions must be considered.

A fundamental idea of the presented method is to keep up a greatest possible degree of flexibility as long as possible to be able to generate an approximately optimal schedule.

The approach is to first determine the resource-specific optimal sequence of all activities to be allocated. In doing so, a detailed perception of the preferred allocation sequence of every resource is gained. Equipped with this information, resource-comprehensive recommendations for allocations can be determined. After all, an explicit allocation decision can be derived from these recommendations.

While determining the recommendations (both the resource-specific recommendations and the resource-comprehensive recommendations), the conditions and objectives described in Section 2.4 must be considered. Since these conditions and objectives can be handled by fuzzy methods in an adequate manner, it is advisable to also determine the recommendations with fuzzy methods.

3 The method for scheduling under vagueness

Based on the fundamental approaches previously discussed, the initial idea for the following new method was developed. The purpose of this method is to get a nearly optimal schedule within an appropriate time considering the vagueness in the scheduling process adequately. The method itself is designed iteratively using a rolling allocation decision mechanism (see figure 1).
Since a specific activity is in the following always clearly assigned to a job, a job is an outer wrapper of its activities specifying activity-comprehensive conditions.

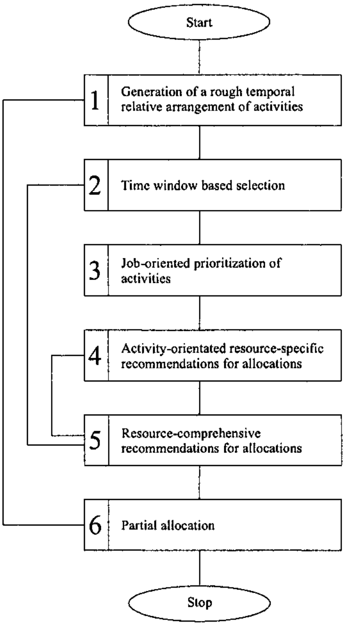

Start
1
Generation of a rough temporal relative arrangement of activities
2
Time window based selection
3
Job-oriented prioritization of activities
4
Activity-orientated resource-specific recommendations for allocations
5
Resource-comprehensive recommendations for allocations
6
Partial allocation
Stop

Fig. 1. The scheduling method (Overview)

3.1 Generation of a rough temporal relative arrangement of activities

Starting from the jobs and their activities to be allocated, a rough temporal relative arrangement of activities is generated. The generation of this arrangement is based on a fuzzy version of a retrograde scheduling method[1]. In literature, a retrograde scheduling method is sometimes also called backward scheduling method.

With this method, the course of scheduling occurs contrary to the technological course; starting from the deadline of a job, the latest possible allocation of its activities is realized [8]. Since dates, times, and duration of times are often vague, the retrograde scheduling method is extended to be capable of handling fuzzy representations of these temporal parameters.

The generated rough temporal relative arrangement is used as optimized input for the following time window based selection.

3.2 Time window based selection

Starting from the generated rough temporal relative arrangement of the activities to be scheduled, the activities, which should be considered in a forward-shifted horizon of fixed size, are taken into account by a time window based selection. In this way, a quantitative restriction of the activities observed by the succeeding steps of the method is performed.

This way of proceeding is ostensibly comparable with the load-oriented order release scheduling method, but the concept is different. The load-oriented order release scheduling method is based on the proposition, that a reduction of the average machining time is only possible with a lowering of the average quantity of the prior activity queue; the presented method uses the time window based selection only to reduce the complexity of the succeeding steps.

Usually, no complete jobs are represented by the time window based selection of activities. However, manufacturing is primarily job-oriented. For this reason, the list of the selected activities is extended with all unscheduled activities assigned either to the same jobs as the activities picked up by the time window based selection or to only partial allocated jobs of a prior run. In this way, a list of activities is generated which contains all unscheduled activities of jobs which are referenced either by the time window based selection or by a prior run.

3.3 Job-oriented prioritization of activities

Considering job-specific and comprehensive conditions, all activities of the activity list generated in the previous step are prioritized. Amongst other

[1] For detailed information about the retrograde termination method, please see [12] or [1].

things, the activities belonging to important jobs are emphasized in contrast to activities of less important jobs. The prioritization is made by a fuzzy rating method basing on the fuzzy decision support system introduced by *Rommelfanger* and *Eickemeier* [10]. Figure 2 shows an example for a job-specific rating.

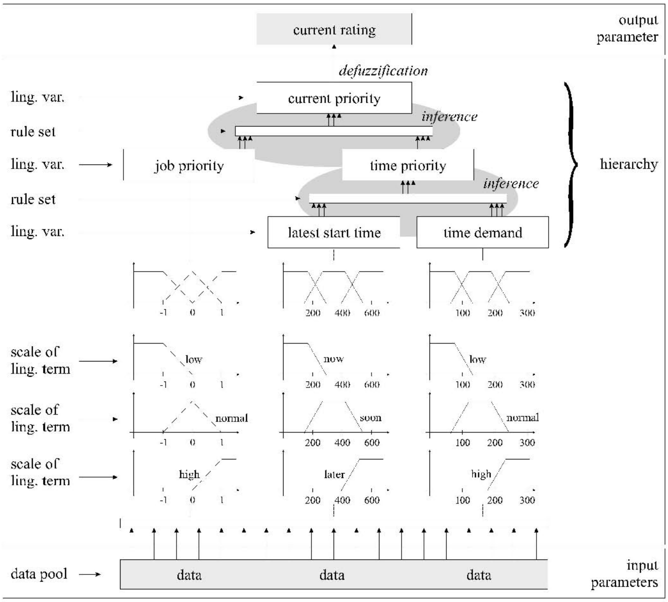


Fig. 2. Rating method based on a fuzzy decision support system

The priorities assigned to the activities induce a partial order. This partial ordered activity list is used as an initial list for the following activity-orientated resource-specific allocation recommendations.

3.4 Activity-orientated resource-specific recommendations for allocations

Considering the given partial order as much as possible, the activities given by the partial ordered activity list are prioritized resource-specific. In order to provide an evaluation process, the method for the resource-specific allocation recommendation also pays attention to several recently scheduled activities. For this purpose, overlapping resource-specific time windows are used within the rolling planning process. The activities given by the partial ordered activity list and any recently scheduled activities to consider are aggregated to a new activity list which is then prioritized resource-specific by a fuzzy rating method.

If an activity cannot be allocated by a specific resource, it is not a member of the corresponding resource-specific activity list to be prioritized; if an activity can be allocated by several resources it is a member of all corresponding activity lists but usually with different priorities depending on the specific resource. Thereby, the resource-specific situation and conditions are considered as well as comprehensive conditions, hard restrictions and objectives.
In doing so, a detailed perception of the preferred allocation sequence of every resource is gained. In this way, resource-specific allocation recommendations can be derived. Equipped with this information, an optimized resource-comprehensive recommendation for allocations can be determined.

3.5 Resource-comprehensive recommendations for allocations

From the resource-comprehensive viewpoint the resource-specific allocation recommendations generated in the last step are not necessarily redundancy-free. The corresponding ordered lists may contain activities which are elements of above one of these lists. Since every activity must be allocated at most one time, a method for redundancy removal is accomplished. Thereby, the consideration of the outer conditions (the strategic conditions in particular) is guaranteed by a fuzzy rating method.
In order to achieve a preferably balanced utilization simultaneously, a rolling allocation rating process is employed, which use resource-specific time windows. This rating process will be repeated until no changes or no more significant changes will occur. In doing so, resource-comprehensive recommendations for an allocation are derived for every resource. These recommendations are limited by the resource-specific sliding time windows.

3.6 Allocation and Continuation

The resource-comprehensive allocation recommendations determined in the last step are transformed to allocation determinations for the considered activities. Subsequently, the allocations are performed.
Afterwards, all activities that are not allocated are brought into a further scheduling process again. Thereby, all allocated activities are no longer considered in the further scheduling process; if all activities of a job are allocated, this applies also to the corresponding job. In this way, the final schedule containing all activities and jobs is constructed step-by-step.

4 Summary

In this paper, a new method was presented, which integrates the vagueness of naturally vague information in specified shape into the scheduling process considering varying, partly vague, basic conditions and conditions.
It was shown how it is possible to integrate that important, but usually hardly

used, source of information using the fuzzy theory and the techniques developed from it. It was also shown the possibility of the integration of human decision and human assessment processes into the scheduling process.
By the conscious use of vague information and the avoidance of over specification, even a complexity reduction can be achieved [7]. For not going beyond the scope of this paper, the approach of self-organizing activities [4] was not considered.

References

1. Adam, D. (1992): Retrograde Terminierung: Ein Verfahren zur Fertigungssteuerung bei diskontinuierlichem Materialfluss oder vernetzter Fertigung. In: Adam D. (ed): Fertigungssteuerung – Grundlagen und Systeme. Volume **38/39**. Wiesbaden, pp 245–262.
2. Borgelt, C.; Kruse, R. (2001): Unsicherheit und Vagheit: Begriffe, Methoden, Forschungsthemen. In: KI, Künstliche Intelligenz **3/01**. Arendtap, Bremen, pp 18–24.
3. Brucker, P. (2001): Scheduling Algorithms. 3rd edn. Springer, Berlin Heidelberg New York.
4. Eiden, W.A. (2003): Prioritätengesteuertes Scheduling auf Basis eines multikriteriellen Fuzzy-Bewertungsverfahrens. Technical Report. Darmstadt University of Technology, Darmstadt.
5. Eiden, W.A. (2003): Flexibles Scheduling auf Basis von Fuzzy-Technologien. In: Geldermann, J.; Rommelfanger, H. (eds): Einsatz von Fuzzy-Sets, Neuronalen Netzen und Künstlicher Intelligenz in industrieller Produktion und Umweltforschung. Fortschritt-Berichte **10/725**. VDI, Düsseldorf, pp 70–84.
6. Eiden, W.A. (2004): Job-Shop Scheduling under Vagueness. Euro XX - 20th European Conference on Operational Research, Paper-ID: 1333. July 4-7, 2004, Rhodes, Greece.
7. Heitmann, C. (2002): Beurteilung der Bestandsfestigkeit von Unternehmen mit Neuro-Fuzzy. PhD Thesis. Peter Lang, Frankfurt am Main.
8. Nebl, T. (2002) Production Management: Oldenbourg, München Wien.
9. Rausch, P. (1999): HIPROFIT – Ein Konzept zur Unterstützung der hierarchischen Produktionsplanung mittels Fuzzy-Clusteranalysen und unscharfer LP-Tools. PhD Thesis. Peter Lang, Frankfurt am Main
10. Rommelfanger, H.; Eickemeier, S. (2002): Entscheidungstheorie – Klassische Konzepte und Fuzzy-Erweiterungen. Springer, Berlin Heidelberg.
11. Schwab, J. (1999): Logistisches Störungsmanagement. 44th International Scientific Colloquium. Technical University of Ilmenau, Ilmenau.
12. Sibbel, R. (1998): Fuzzy-Logik in der Fertigungssteuerung am Beispiel der retrograden Terminierung. LIT Verlag Dr. Wilhelm Hopf, Münster.

Generalized DEA-Range Adjusted Measurement

Andreas Kleine and Dennis Sebastian

University of Hohenheim (510 B), Institute of Business Administration,
Operations Research, D-70593 Stuttgart, E-mail: ankleine@uni-hohenheim.de,
dennis.sebastian@rub.de

Abstract. Data Envelopment Analysis (DEA) is designed to measure the efficiency of decision making units (DMUs). A scalarizing function reduces all numerical information on inputs and outputs of a DMU to a single efficiency score. A range adjusted measure normalizes input and output values through equalization factors, already known from Multicriteria Decision Making. Cooper et al (1999) introduce a range adjusted DEA model for a non-radial measure with variable returns to scale. We extend this approach by the help of a general DEA model framework. Calculation of suitable range adjusted measures is incorporated in a web-based DEA-tool. This tool allows processing of individual sets of data with a broad choice of model characteristics.

1 Introduction

Data Envelopment Analysis (DEA) is an approach to compare relative efficiency of decision making units (DMUs) such as schools, hospitals, libraries, banks etc. As the investigated DMUs use multiple inputs to produce multiple outputs direct comparisons are generally difficult. DEA makes use of linear programs to determine their relative performance which is defined as the distance between the efficient frontier and the DMU under consideration. Since the famous publication by Charnes, Cooper and Rhodes (CCR) [3] a large number of DEA models with various extensions and applications has been published [7,14].

Assumptions made for those models differ fundamentally with respect to technologies and aggregation rules of scalarizing functions. In contrast to the classical CCR model a range adjusted measure (RAM) utilizes technology-dependent ranges of feasible inputs and outputs [1,13]. Cooper, Park and Pastor [5] introduce a particular RAM model for a technology with variable returns to scale. Steinmann and Zweifel [11] discuss RAM properties for technologies with constant returns to scale. We extend the range adjusted measure by the help of a DEA model framework and illustrate the calculation of equalization factors in general.

The paper is organized as follows: section 2 introduces a DEA model framework. On this basis we analyze range adjusted models in section 3. Section 4 presents a brief description of the technical environment used for calculations. A numerical example serves to illustrate the concept for various technologies and scalarizing functions.

2 DEA-Model Framework

In the following we assume a set of J DMUs. Each DMU$_j$ is characterized by M inputs x_{mj} $(m = 1, \ldots, M)$ and N outputs y_{nj}. The set of all feasible activities (productions) consists of linear combinations of all DMUs, incorporating activity levels $(\lambda_1, \ldots, \lambda_J)' \in \Lambda$:

$$\Lambda = \left\{ \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_J \end{pmatrix} \in \mathbb{R}^J_+ \;\middle|\; \begin{array}{c} \underline{\lambda} \leqq \sum_j \lambda_j \leqq \overline{\lambda} \\ \cdots \end{array} \right\}.$$

Usually the sum of all activity levels has a lower bound $\underline{\lambda}$ and an upper bound $\overline{\lambda}$ $(0 \leqq \underline{\lambda} \leqq \overline{\lambda})$. As the dots indicate, additional conditions are conceivable: the multipliers λ_j are defined as integer variables, a restricted number of DMUs serving as a reference, etc. [7].

All efficient input-output combinations constitute the efficient frontier. A combination is called efficient, if no other one exists which is better in at least one input or output and not worse in the others. In the sense of Multicriteria Decision Making we minimize inputs and maximize outputs simultaneously.

The efficiency of a DMU$_0$ is calculated by comparing its inputs x_{m0} and outputs y_{n0} to a point on the efficient frontier. Deviations are captured by nonnegative variables d^-_m and d^+_n respectively. Those are well known from goal programming [12,10]. Here a scalarizing function ψ aggregates all weighted input-output deviations [8,9]:

$$\begin{array}{lll} (DEAM) & \max \; \psi(w^-_1 d^-_1, \ldots, w^-_M d^-_M, w^+_1 d^+_1, \ldots, w^+_N d^+_N) & \\ & s.t. \; 1) \sum_j x_{mj}\, \lambda_j + d^-_m = x_{m0} & (m = 1, \ldots, M) \\ & \quad\;\; 2) \sum_j y_{nj}\, \lambda_j - d^+_n = y_{n0} & (n = 1, \ldots, N) \\ & \quad\;\; 3) \; \lambda \in \Lambda & \\ & \quad\;\; 4) \; d^-_m, \, d^+_n \geqq 0 \quad (m = 1, \ldots, M; n = 1, \ldots, N) & \end{array}$$

In $(DEAM)$ we maximize the distance from any given DMU$_0$ to an efficient input-output-combination. If DMU$_0$ lies on the efficient frontier all deviations are zero and $\psi = 0$. Inefficient DMUs in contrast are characterized by a strictly positive efficiency score. The individual optimal value of scalarizing function ψ serves to rank inefficient DMUs. Many DEA models transform the objective function, so that efficient DMUs are indicated by $\psi = 1$.

The general framework $(DEAM)$ allows us to analyze various models developed in DEA. The models differ in

- technology (vrs, crs, nirs, ndrs, fdh, ...),

- weighting (DMU-specific, range adjusted, ...),
- orientation (input-, output- or non-oriented),
- aggregation rule (non-radial (additive), radial (maximin norm), ...).

For example, the input-oriented CCR formulation of Charnes, Cooper and Rhodes [3] consists of

- a technology with constant returns to scale (crs: $\underline{\lambda} = 0$, $\overline{\lambda} = \infty$),
- DMU-specific weights and input orientation ($w_m^- = 1/x_{m0}$, $w_n^+ = 0$), and
- a maximin norm ($\psi = \min_{m,n}\{w_m^- \, d_m^-, w_n^+ \, d_n^+\}$).

A classification of DEA models is presented in Kleine [9]. In the following section we analyze models with so called range adjusted weights.

3 Range Adjusted Measure

3.1 Range Adjusted Model

Cooper, Park and Pastor [5] introduce a non-oriented model. In contrast to additive models [4] with unit weights ($w_m^- = w_n^+ = 1$) they recommend an additive model with range adjusted weights R_m^- for each input and similarly R_n^+ for outputs. Assuming a technology with variable returns to scale (vrs), the ranges are:

$$\begin{aligned} R_m^- &= \max_j\{x_{mj}\} - \min_j\{x_{mj}\} \quad (m = 1, \ldots, M) \\ R_n^+ &= \max_j\{y_{nj}\} - \min_j\{y_{nj}\} \quad (n = 1, \ldots, N) \end{aligned}$$

Thus, a special formulation of (DEAM) with an additive scalarizing function ψ and a technology with variable returns to scale follows.

$$\begin{aligned} (RAM) \quad & \max \frac{1}{M+N}\left(\textstyle\sum_m \frac{1}{R_m^-}\, d_m^- + \sum_n \frac{1}{R_n^+}\, d_n^+\right) \\ s.t. \; & 1) \textstyle\sum_j x_{mj}\, \lambda_j + d_m^- = x_{m0} \qquad (m = 1, \ldots, M) \\ & 2) \textstyle\sum_j y_{nj}\, \lambda_j - d_n^+ = y_{n0} \qquad (n = 1, \ldots, N) \\ & 3) \textstyle\sum_j \lambda_j = 1 \\ & 4) \lambda_j,\, d_m^-,\, d_n^+ \geqq 0 \quad (j = 1, \ldots, J; m = 1, \ldots, M; n = 1, \ldots, N) \end{aligned}$$

Figure 1 illustrates a two-dimensional example using a set of 7 DMUs (A to G) with the efficient frontier indicated by a solid line. Input range R^- results from efficient DMU$_A$ und inefficient DMU$_E$, output range R^+ from efficient DMU$_A$ and efficient DMU$_D$.

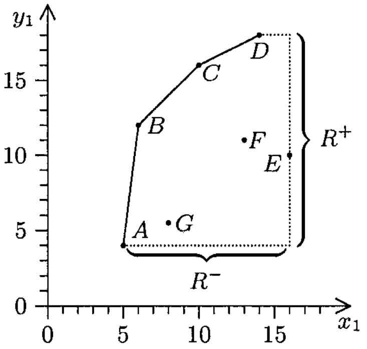


Fig. 1. Range adjusted weights for technology with vrs

3.2 General Range Adjusted Measurement

A range adjusted measure ψ used in (RAM) satisfies a lot of properties [5,6], e.g. ψ is invariant to alternative units in which inputs or outputs are measured. Another key property of ψ is the identification of fully inefficient DMUs:

$$\psi = \begin{cases} 0 & \text{DMU}_0 \text{ is efficient} \\ 1 & \text{DMU}_0 \text{ is fully inefficient} \end{cases}$$

A fully inefficient DMU represents the worst case in the underlying criterion space. Steinmann and Zweifel [11] demonstrate that measure ψ violates this property for a technology with constant returns to scale (crs). In order to guarantee an identification of fully inefficient DMUs the calculation of ranges has to depend on the technology assumed. If we apply equalization factors in general [12, p 201], then the following ranges result:

$$R_m^- = \max_j\{x_{mj}\} - \min\left\{ x_m \middle| \begin{pmatrix} x \\ y \end{pmatrix} \in T \right\} \quad (m = 1, \ldots, M)$$

$$R_n^+ = \max\left\{ y_n \middle| \begin{pmatrix} x \\ y \end{pmatrix} \in T \right\} - \min_j\{y_{nj}\} \quad (n = 1, \ldots, N)$$

$$\text{with } T = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \mathrm{I\!R}^{M+N} \middle| \begin{array}{ll} \sum_j x_{mj}\,\lambda_j \leqq \max_j\{x_{mj}\} & (m = 1, \ldots, M) \\ \sum_j y_{nj}\,\lambda_j \geqq \min_j\{y_{nj}\} & (n = 1, \ldots, N) \\ \lambda \in \Lambda & \end{array} \right\}$$

According to this definition each technology has an individual set of ranges. A General Range Adjusted Model (GRAM) differs from (RAM) in factors R_m^- and R_n^+ and the set of feasible activities Λ. Figure 2 shows the range differences for technologies with constant and non-increasing returns to scale. Using these ranges yields $\psi = 1$ for each fully inefficient DMU. A DMU is fully inefficient, if all deviations correspond with ranges defined. This case will only arise rarely.

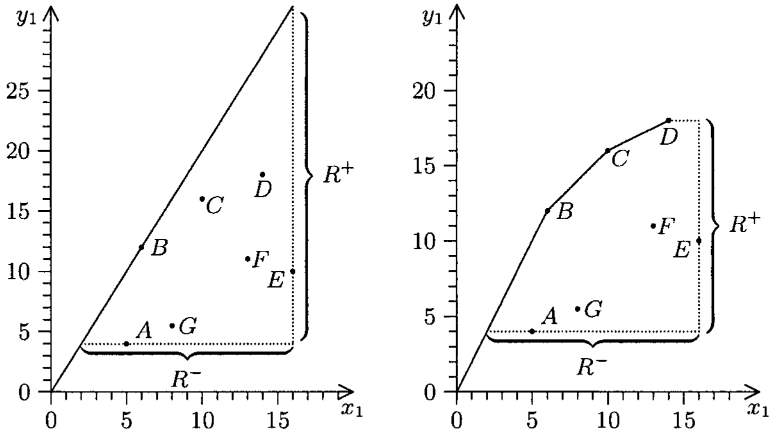


Fig. 2. Range adjusted weights for technology with crs and nirs

If ranges change, we must be conscious of the fact that the ranking of inefficient DMUs can alter. However, for a given technology a change of weights has no influence on the set of efficient DMUs.

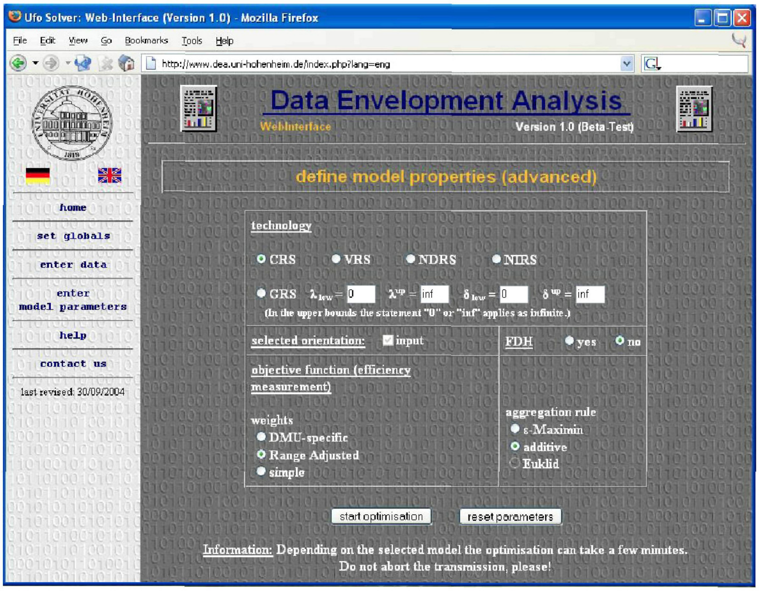
4 Application of RAM

4.1 DEA-Tool with RAM

To facilitate analyses of numerous different data sets, especially of higher-dimensional problems, we developed a web-based integrated application. Besides direct input via input masks data may also be imported from structured flat files. Without deeper knowledge of the underlying model structure, practitioners are able to choose from a broad selection of standard models (CCR, BCC, FDH, RAM, ...). Advanced users may alternatively make individual settings for each model property, allowing even more combinations (cf. Fig 3).

All required command and model files are dynamically generated by the tool and prepared for solving. The following automated sequential optimization includes calculation of feasible ranges using separate linear programs for each dimension. The resulting technology-dependent ranges go directly into (GRAM) as parameters. The tool solves the linear program (GRAM) for each DMU separately. The maximum processable problem size depends on factors such as technology and number of DMUs. During this whole process no further interaction with the user is necessary.

In a final step, solution data is presented in different ways. A compact solution report is provided for download along with an Excel macro for comfortable formatting to enable further processing by the user. This allows comparisons between results obtained from different models, for example with and without range adjusted objective functions.


Ufo Solver: Web-Interface (Version 1.0) - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.dea.uni-hohenheim.de/index.php?lang=eng
home
set globals
enter data
enter model parameters
help
contact us
start optimisation
reset parameters

last revised: 30/09/2004

Data Envelopment Analysis

WebInterface Version 1.0 (Beta-Test)

define model properties (advanced)

technology

● CRS ○ VRS ○ NDRS ○ NIRS

○ GRS λ_{low} = 0 λ^{up} = inf δ_{low} = 0 δ^{up} = inf

(In the upper bounds the statement "0" or "inf" applies as infinite.)

selected orientation: ☑ input

FDH ○ yes ● no

objective function (efficiency measurement)

weights
○ DMU-specific
● Range Adjusted
○ simple

aggregation rule
○ ε-Maximin
● additive
○ Euklid

Information: Depending on the selected model the optimisation can take a few minutes. Do not abort the transmission, please!

Fig. 3. Property setting dialogue

4.2 Numerical Example

In order to isolate the effect of alternative range adjustments, we use an additive non-oriented crs model and compare its results using different ranges as weights. These are obtained from range calculations assuming variable returns to scale in one case and constant returns in the other. Using data taken from an example in Zhu (cf. Tab. 1) we add DMU *AddCom* with particularly few employees. This way remarkable effects can be observed:

- some DMUs change their relative ranking
- reference set and references change

As shown in the above section, technologies other than vrs generally lead to broader feasible ranges, although they differ in the extent of change. Going from vrs to crs in our example all input ranges hardly change, whereas both output ranges grow extraordinarily large. One of them is approximately eleven times as big. This fact does not only alter the partial slopes of the objective function, but influences the relevance of deviations in each dimension as well. That may lead to distinct changes in the overall efficiency measure in both directions. Assumably, this particularly applies to inefficient DMUs having an inhomogenous structure. Consequently, DMUs *Mitsubishi*, *Ford Motor*, *Toyota Motor*, *Royal Dutch/Shell Group* and *Hitachi* are now ranked differently among all inefficient DMUs.

Apart from their influence on inefficient DMUs, alternative weights can also have an effect on their efficient counterparts. A subset of efficient DMUs

Table 1. Data of 16 companies [14, p 14] with input/output-ranges R_m^-,R_n^+

Company	Assets	Equity	Employees	Revenue	Profit
Mitsubishi (A)	91920.6	10950.0	36000	184365.2	346.2
Mitsui (B)	68770.9	5553.9	80000	181518.7	314.8
Itochu (C)	65708.9	4271.1	7182	169164.6	121.2
General Motors (D)	217123.4	23345.5	709000	168828.6	6880.7
Sumitomo (E)	50268.9	6681.0	6193	167530.7	210.5
Marubeni (F)	71439.3	5239.1	6702	161057.4	156.6
Ford Motor (G)	243283.0	24547.0	346990	137137.0	4139.0
Toyota Motor (H)	106004.2	49691.6	146855	111052.0	2662.4
Exxon (I)	91296.0	40436.0	82000	110009.0	6470.0
Royal Dutch/Shell Group (J)	118011.6	58986.4	104000	109833.7	6904.6
Wal-Mart (K)	37871.0	14762.0	675000	93627.0	2740.0
Hitachi (L)	91620.9	29907.2	331852	84167.1	1468.8
Nippon Life Insurance (M)	364762.5	2241.9	89690	83206.7	2426.6
Nippon Telegraph & Telephone (N)	127077.3	42240.1	231400	81937.2	2209.1
AT&T (O)	88884.0	17274.0	299300	79609.0	139.0
AddCom (X)	120034.0	8590.0	3600	173764.8	362.4
	R_1^-	R_2^-	R_3^-	R_1^+	R_2^+
vrs-ranges	326891.5	56744.5	705400.0	104756.2	6783.4
crs-ranges	340796.3	56972.6	707350.7	1136031.6	12536.9

constitutes the reference set which some may enter or leave. In our exemplary case the new enlarged reference set additionally contains DMU *AddCom.* This takes into account relative advantages in one or more dimensions. Because of their unique input/output ratio, such DMUs represent complementary productions. In case of crs-ranges input levels get more important. Consequently both outputs do not exceed their lower bounds any more, neglecting occasionally great slacks in the inputs and vice versa for vrs-ranges. DMU *Ford Motor* illustrates this feature most prominently. This behaviour can be attributed to smaller objective weights for output deviations due to a much

Table 2. Scores ψ for inefficient DMUs

	crs-ranges		vrs-ranges	
DMU	rank	references	rank	references
Mitsubishi (A)	2	E,I	3	C,D,E
Marubeni (F)	1	C,E	1	C,E,X
Ford Motor (G)	4	C,D,I	7	C,D,E,I
Toyota Motor (H)	5	E,I	4	E,I
Royal Dutch/Shell Group (J)	3	I	2	E,I
Hitachi (L)	7	E,I	5	E,I
Nippon Telegraph & Telephone (N)	8	E,I	8	E,I
AT&T (O)	6	E,I	6	E

broader range. Thus, an optimal theoretical reference seeks to economize on inputs.

Summary

The analyses in section 4 clarify that ranges do not depend on subjective estimates of experts, but rather on the technology assumed. The computation of ranges is directly supported by the tool. However, a range represents – for a given technology – the difference of feasible quantities in best case and worst case. The hypothetical ranges are usually not achieved. Hence, it can be helpful to define additional limits for ranges. Moreover, a ranking of DMUs can depend on inefficient DMUs [2]. If an inefficient DMU with an extreme input is added, as a consequence ranges change and – as shown above – the resulting ranks and reference sets can differ.

References

1. Aida, K., Cooper W.W., Pastor J., Sueyoshi T. (1998) Evaluating Water Supply Services in Japan with RAM (A Range Adjusted Measure of Inefficiency). Omega, **26**, 207–232.
2. Bouyssou D. (1999) Using DEA as a tool for MCDM: some remarks. Journal of the Operational Research Society, **50**, 974-978
3. Charnes A., Cooper W.W., Rhodes E. (1978) Measuring the Efficiency of Decision Making Units. European Journal of Operational Research **2**, 429-444
4. Charnes, A., Cooper W.W., Golany B., Seiford L.M., Stutz, J. (1985) Foundations of Data Envelopment Analysis for Pareto-Koopmans Efficient Empirical Production Functions. Journal of Econometrics, **30**, 91–107.
5. Cooper W.W., Park K., Pastor J. (1999) RAM: A Range Adjusted Measure of Inefficiency for Use with Additive Models, and Relations to Other Models and Measures in DEA. Journal of Productivity Analysis, **11**, 5–42
6. Cooper W.W., Park K., Pastor J. (2001) The Range Adjusted Measure (RAM) in DEA: A Response to the Comment by Steinmann and Zweifel. Journal of Productivity Analysis, **15**, 145–152
7. Cooper W.W., Seiford L.M., Tone K. (2000) Data Envelopment Analysis. Kluwer, Boston.
8. Kleine, A. (2002) DEA-Effizienz. DUV, Wiesbaden.
9. Kleine, A. (2004) A general model framework for DEA. Omega **32**, 17–23.
10. Miettinen, K. (1999) Nonlinear Multiobjective Optimization. Kluwer, Boston.
11. Steinmann L., Zweifel P. (2001) The Range Adjusted Measure (RAM) in DEA: A Comment. Journal of Productivity Analysis, **15**, 139–144
12. Steuer, R.E. (1986) Multiple Criteria Optimization: Theory, Computation and Application. Wiley, New York.
13. Sueyoshi T., Hwang S.-N. (2004) Parallel Network Computing Approach for DEA-RAM Measurement. Asia Pacific Journal of Operational Research, **21**, 69–95.
14. Zhu, J. (2003) Quantitative Models for Performance Evaluation and Benchmarking. Kluwer, Boston.

A fuzzy DEA approach

-Ranking production units in Taiwanese semiconductor industry-

Elmar Reucher and Wilhelm Rödder

Lehrstuhl für Betriebswirtschaftslehre,
insb. Operations Research
FernUniversität in Hagen

elmar.reucher@fernuni-hagen.de
wilhelm.roedder@fernuni-hagen.de

Abstract. The theoretical concept of data envelopment analysis (DEA) was published by Charnes, Cooper and Rhodes in 1978. It is a powerful tool for assessing the efficiency of decision making units (DMUs) by comparing their respective input-factors and output quantities. More precisely, the efficiency of a DMU is measured as the relation between the sum of weighted outputs and of weighted inputs, where the weights for a DMU are the solution of a linear optimization problem. But this approach suffers from a lack of objectivity due to the fact that every DMU wants an advantageous evaluation for itself, called self-appraisal in the respective literature. The purpose of this paper is to present an extension of the evaluation process in such a way that every DMU's performance will be calculated in an objective and fair manner. For this a fuzzy oriented approach will be formulated which yields a nonlinear optimization problem. The resulting input/output weights are a compromise of all DMU's individual interests. The so far developed theoretical concept will be applied to a company of Taiwanese semiconductor industry. Five DMUs' efficiencies will be calculated over three years. Professional software solves the nonlinear program and the results permit meaningful economical interpretations.

1 Introduction

Data envelopment analysis (DEA) is a mathematical technique used to calculate the efficiency of decision making units (DMUs), which can be organizations such as business firms, hospitals, stores, restaurants, etc. The results from the analysis yield information about the input-output relation for each DMU. The approach developed by Charnes et. al. [2] generalizes the single input output technical efficiency measure by Farrell [6] to a multiple input output technique. DEA is a powerful tool which is used for different fields of application [1], [5], [8]. In this paper we present a modification of the classical DEA technique which yields a multiple objective programming approach and make it a fair efficiency assessment. Therefore we pick up an idea by Chiang, Tzeng [3] and illustrate both, the philosophy of the classical DEA concept and its modification to a fuzzy DEA approach by a a real case study of Taiwanese semiconductor industry. Either model will be solved by a professional

software tool and we give interesting interpretations for the different results between both approaches. The paper is structured as follows.
First, in section 2, we sketch the mathematical model based on Charnes et. al. In section 3 we pick up the idea of a fuzzy oriented model which allows to evaluate DMUs in a more objective way than the classical DEA approach does. Section 4 applies the theoretical ideas to a company of Taiwanese semiconductor industry and compares the results between both approaches. A conclusion in section 5 completes this paper.

2 The classical DEA-method "CCR"

This section deals with the model which was initially proposed by Charnes, Cooper and Rhodes (CCR) [2]. We give a short survey of the mathematical background and avoid details. The reader finds a good discussion about the theory in [2], [4]. Efficiency or productivity analysis are managerial control tools for assessing the degree to which inputs are utilized in the process of obtaining outputs. If the production function is known the efficient input/output combinations can be determined directly. But it is very difficult to formulate an explicit functional relationship between inputs and outputs, in general. If the production function is not known, efficiency has to be estimated based on observed input and output data. Therefore the DEA tries to overcome the difficulties in such a way that CCR refers to the unknown production function as an envelope built up relative to observational data from all of the DMUs j, $j = 1, ..., n$. Let $\mathbf{x}_j = (x_{1j}, \cdots, x_{mj})$ be observed input factors and $\mathbf{y}_j = (y_{1j}, \cdots, y_{sj})$ be observed output factors of all DMUs $j = 1, ..., n$. Than the efficiency eff_{j_0} of a DMU j_0 is measured by the output to input ratio

$$\mathit{eff}_{j_0} = \frac{\sum_{r=1}^{s} u_r y_{rj_0}}{\sum_{i=1}^{m} v_i x_{ij_0}}, \tag{1}$$

where u_r and v_i represent virtual weights of the outputs and inputs, respectively. Thus, the problem of determing the efficiency of a DMU $j_0 \in \{1, ..., n\}$ becomes the problem of calculating the weights u_r, v_i in such a way that they solve the following fractional program

$$\begin{aligned} \max \mathit{eff}_{j_0} &= \frac{\sum_{r=1}^{s} u_r y_{rj_0}}{\sum_{i=1}^{m} v_i x_{ij_0}}, \ j_0 \in \{1, ..., n\} \\ s.t. \quad & \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \leq 1, j = 1, ..., n \\ & u_r, v_i \geq 0, i = 1, ..., m, r = 1, ..., s. \end{aligned} \tag{2}$$

x_{ij} is the observed amount of the i-th input for the j-th DMU and y_{rj} is the observed amount of the r-th output for the j-th DMU. (2) for each DMU j_0 determines the maximal ratio of the sum of its weighted outputs divided by the sum of its weighted inputs. Charnes et al. postulate properties for the production possibility set T containing all possible non negative input and output combinations. Using the theory of Shepard's distance function [11] the authors deduce the linear problem (3) to be equivalent to (2).

$$\begin{aligned} &\max \sum_{r=1}^{s} u_r y_{rj_0},\ j_0 \in \{1,...,n\} \\ s.t.\quad &\sum_{i=1}^{m} v_i x_{ij_0} = 1, \\ &\sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \le 0, \\ &j = 1,...,n,\ u_r, v_i \ge 0\,\forall r, i. \end{aligned} \tag{3}$$

For more details cf. [2] (p. 1082). The objective here is to find the sum of weighted outputs for DMU j_0 while keeping the sum of weighted inputs equal to 1 and forcing the difference between the sum of weighted outputs and the sum of weighted inputs for *any* DMU to be less or equal 0 [2], [7]. The last restriction guarantees that output can never exceed input, and that is valid for any DMU. Given model (3), the efficiency assessment of n DMUs needs n optimizations, one for each DMU j_0. Let $(\mathbf{u}^*_{j_0}, \mathbf{v}^*_{j_0})$ be the solution of (3) for any $j_0 = 1, ..., n$. Than each DMU j_0 appraises itself which makes the CCR-approach suffer from a strong subjectivity. More about this in section 4. In the next section we present an objective oriented method, called "Maximal Fairness" (MF) which is based on a fuzzy optimization model, cf. [3].

3 The model of Maximal Fairness

In the CCR-Model (3) every DMUj_0 measures the efficiency by its own best virtual weights $(\mathbf{u}^*_{j_0}, \mathbf{v}^*_{j_0})$. The idea of "Maximal Fairness" is to evaluate all DMUs in such a way that they maximize their self-appraisal simultaneously. This yields the multiple objective program

$$\begin{aligned} &\max z_1 = \frac{\sum_{r=1}^{s} u_r y_{r1}}{\sum_{i=1}^{m} v_i x_{i1}}, \cdots, \max z_n = \frac{\sum_{r=1}^{s} u_r y_{rn}}{\sum_{i=1}^{m} v_i x_{in}} \\ s.t.\quad &\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \le 1,\ j = 1, ..., n \\ &u_r, v_i \ge 0,\ \forall r, i. \end{aligned} \tag{4}$$

Suppose the top-management of a group of DMUs is interested in a compromise solution in such a way that the individual interests for all DMUs will be satisfied best possible. This consideration results in a fuzzy approach. For each DMU a membership function must be defined, which expresses its degree of satisfaction with respect to its (own) objective value. Therefore the membership function is

$$\mu_j(z_j) = \begin{cases} 0 & : \ z_j \leq z_j^L \\ \frac{z_j - z_j^L}{z_j^U - z_j^L} & : \ z_j^L < z_j < z_j^U \\ 1 & : \ z_j \geq z_j^U, \end{cases} \tag{5}$$

cf. [3]. Here z_j^L ans z_j^U denote the lower and upper bounds for satisfication of DMU j. $\mu_j(z_j) = 1$ means highest satisfication and $\mu_j(z_j) = 0$ lowest. It is well known the the conjunction of fuzzy sets is defined by the minimum-operator "min" [10] so that the top-managements' degree of satisfaction can be expressed by the membership function

$$\min_j^n \mu_j(z_j). \tag{6}$$

In the sense of rational behaviour the management is interested in the highest satisfication which yields the fuzzy objective optimization problem

$$\begin{aligned} & \max\nolimits_{\mathbf{u},\mathbf{v}} \ \min_{j=1}^n \mu_j(z_j) \\ s.t. \quad z_j = \ & \frac{\sum_{r=1}^s u_r y_{rj}}{\sum_{i=1}^m v_i x_{ij}} \leq 1, \ j = 1, ..., n \\ & u_r, v_i \geq 0 \ \forall r, i. \end{aligned} \tag{7}$$

Because $z_j \in [0;1] \quad \forall j$, we put $z_j^L = 0$ and $z_j^U = 1$, which simplifies the membership function to $\mu_j(z_j) = z_j$. Introducing an additional variable λ, problem (7) becomes

$$\begin{aligned} & \max \lambda \\ s.t. \ & \frac{\sum_{r=1}^s u_r y_{rj}}{\sum_{i=1}^m v_i x_{ij}} \leq 1, \ j = 1, ..., n \\ & \lambda \leq \frac{\sum_{r=1}^s u_r y_{rj}}{\sum_{i=1}^m v_i x_{ij}}, \ j = 1, ..., n \\ & u_r, v_i \geq 0 \ \forall r, i. \end{aligned} \tag{8}$$

With an easy transformation finally we get

$$\begin{aligned} & \max \lambda \\ s.t. \ & \sum_{r=1}^s u_r y_{rj} - \sum_{i=1}^m v_i x_{ij} \geq 0, \ j = 1, ..., n \\ & \sum_{r=1}^s u_r y_{rj} - \lambda \sum_{i=1}^m v_i x_{ij} \geq 0 \\ & u_r, v_i \geq 0 \ \forall r, i. \end{aligned} \tag{9}$$

Let $(\mathbf{u}^*, \mathbf{v}^*)$ be the combination of input and output weights, which solves (9). Than the efficiency eff_j^* of each DMU j, $j = 1, ..., n$ can be measured by the ratio

$$\mathit{eff}_j^* = \frac{\sum_{r=1}^{s} u_r^* y_j}{\sum_{i=1}^{m} v_i^* x_j}. \tag{10}$$

Thus the evaluation of every DMU has a more objective basis than the classical DEA-approach, because all DMUs are evaluated by the *same* tuple of weights. The differences between both approaches are illustrated by an interesting case study in the following section.

4 Evaluating DMUs of a semiconductor company in Taiwan

4.1 Characteristics of semiconductor fab operation

The Taiwan semiconductor industry has developed impetuously since 1987. In the last decade, however this industrial sector endured three years of negative growth - 1996, 1998 and 2001. Our case study considers five DMUs C1, C2, C3, C4, C5 over a three year time-period from 1999 to 2001. During this time the business cycle for the respective industry sector showed a good increase in 1999-2000, but a significant decline of production in 2001.

	Input				Output		
Fab	Sal	CRM	LC	Step	GP	Mar	Waf
C1-99	2,81%	1,25%	0,76%	2,25%	2,65%	1,21%	3,58%
C2-99	9,61%	6,49%	4,19%	12,55%	3,79%	7,84%	13,79%
C3-99	7,37%	8,45%	8,28%	8,39%	5,97%	8,24%	7,15%
C4-99	6,37%	8,38%	9,44%	6,57%	9,95%	7,61%	6,36%
C5-99	4,89%	6,25%	9,04%	4,81%	3,13%	1,97%	3,50%
C1-00	2,69%	1,27%	0,76%	2,41%	6,07%	1,56%	3,48%
C2-00	10,44%	6,53%	4,19%	14,04%	6,54%	11,11%	15,81%
C3-00	8,48%	9,87%	8,42%	9,85%	6,92%	14,30%	9,28%
C4-00	7,32%	9,23%	10,46%	7,59%	7,11%	13,84%	7,54%
C5-00	7,78%	10,25%	10,16%	8,24%	5,31%	9,39%	6,70%
C1-01	1,71%	1,18%	0,76%	1,05%	5,78%	0,36%	1,59%
C2-01	8,70%	3,95%	4,19%	6,99%	7,01%	5,37%	7,77%
C3-01	7,60%	8,42%	8,49%	5,88%	8,06%	7,76%	5,26%
C4-01	7,29%	8,00%	10,55%	4,54%	13,27%	6,48%	4,45%
C5-01	6,97%	10,48%	10,32%	4,86%	8,44%	2,95%	3,73%

Table 1. Input-Output-Data.

In our study we consider four input and three output variables.

Input (per year): **Sal**ary in U.S. $, **C**osts of **Ra**w **M**aterial in U.S.$, **L**abour **C**osts in U.S. $ and no. of **Step**moves. A stepmove is a technical working step, typical for the sector.

Output (per year): **Mar**gin in U. S. $, Quantitiy of **G**ranted **P**atents, granted from the U.S. patent office, and no. of **Waf**ers processed.

For reasons of secrecy, all data in Table 1 are given in percent instead of exact quantities. The numerical calculations of the mathematical models, however, are based on the real input-output factors as listed above. So for example the first value in the Sal-column of Table 1 means that the DMU C1 in 1999 (C1-99) payed 2,81 % of the accumulated total of all salaries, all DMUs and all years. The other values read analogously.

4.2 Results

Given the data in Table 1, we solve model (3) CCR and (9) MF for each company C1-99 to C5-01. The linear program (3) and the nonlinear program (9) are solved by LINGO, a powerful professional software package [9]. The results shows Table 2. The upper part presents the solutions for the CCR-model and the lower part the results for the MF-model. All DMUs are listed in falling order, corresponding to their efficiency per year. Table 2 lists the numerical efficiencies of the five DMUs over the three years and offers some interesting interpretations.

1999					2000					2001				
CCR														
C1	C2	C4	C3	C5	C1	C2	C3	C4	C5	C1	C4	C2	C3	C5
1.	.96	.93	.79	.54	1	1	1	1	.76	1	1	.91	.83	.59
Maximal Fairness (MF)														
C1	C2	C4	C3	C5	C1	C4	C2	C3	C5	C1	C2	C4	C3	C5
1.	.87	.85	.76	.53	1	1	.94	.92	.75	.87	.78	.76	.75	.53

Table 2. Ranking by CCR and Maximal Fairness

- Obviously, the evaluation by the MF-concept for alls DMUs over the three-years period is significant lower than the self-appraisals. This effect was expected and demonstrates the fuzzy oriented concepts' superiority.

- Over the three-years period DMU C1 is the most efficient one for both approaches: CCR = (1,1,1), MF = (1,1,.87).
C5, on the other hand, is the one with lowest efficiencies among all DMUs, equaly for both approaches: CCR = (.54, .76, .59), MF = (.53, .75, .53).

- The fuzzy oriententend MF-concept reflects the business cycle during the three-years time periode much better than the classical CCR-technique does.

5 Conclusion

The powerful tool DEA – in particular the CCR-model – allows the assessment of efficiencies to DMUs in such cases where traditional approaches of productivity measurement fail. In this paper we use an extension of the classical CCR-model to a fuzzy model which allows a more objective evaluation of DMUs, than CCR does. This extension results in a nonlinear optimization programm. The philosophy of the MF-approach is demonstrated by a small real world example of semiconductor industry in Taiwan. For this the nonlinear problem was solved by professional software. Even for such professional software, larger examples may cause numerical problems. The MF-approach is a powerful alternative method superior to classical DEA, for evaluating business firms, hospitals, restaurants and other institutions, and to make them comparable in an objective way.

References

1. Birman S. V., Pirondi, P. E., Rodin, E. Y. (2003) Application of DEA to Medical Clinics, Mathematical and Computer Modeling 37, 923-936.
2. Charnes A., Cooper W. W., Rhodes E. (1978) Measuring the efficiency of decision making units, European Journal of Operations Research, 2, 429-444, North Holland Publishing Company.
3. Chiang, C. I., Tzeng, G. H. (2000) A multiple objective programming approach to data envelopment analysis, 270-284, Shi, Y. and Zeleny, M. (eds), New Frontiers of Decision Making for the Information Technology ERA, World Scientific Publishing Company, 270-284.
4. Cooper W. W., Seiford L. M., Tone K. (2000) Data Envelopment Analysis, Kluwer, Academic Publishers, Norwell, Dordrecht.
5. Easton, L., Murphy, D. L., Pearson, J. N. (2003) Purchasing performance evaluating: with data envelopment analysis, European Journal of Purchasing and Supply Management 8, 123-134.
6. Farell M. J. (1957) The measurement of productive efficiency, Journal of the Royal Statistical Society 120, 253-290.
7. Golany, B., Roll, Y. (1989) An Application Procedure for DEA, Omega Int. Journal of Management Science 3, 237-250.
8. Homburg, C. (2001) Using Data Envelopment Analysis to benchmark activities, International Journal of Production Economics, 73, 51-58.
9. Lingo (2004) www.lindo.com.
10. Rommelfanger, H.: Fuzzy Decision Support-Systeme, Springer, Berlin, 1994.
11. Shephard, D. (1970) The Theory of Cost and Production Functions, Princeton University Press, Princeton, N.J.

Verknüpfung von Standortdaten und Vegetationsmodellen über die Zeigerwerte nach Ellenberg

Eike Rommelfanger, Wolfgang Köhler

Abstract

Um aus Standortdaten, die in Form digitaler Karten aus geographischen Informationssystemen vorliegen, Prognosen über die vorkommenden Pflanzenarten abzuleiten, benötigt man ein Regelwerk zur Verknüpfung dieser Informationen. Solche Regelwerke fehlen bislang. In Form linguistischer Terme liegt zwar ein bewährtes, anerkanntes Informationssystem vor, das Zeigerwertsystem nach Heinz Ellenberg, um dieses Zeigerwertsystem aber nutzbar zu machen, muss es zunächst übersetzt werden. Das hier vorgestellte System stellt diese Übersetzung in Form eines Fuzzy-Expertensystems dar. Durch die Verknüpfung von Standortdaten und den Zeigerwerten nach Ellenberg über die Übersetzung der Zeigerwerte, kann so eine Verknüpfung von Standortdaten und den vorkommenden Pflanzen abgeleitet werden. Diese Verknüpfung erfolgt in Form eines Filters, der für jede untersuchte Pflanzenart die Auftrittschance auf dem Standort in Form einer Zugehörigkeitsbeschreibung bestimmt. Auf dieser Information lassen sich eine Vielzahl weiterer Ansätze aufbauen.

1. Einleitung

Im Rahmen des Sonderforschungsbereichs 299 „Landnutzungskonzepte für periphere Regionen“ (Frede et al. 2002) wird das Verbundmodell ITE^{2}M entwickelt. Es beinhaltet die Modelle ProLand (Kuhlmann et al. 2002), zur Landnutzungsprognose anhand wirtschaftlicher Faktoren, das Modell SWATG zur Prognose der Auswirkungen von Landnutzungswechseln auf den Wasserhaushalt und das Modell ANIMO (Steiner u. Köhler 2003), zur Prognose der Auswirkungen von Landnutzungsänderungen auf die Artenvielfalt. Die gemeinsame Grundlage der Modelle ist das digitale Kartenmaterial, welches wichtige Standortinformationen beinhaltet.

Die räumliche Modellierung von biologischer Diversität ist durch Verwendung von Geographischen Informationssystemen (GIS) möglich. Es gibt jedoch noch kein System, welches ausreichende Informationen über Gefäßpflanzenvorkommen aus digitalen Standortdaten generieren kann. Hierzu ist sowohl ein komplexes Regelwerk über die Anpassung von Pflanzen an bestimmte Standorteigenschaften, als auch ein System zur Anwendung dieses Regelwerkes nötig.

Das von Heinz Ellenberg entwickelte System der Zeigerwerte von Pflanzen in Mitteleuropa, ist ein solches Regelwerk (Ellenberg et al. 1992). Es betrachtet die 8 Standorteigenschaften Nährstoffverfügbarkeit, Feuchte, Temperatur, Kontinentalität, Licht, Reaktion, Salzgehalt und Schwermetallbelastung. Den einzelnen Pflanzenarten ist pro Standorteigenschaft je ein Zeigerwert zugewiesen. Diese Zeigerwerte sind ordinal skaliert und durch verbale Beschreibungen definiert. Ein System zur Anwendung dieses Regelwerkes muss die verbal beschriebenen Zeigerwerte mit den metrischen Daten der geographischen Informationssysteme verknüpfen. Ansätze hierzu wurden von Wamelink et al. (1998) und von Ertsen et al. (1998) veröffentlicht. Das Ziel dieser Ansätze war eine Kalibrierung der Zeigerwerte durch Darstellung der Zeigerwerte anhand metrischer Bezugsgrößen. Schwachstellen dieser Ansätze sind das Fehlen metrisch messbarer, eindeutig zuweisbarer Bezugsgrößen, ein sehr hoher Bedarf an Datenmaterial und die ordinale Skalierung der Zeigerwerte, die einen Zusammenhang zwischen Zeigerwert und metrischer Bezugsgröße schwer erfassen lässt. Der hier verwendete Ansatz geht umgekehrt vor.

Da sich Fuzzy-Expertensysteme besonders eignen, verbal beschreibbare Zusammenhänge in einem System abzubilden, verfolgt dieser neue Ansatz die Strategie über die Fuzzifizierung der Standortparameter und mehrere gekoppelte Inferenzen das Zeigerwertsystem nach Ellenberg mit den Standortinformationen zu koppeln. Dabei wird jedem Zeigerwert eine Zugehörigkeitsbeschreibung zugewiesen, die zwischen „keine Zugehörigkeit zum Standort“ und „Hohe Zugehörigkeit zum Standort“ liegt.

Aus diesem Regelsystem wird ein Filter entwickelt, der aus einer Datenbank mit den Pflanzenarten des Grünlands und ihren Zeigerwerten, die Pflanzenarten herausfiltert, welche auf einem gegebenen Standort wachsen können. Über diese neue Datenbasis kann eine breite Schnittstelle zwischen den Verbundmodellen ProLand und ANIMO geschaffen werden, die die Verknüpfung der Modelle weiter optimiert.

2. Problemstellung

Um eine Schätzung des Artenreichtums der Gefäßpflanzen des Grünlands abhängig von der Nutzung der Fläche zu bewerkstelligen, muss sowohl die Standorteignung für die Pflanzen als auch die Nutzung der Fläche betrachtet werden. Die Nutzung der Fläche wirkt direkt auf die Standorteigenschaften, beispielsweise ändert sich die Nährstoffversorgung durch Düngung und die Bodenreaktion durch Kalkung, zur Erhöhung des pH-Werts.

Ein System, welches diese Aufgabe erfüllen kann muss demnach zunächst überprüfen können, für welche Pflanzen sich der Standort unabhängig der Nutzung eignet. Einfließende Parameter sind unter anderem Hangneigung und die

Himmelsrichtung in welche ein Hang exponiert ist (Exposition), sowie Bodenart und Niederschlag. Aus diesen Parametern muss eine Standorteignung für die untersuchten Pflanzenarten des Grünlands ermittelt werden, die durch nutzungsabhängige Standortveränderungen, wie die Düngung erweitert wird. Die Entwicklung eines mathematischen Modells, welches mit Hilfe von Formeln die komplexen Beziehungen der einzelnen Pflanzenarten abbildet, ist nahezu unmöglich, da der Bedarf an Messdaten ungeheuer groß wäre. Ein bereits existierendes Regelwerk, das die Standortansprüche einzelner Pflanzen abbildet ist das Zeigerwertsystem nach Ellenberg. Das System beschreibt die Standortansprüche der Gefäßpflanzen Mitteleuropas anhand linguistischer Variablen. Durch eine Verknüpfung der Zeigerwerte mit den einfließenden Standortparametern ließe sich eine Verknüpfung von Standortparametern und Informationen über die Auftrittschance einzelner Pflanzenarten ableiten.

Auf dieser Basis aufbauend können weitere Ansätze, wie die Betrachtung der Ausbreitungsmechanismen, Stressfaktoren durch Mahd, Verbiss und Tritt, sowie natürliche Häufigkeit, beziehungsweise Seltenheit einer Pflanze entwickelt werden.

3. Ziele

Das Fuzzy-Expertensystem muss aus den, in der Datenbank vorhandenen Pflanzenarten, jene ausschließen können, die nicht auf dem Standort vorkommen können. Wichtig ist jedoch vor allem, dass keine Pflanze ausgeschlossen wird, die unter Umständen doch an dem Standort wachsen kann. Da das Vorkommen einer Pflanzenart über die Standorteignung hinaus von weiteren Faktoren wie aktuelle Nutzung und Nutzungsgeschichte des Standorts, sowie artspezifischer Häufigkeit der Pflanze abhängt, kann diese erste Phase des Systems nur einen Ausschluss von Pflanzenarten gewährleisten, welche aufgrund der betrachteten Faktoren nicht vorkommen können. Aufbauend auf dieser Basis kann das System erweitert werden.

Die Umsetzung des Systems erfolgt über eine Datenbank. Um das System möglichst flexibel zu gestalten werden alle Schätzungen zur Fuzzifizierung der Parameter in gesonderte Tabellen eingegeben, die bei Bedarf geändert werden können, sollten sich neue Erkenntnisse aus der Validierung oder aus Expertenmeinungen ergeben.

Die grundsätzliche Aussage soll, über die Information zu den einzelnen Pflanzenarten hinaus, Größen ausgeben, welche die Artenvielfalt in Bezug auf Gefäßpflanzen anschaulich beschreibenden. Besonders interessant sind dabei Betrachtungen zu unterschiedlichen Gruppen von Pflanzenarten. So sollen die linguistischen Terme „gering", „mittel" und „hoch" die Größen „Anzahl vorkommender Arten" und „Anzahl vorkommender Arten der roten Liste" ebenso beschreiben, wie die Anzahl vorkommender Arten bestimmter funktioneller Artengruppen, die in bestehende Module des Modells ANIMO einfließen können.

Durch die Beschreibung des Outputs anhand anschaulicher Größen soll die Erstellung neuer digitaler Karten möglich werden. Durch die Verknüpfung mit dem Modell ProLand innerhalb des Verbundmodells ITE^{2}M können die Auswirkungen agrarpolitischer Entscheidungen, wie beispielsweise die Höhe der Flächenprämien einer Kulturart auf eine bestimmte Höhe, auf ihre Auswirkung auf das Angebot an Lebensraum für bestimmte Arten untersucht werden.

4. Zeigerwerte nach Ellenberg

Die Zeigerwerte nach Ellenberg stützen sich hauptsächlich auf Feldbeobachtungen. Sie entstanden über mehrere Jahre durch Untersuchung der Standorte an denen einzelne Pflanzenarten vorkommen. Das Zeigerwertsystem umfasst nahezu alle Pflanzen Mitteleuropas und weist jeder Pflanze eine Beschreibung ihrer Standortoptima zu, bezogen auf die Standortfaktoren Temperatur, Licht, Kontinentalität, Reaktion, Nährstoffverfügbarkeit, Salz- und Schwermetalltoleranz sowie Feuchte. Die Beschreibung für den Zeigerwert Feuchte 5 ist beispielsweise „Frischezeiger, Schwergewicht auf mittelfeuchten Böden, auf nassen sowie auf öfter austrocknenden Böden fehlend".

Die Standortoptima beziehen sich auf die ökologischen Optima der Pflanzen. Das ökologische Optimum bezeichnet die Standortbedingungen unter denen sich eine Pflanzenart am besten gegen andere Pflanzenarten behaupten kann, ihre Konkurrenzkraft gegenüber anderen Arten also am stärksten ist. Erfüllt ein Standort diese Bedingungen, so hat eine Pflanze, sofern sie ihre Sporen an dem Standort ausbringen kann, gute Chancen tatsächlich an diesem Standort vorzukommen.

Die Zeigerwerte dienen in der Ökologie hauptsächlich zur Beschreibung von Standorten in Form von Zeigerwertspektren. Die Pflanzen, die bei einer Vegetationsaufnahme gefunden wurden, werden beispielsweise zur Beschreibung der Feuchtigkeitsverhältnisse in einer Verteilung ihrer jeweiligen Zeigerwerte dargestellt, um einen Eindruck von der Fläche zu vermitteln. In ihrer stark verbreiteten Anwendung weit über Deutschland hinaus, der großen Zahl betrachteter Pflanzenarten, als auch in ihrer breiten Akzeptanz in der Ökologie liegen die Stärken des Zeigerwertsystems, die es für die Verwendung in diesem System interessant machen.

5. Expertensystem

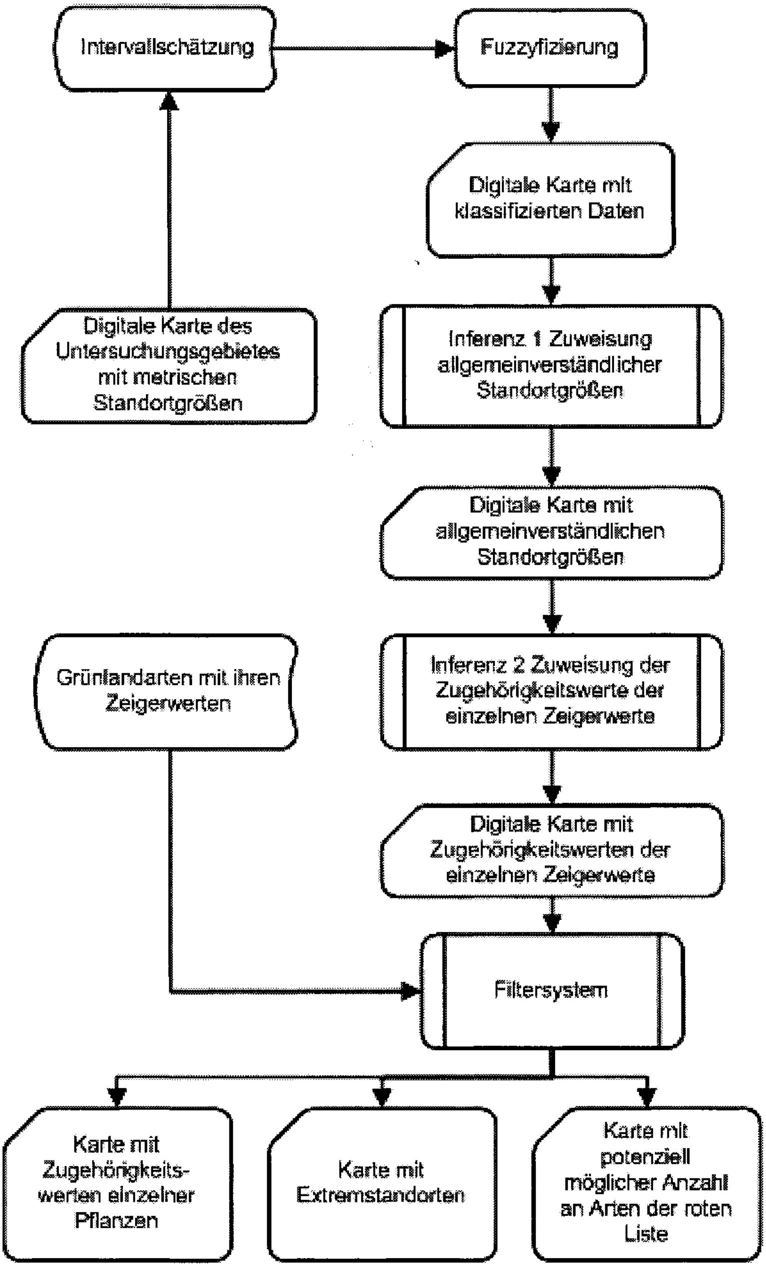

Intervallschätzung
Fuzzyfizierung
Digitale Karte mit klassifizierten Daten
Digitale Karte des Untersuchungsgebietes mit metrischen Standortgrößen
Inferenz 1 Zuweisung allgemeinverständlicher Standortgrößen
Digitale Karte mit allgemeinverständlichen Standortgrößen
Grünlandarten mit ihren Zeigerwerten
Inferenz 2 Zuweisung der Zugehörigkeitswerte der einzelnen Zeigerwerte
Digitale Karte mit Zugehörigkeitswerten der einzelnen Zeigerwerte
Filtersystem
Karte mit Zugehörigkeits-werten einzelner Pflanzen
Karte mit Extremstandorten
Karte mit potenziell möglicher Anzahl an Arten der roten Liste

Abb. 1. Schema der Systems

5.1 Fuzzyfizierung der Standortparameter

Das digitale Kartenmaterial enthält Standortparameter, die oft als metrische Größen angegeben sind. Die Fuzzyfizierung der Standortparameter kann sich bei der Mehrzahl der Faktoren auf Klassendefinitionen aus der Literatur stützen. So ist z.B. die nutzbare Feldkapazität offiziell in fünf Klassen von „sehr gering" bis „sehr hoch" eingeteilt Kuntze et al. 1994). Die Höhe ist ebenfalls in linguistischen Termen beschrieben. Diese sind vegetationsspezifisch und sind absteigend von „nival" über „alpin", „subalpin", „montan" und „submontan" bis „kollin" eingeteilt Pfadenhauer 1997). Die Höheneinteilung ist jedoch unscharf und stark abhängig von dem betrachteten Gebiet.

Ebenfalls aus Literaturangaben lässt sich grob die Klassifizierung der Hangneigung ableiten. All diese Einteilungen sind in Form von scharfen Mengen dargestellt. Eine Aufweichung dieser Grenzen ist für die Integration in das System nötig und kann sich unter anderem auf Schwankungen der Literaturangaben und Expertenschätzungen stützen.

Der Faktor Niederschlag wird in die Klassen „gering", „mittel" und „hoch" eingeteilt. Diese Klassen werden speziell für das Untersuchungsgebiet gewählt und stützen sich auf die Klimaaufzeichnungen der vergangenen Jahre.

Ein Teil der Parameter liegt in Form nominalskalierter Klassen vor, so zum Beispiel die Exposition und die Bodenart. Diese werden in ordinalskalierte Klassen überführt. Die Rangfolge wird bei der Exposition aus der Sonneneinstrahlung und bei der Bodenart aus der Wasserdurchlässigkeit abgeleitet.

Andere in den Karten vorhandenen Parameter sind bereits klassifiziert. So liegt beispielsweise der Faktor „Basenversorgung des Bodens" bereits in klassifizierter Form vor.

5.2 Inferenz

Um eine nachvollziehbare Verknüpfung mit den Zeigerwerten nach Ellenberg zu erreichen muss die Inferenz des Systems in zwei Teile aufgespalten werden. Zunächst werden aus den klassifizierten Standortparametern verständliche Einflussgrößen abgeleitet, aus denen Faktoren wie Feuchtigkeit oder Temperatur erschlossen werden können. Aus diesen Einflussgrößen, kombiniert mit Standortfaktoren, wird anschließend, in einer zweiten Phase, die Inferenz zur Zuweisung der Zeigerwerte erstellt.

5.21 Inferenz 1 – Ableitung klimatischer und geländespezifischer Parameter

Dieses vorgeschaltete Regelwerk dient der Vereinfachung des Regelwerks, einer erhöhten Nachvollziehbarkeit und einer deutlichen Verringerung der Anzahl an Regeln im Hauptteil der Inferenz.

Hier sollen die Standortgrößen Verdunstung, Oberflächenabfluss, Sonneneinstrahlung und gespeichertes Wasser abgeleitet werden. Bei der Ableitung dieser Größen wird ein herkömmliches fuzzy-control Regelwerk verwendet. Die einfließenden Faktoren weisen über Wenn-Dann Operatoren auf eine Klasse der Outputstandortgröße hin.

Die einfließenden Faktoren für die Sonneneinstrahlung sind Exposition und Hangneigung. Die Sonneneinstrahlung ist an einem steilen Südhang am stärksten. Bei geringerer Hangneigung oder einer anderen Himmelsrichtung wird die Einstrahlung geringer sein. Bei einem steilen Nordhang wiederum ist die Einstrahlung am geringsten und wird mit geringerer Hangneigung zunehmen. Ähnlich wird die Verdunstung abgeleitet, in die zusätzlich die Faktoren Bodenart und Nutzung der Fläche eingehen.

Der Oberflächenabfluss wird aus der Bodenart und der Hangneigung bestimmt. Je schlechter der Niederschlag einsickern kann und je steiler der Hang ist, desto größer wird der Anteil des Niederschlags sein, der an der Oberfläche abfließt. Wie viel Wasser gespeichert werden kann hängt von der Speicherfähigkeit des Bodens und der Menge des Niederschlags ab.

5.22 Inferenz 2 – Ableitung der Zugehörigkeitsklassen der einzelnen Zeigerwerte

In diesem Teil des Systems wird den einzelnen Zeigerwerten eine Zugehörigkeit zugewiesen. Die Zeigerwerte können als linguistische Terme betrachtet werden, besitzen jedoch sehr heterogene Beschreibungen, sodass jeder einzelne gesondert betrachtet werden muss. Für jede beschriebene Standorteigenschaft muss ein gesondertes Regelwerk, mit unterschiedlichen Einflussparametern, erstellt werden. Jede Standorteigenschaft besitzt mindestens 9 Zeigerwerte. Für jeden muss geprüft werden, für welche Kombinationen der Einflussgrößen der Wert „keine Zugehörigkeit“, für welche er „geringe Zugehörigkeit“, „mittlere Zugehörigkeit“ oder „hohe Zugehörigkeit“ aufweist.

Als Beispiel sei Zeigerwert Feuchte 7 angeführt. Dieser Zeigerwert ist durch den linguistischen Term „Schwergewicht auf gut durchfeuchteten, aber nicht nassen Böden“ definiert. Die Einflussfaktoren für Feuchte sind in dem Regelwerk die Verdunstung, die Menge gespeicherten Wassers, der Oberflächenabfluss und die Staunässeneigung. Neigt der Boden zu Staunässe muss die Zugehörigkeit des Zeigerwertes gering bis null sein, da nasse Böden ausgeschlossen werden. Herrscht an dem Standort starke Verdunstung und wird nur eine geringe Menge Wasser gespeichert, so muss die Zugehörigkeit des Zeigerwertes ebenfalls null sein, da er dann nicht gut durchfeuchtet sein kann. Auch ein hoher Oberflächenabfluss wirkt sich negativ auf die Zugehörigkeit aus.

5.3 Filtersystem

Aus dem entwickelten Regelwerk kann gefolgert werden, welche Zeigerwerte auf bestimmten Standorten nicht vertreten sein können. Aus dieser Information lässt sich ableiten, welche Pflanzen auf diesen Standorten nicht wachsen können. Diese Pflanzen werden über ein Standortfiltersystem ausgeschlossen.

Der Filter wird an eine Datenbank angelegt, welche Informationen zu den Gefäßpflanzen Mitteleuropas beinhaltet. Außer den Zeigerwerten sind beispielsweise auch Informationen zum Gefährdungsstatus enthalten.

Die Pflanzen, die nicht herausgefiltert werden, lassen zahlreiche Schlussfolgerungen über den betrachteten Standort zu. Die gewonnenen Informationen können in Form von Karten anschaulich gemacht werden.

Durch Vergleich der Ergebnisse des Systems mit der realen Situation im Gelände kann das System auf Fehler untersucht werden. Das System darf auf keinem Standort eine Pflanze ausschließen, die tatsächlich dort vorkommen kann. Sollten sich Fehler dieser Art zeigen, so muss geprüft werden, ob ein konzeptioneller Fehler in der Fuzzifizierung der Eingangsgrößen vorliegt oder ob die Inferenzen fehlerhafte Regeln beinhalten.

Literaturverzeichnis

Ertsen ACD, Alkmade JRM, Wassen MJ (1998) Calibrating Ellenberg indicator values for moisture, acidity, nutrient availability and salinity in the Netherlands. Plant Ecology 135: 113-124.

Frede HG, Bach M, Fohrer N, Möller D, Steiner N (2002) Multifunktionalität der Landschaft – Methoden und Modelle. PGM 2002/6, 58-63.

Kuhlmann F, Möller D Weinmann B (2002) Modellierung der Landnutzung: Regionshöfe oder Raster-Landschaft? Berichte über Landwirtschaft Bd. 80, 3.

Kuntze H, Roeschmann G, Schwerdtfeger G (1994) Bodenkunde. Eugen Ulmer, Stuttgart.

Pfadenhauer J (1997) Vegetationsökologie. IHW-Verlag, Eching.

Steiner N, Köhler W (2003) Effects of landscape patterns on species richness - a modelling approach. Agriculture, Ecosystems & Environment 98, 353-361.

Wamelink GWW, van Dobben HF, van der Eerden LJM (1998) Experimental calibration of Ellenberg's indicator value for nitrogen. Environmental Pollution 102, 371-375.

Extracting Rules from Support Vector Machines

Klaus B. Schebesch and Ralf Stecking

Institut für Konjunktur- und Strukturforschung,
Universität Bremen, D-28359 Bremen, Germany

Abstract. Support Vector Machines (SVM) from statistical learning are very powerful methods which can be used as (e.g.binary) classifiers or discriminators in a wide range of applications. Advantages of SVM are that weak prior assumptions about both model and data suffice. Moreover, optimization of the SVM essentially regularizes the emerging data model by restricting the model to special data points, the support vectors, usually a small subset from the training data. In our paper we discuss ways of detecting informative and typical subsets from SVM solutions, with the aim of extracting simple rules.

1 Introduction

Classification of labeled high dimensional data is an important step within computational decision support in many industries. Classification rules obtained from large data sets can often be used as decision rules. An example is the decision of whether to accept or to reject a new credit applicant by a bank. Using personal records on many past credit applicants as a training set, a method from Discriminant Analysis extracts a classification rule, which is used to forecast whether a new credit applicant is likely to default. Among the many methods for separating data into classes we mention Linear Discriminant Analysis (LDA), which is computationally cheap and most suitable for linearly separable data but also more expensive methods like Support Vector Machines (SVM), which can "lift" arbitrarily complex data into high dimensional feature spaces until all nonlinearities are disentangled such that linear separation becomes possible [9]. SVM identify special data points (the support vectors) from the set of input data, which describe the boundaries between the cases. However, as an essentially geometrical concept in high dimensional space they are not of any direct use to the practitioner.

There are several alternative approaches which express classification by a small set of conjunctions / disjunctions or other standardized "linguistic" rules [5], or which link fuzzy decision rules with certain non-linear SVM [2]. However, many of these approaches are "open ended" as they employ evolutionary search and genetic programming to discover useful rules from the vast number of feasible rules [3] and, hence, they also tend to be unacceptable for practitioners.

Classification and regression trees (CART) in the sense of [1] are a widely accepted procedure for building decision rules based on recursively placing split points on the values of single input variables. This recursive partitioning is simple and inexpensive, but compared to general rule systems its expressive power is limited.

While using SVM has been shown to be useful for credit scoring [10], [8] one would still like to use the attractively simple features of recursive decision trees as a back end for practitioners. In the remaining part of the paper we therefore propose to use the support vectors of an SVM solution as subset preselection for computation of the decision trees. After a short description of the SVM mechanism and different types of support vectors, we first show the reasonability of the approach with some two dimensional "toy" examples and then proceed with the results on real life credit data.

2 SVM and different types of support vectors

Following [8] we give an abridged exposition of the SVM mechanism. Given $N > 0$ training cases $\{x_i, y_i\}_{i=1}^{N}$, with input data $x_i \in X \subset \mathbb{R}^m$ (e.g. the description of a credit applicant) and the associated class labels $y_i \in \{-1, +1\}$ (e.g. the recorded defaulting behavior), assume a feature space $\mathcal{F}$, with dim($\mathcal{F}$) $\geq$ dim(X), i.e. at least as expressive as input space X. There should also exist a map which transform points x into points u. Now imagine $u \in \mathcal{F}$ being such that a given, but arbitrarily distributed data set from X can now be linearly separated in $\mathcal{F}$. The central ingredient of SVM is to linearly separate points in $\mathcal{F}$ by a **maximal margin**, i.e. points form one class described by $\langle u_{(-1)}, w\rangle + b \leq -1$ should be as much apart as possible from the points of the other class $\langle u_{(+1)}, w\rangle + b \geq 1$, which finally amounts to maximize $2/||w||$. While this would minimize the expected misclassification rate, in practice, hard margins are replaced by soft margins by minimizing $||w||$ or

$$C\sum_{i=1}^{N}\zeta_i \;+\; \min_{w,b}\frac{1}{2}\langle w, w\rangle \quad \text{s.t.} \quad y_i[\langle u_i, w_i\rangle + b] \geq 1 - \zeta_i, \quad i = 1, ..., N,$$

with slacks $\zeta_i \geq 0$. Capacity $C > 0$ now controls the misclassification rate which is used to avoid over-fitting to the data. The associated dual SVM, which disposes of the use of unknown space $\mathcal{F}$, reads

$$\max_{\alpha}\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k(x_i, x_j) \quad \text{s.t.} \quad \sum_{i=1}^{N}y_i\alpha_i = 0, \quad C \geq \alpha_i \geq 0.$$

The "kernels" $k(.,.)$ are just scalar products of the inputs $\langle x_i, x_j\rangle$ in the case of linear SVM, i.e. if $u_i = x_i$ or $(X = \mathcal{F})$. In general $(X \neq \mathcal{F})$, they are chosen from some set of nonlinear functions like e.g. $\exp(s||x_i - x_j||^2)$, $s > 0$.

A data point x_i is called a **support vector** if $\alpha_i > 0$ (i.e. the point i is a binding inequality in the primal). Note that there are at least two kinds of support vectors. In case of $\alpha_i < C$ the support vector i is located on the margin, and in case of $\alpha_i = C$ the support vector is within the margin (i.e. $\zeta_i > 0$) where the probability of correctly predicting the class membership of a new data point is low. Support vectors of the first kind define the direction of the linear decision rule in feature space.

For a new case $x \in X$, a sign rule on the SVM outputs computes the class label by $y^*(x) = \text{sign}\{S(x)\} = \text{sign}\left\{ \sum_{i=1}^{N} \alpha_i^* \langle x, x_i \rangle + b^* \right\}$, where $\alpha^* \geq 0$ and $b^* \in \mathbb{R}$ are read off or easy derived from the dual solution.

3 Combining linear SVM with CART

In order to illustrate the effect of using CART on subsets which are the results of linear SVM we use a two-dimensional example of 244 randomly placed positive and negative cases (left plot of fig. 1). The class of positive cases contains much less (63) examples, and the position of some positive case where modified such that the set of positives looks "star-like" with outliers. Similarly to the credit scoring data this set is not linearly separable (decision set in left plot of fig. 1) and the linear SVM produces many (183) support vectors of both kinds. Next, recursive partitions are computed by CART on the whole data set and on two different data subsets resulting from the linear SVM. The two subsets are the bounded support vectors and the unbounded support vectors plus non support vectors, respectively (middle and right plots of fig. 1).

The linear SVM misclassifies 58 (out of 244) cases. CART on all 244 data points misclassifies 32 cases with seven simple rules. CART on the 94 bounded support vectors misclassifies 20 cases with six simple rules, which are shown as a composite rule for the negative cases in the middle plot of fig. 1 (shaded areas). One of the rules (light shade) contains the same number of positives and negatives (complete uncertainty), thus contributing with seven misclassified positives. Finally, CART on the unbounded support vectors and on all non support vectors (150 in total) classifies all cases correctly using three simple rules (right plot of fig. 1).

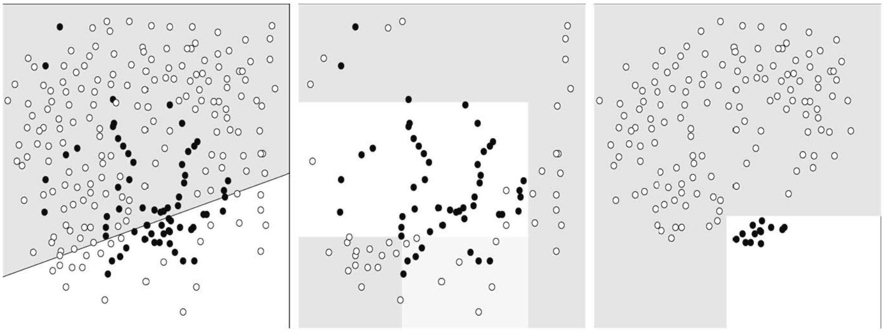
We note that in this example the SVM is not optimized and all results are in sample. To take advantage of the procedure for new unlabeled data, however, some region description of the subsets is needed! In the next chapters we will extend the procedure to high dimensional real-life credit scoring data using out-of-sample validation.

4 Extracting Rules from SVM: Empirical results

4.1 Credit Scoring Data

We use a data set of 69775 cases describing applicants for building and loan credits. A total of 2336 out of 69775 applicants (3.3%) defaulted on their

Fig. 1. Example with negative (white) and positive (black) cases, not separable in two dimensions. Left: All data and the decision set $S(x) < 0$ (shaded) of a linear SVM. Middle: CART on the bounded support vectors ($\alpha_i = C$). The shaded areas are the combined rule for negative cases. The lightly shaded area denotes a region with equal number negative and positive cases (complete uncertainty). Right: CART on all non support vectors plus the unbounded support vectors ($\alpha_i < C$).



repayments and were classified as "bad". The goal of a binary credit scoring function is to assign new credit applicants to good or bad risk classes. There are six metric input variables describing the individual historical credit performance of each applicant and one binary output variable describing the state of credit (good or bad). Input variable transformations are necessary to correct for effects of scale and will be done by $x_i' = \frac{x_i - x_{MIN}}{x_{MAX} - x_{MIN}}$, with x_i' as rescaled input field and x_{MIN} and x_{MAX} the minimum respectively maximum value for variable x. Each rescaled input variable is bounded between 0 and 1.

4.2 SVM Model building

By using SVM with linear kernel, upper bound C for α is the only parameter to be set in advance. As a rule of thumb C should not be smaller than ten times the input dimension of the model [9]. In our case a linear SVM with $C = 60$ was built to predict the state of credit. Additionally, to correct for unequal share of good and bad credit applicants in the data set, a cost factor was set to allow for more accuracy towards the classification of bad applicants [7].

4.3 Subset preselection

SVM divide the patterns of the input space into three subsets [8]: the essential support vectors with $0 < \alpha_i < C$ (where C is the user set upper bound for α_i) used for the functional representation of the margin, the "bounded" support vectors with $\alpha_i = C$ and the non support vectors with $\alpha_i = 0$. In the following the essential support vectors will be called *informative* patterns. Non support vectors that can be separated without any error by the SVM will be called *typical* patterns. In the non separable case, where SVM with soft margin hyperplanes are used, bounded support vectors occur. Such support vectors cannot be separated well by the SVM. They will be called *critical* patterns.

4.4 Extracting Rules with CART from SVM

Extracting rules with *CART* from SVM will be done in two different ways. First, one can use only the essential support vectors (informative patterns) as input for the tree method. This will enable the classification tree to construct a set of decision rules along the margin of the separating hyperplane of the SVM. In our case 22 essential support vectors are found. They can be separated without any error by four simple rules, based on three variables $X3$, $X5$ and $X6$ from the input set (see table 1).

Table 1. Rule extraction from informative patterns with CART.

Rule 1 for bad:	
`if X6 < 0.6786 and X5 >= 0.7748`	`then bad`
Rule 2 for bad:	
`if X6 >= 0.6786 and X3 < 0.7951`	`then bad`
Rule 1 for good:	
`if X6 < 0.6786 and X5 < 0.7748`	`then good`
Rule 2 for good:	
`if X6 >= 0.6786 and X3 >= 0.7951`	`then good`

A second way is to use the SVM as a preprocessing method, identifying typical and critical patterns. *CART* then is performed by using only typical patterns. This procedure leads to a small set of essential rules with low error. Table 2 shows the results. Three rules can be extracted, based on two variables X3 and X6. Out of 69775 credit applicants 55613 cases are critical patterns and therefore were eliminated. The remaining 14162 cases can be classified by three rules, with an error of 0.44%.

4.5 Extracting Rules with Kohonen Mapping from SVM

The self-organizing map (*SOM*) from Kohonen [6] is a well known neural network type that can be used to perform spatially organized clustering. The *SOM* generates a two dimensional representation of a multidimensional input set. Similar cases are grouped together in one or more (neighbouring)

Table 2. Rule extraction from typical patterns with CART.

Rule 1 for bad:	
`if X6 >= 0.6378 and X3 < 0.6729`	`then bad`
Rule 1 for good:	
`if X6 < 0.6378`	`then good`
Rule 2 for good:	
`if X6 >= 0.6378 and X3 >= 0.6729`	`then good`

neurons. We use *SOM* with subsequent preprocessing via SVM. By using only support vectors and non support vectors one can intuitively expect to get a mapping with homogenous regions of good and bad credit applicants. Each region consists of a number of neighbouring neurons (or prototypes) where each prototype is defined by its centre. Prototype–based rules then can be generated in the following way: First, identify the region(s) of bad applicants. For each neuron of the bad region: Calculate the distances between prototype centre and support vectors. Support vectors are lying exactly on the margin between good and bad credit applicants. The maximum distance between centre and support vector then will be used as a threshold. If the distance between an input pattern and a "bad" prototype centre is below this threshold, the pattern will be classified as bad [4].

Table 3. Kohonen Mapping using informative and typical patterns: average profiles of two bad prototypes compared to the average of good prototypes.

Kohonen Mapping - Prototype centres						
	X1	X2	X3	X4	X5	X6
bad–9.4	0.1967	0.5506	0.1632	0.2221	0.3193	0.8478
bad–9.5	0.5212	0.5761	0.4663	0.2639	0.4145	0.8343
average good	0.1400	0.4690	0.2077	0.6037	0.2722	0.2228

One bad region with two neighbouring neurons (out of a total of 70 neurons of the whole map) could be identified. The share of bad applicants connected with these neurons is 99.35% (461 out of 464 credit applicants). The profiles for the three prototypes bad–9.4 and bad–9.5 (the last two numbers indicating the coordinates on the 10 × 7 Kohonen map) can be seen in table 3. The three rules for bad credit applicants are:

- `if` $D(\text{bad–9.4}; X) < 0.4148$ `then bad`
- `if` $D(\text{bad–9.5}; X) < 0.3637$ `then bad`

- `else` `then good`

$D(\text{bad–9..}; X)$ is the Euclidian distance between the unlabeled credit applicant X and the bad region centre of bad–9.4 and bad–9.5 respectively.

4.6 Tenfold Cross Validation Classification Results

To perform tenfold cross validation the whole data set is divided into ten equally sized intervals at random. Subsequently each interval is being used as an out–of–sample validation set for the model that was generated using the remaining nine intervals as a training set. By repeating this ten times the whole data set is being used as an out–of–sample validation set.

Table 4. Overview of tenfold cross validation model performance. There are 2336 bad and 67439 good credit applicants in total. Bad credit applicants classified as good are called *Bad accepted*, good credit applicants classified as bad are called *Good rejected*. *Total error* is the percentage of total wrong in relation to all 69775 credit applicants. The last column shows the minimum, maximum and average number of rules for every rule–based model.

	Tenfold Cross Validation					
					Total	*No. of rules*
Model	*Bad accepted*		*Good rejected*		*error*	*[MIN; MAX; AVE]*
SVM	688	(29.5%)	28009	(41.5%)	41.1%	–
LDA	766	(32.8%)	25290	(37.5%)	37.3%	–
CRTree (Full Set)	799	(34.2%)	22019	(32.7%)	32.7%	[4; 10; 6.9]
CRTree (Informative)	956	(40.9%)	23993	(35.6%)	35.8%	[2; 4; 2.9]
CRTree (Typical)	1203	(51.5%)	16913	(25.1%)	26.0%	[3; 3; 3.0]
Kohonen	847	(36.3%)	31682	(47.0%)	46.6%	[2; 3; 2.8]

The following models are compared in the sequel: (1) Linear Support Vector Machine, (2) Linear Discriminant Analysis (LDA), (3) C&R-Tree without subset preselection, (4) C&R-Tree using informative patterns only as input, (5) C&R-Tree using typical patterns only as input and (6) Kohonen Mapping using informative and typical patterns. In terms of total error it can be seen that CRTree (Typical) outperforms all other models. CRTree (Informative) does better than SVM and LDA but is worse than CRTree (Full Set). Kohonen Mapping finally is worse than any other model. A closer look should be taken towards the number of generated rules. Whereas CRTree (Full Set) has a low error, it generates quite a high number of rules (an average of about 7) with a range of 4 up to 10 rules over all ten folds. CRTree (Informative) generates only three rules in average with a range of 2 up to 4. The most stable results are from CRTree (Typical). For each of the ten folds three (almost identical) rules are generated. Kohonen Mapping generates the smallest number of rules, but results in a poor out–of–sample classification.

5 Conclusions and outlook

It was shown that benefits can be gained from applying classification trees on data subsets selected by SVM solutions. Using these rule–based models is attractive for practitioners, it can reduce overall misclassification rate and it also leads to a small and stable solution. For higher dimensional data some further decision set analysis seems to be imperative. A future direction will be to use graphs to further characterize data subsets as e.g defined by classes and support vectors of linear and nonlinear SVM.

References

1. BREIMAN, L., FRIEDMAN, J.H., OLSHEN, R.A. and STONE, C.J. (1984): *Classification and Regression Trees.* Wadsworth & Brooks, Pacific Grove, California.
2. CHEN, Y. and WANG, J.Z. (2003): Support Vector Learning for Fuzzy Rule-Based Classification Systems, Working Paper, Department of Computer Science and Engineering, The Pennsylvania State University, 1–30
3. DeFALCO, I., Della CIOPA, A. and TARANTINO, E. (2002): Discovering interesting classification rules with genetic programming. *Applied Soft Computing* **1**: 257–269
4. DUCH, W., SETIONO, R. and ZURADA, J.M. (2004): Computational Intelligence Methods for Rule–Based Data Understanding. *Proceedings of the IEEE, Vol.92, No.5, May 2004*, 771–805
5. HOFFMANN, F., BAESENS, B., MARTENS, J., PUT, F. and VANTHIENEN J. (2002): Comparing a genetic fuzzy and a neurofuzzy classifier for credit scoring. Royal Institute of Technology, Center for Autonomous Systems, 8pp, `http://www.nada.kth.se/ hoffmann/FLINS2002.pdf`
6. KOHONEN, T. (2001): *Self–Organizing Maps.* Springer, New York.
7. SCHEBESCH, K.B. and STECKING, R. (2003a): Support Vector Machines for Credit Scoring: Extension to Non Standard Cases. Submitted to Proceedings of the 27th Annual Conference of the GfKl 2003.
8. SCHEBESCH, K.B. and STECKING, R. (2003b): Support Vector Machines for Credit Applicants: Detecting Typical and Critical Regions, in: *Credit Scoring & Credit Control VIII*, Credit Research Center, University of Edinburgh, 3–5 September 2003, 13pp
9. SCHÖLKOPF, B. and SMOLA, A. (2002): *Learning with Kernels.* The MIT Press, Cambridge.
10. STECKING, R. and SCHEBESCH, K.B. (2003): Support Vector Machines for Credit Scoring: Comparing to and Combining with some Traditional Classification Methods, in: Schader, M., Gaul, W. and Vichi, M. (Eds.): *Between Data Science and Applied Data Analysis*, Springer, Berlin, 604-612

A square law for power of positions in a network[1]

Herman Monsuur

Royal Netherlands Naval College, P.O. Box 10.000, 1780 CA Den Helder, The Netherlands, e-mail: H.Monsuur@kim.nl

Abstract. We present a model for the pressure between positions in a network. This notion is made operational through identifying the locations and sources of network value. Precisely these 'markets', where network value is obtained and generated, can be at stake and determine the pressure a position may experience. Using a limit approach, we also derive our square law of network power.

1. Introduction

We present a model for the notion of structural tension between pairs of positions in a value generating network. In a network, the value or worth of a position cannot be determined autonomously or exogenously. A position with a large number of links may derive more value from the network than a peripheral position can ever claim. Not only does this network value depend on the number of relations to other positions, it also depends on the network values of these positions: if a position somehow succeeds in gaining extra value, this also adds to the values of neighboring positions. Generally speaking, network value is the advantage positions have because of their connections within the network.

We assume that each pair of positions is aware of this mutual dependence, also through changes in network values of their common neighbors. As they 'feel' the effects of each other's moves, they monitor the other‘s network situation and are prone to react. We propose measures for this latent potential to engage in rivalrous behavior. Our measures are derived from the notion of competitive pressure between firms operating in the same industry, offering similar products, see for example (Chen 1996). This competitive pressure that a focal firm experiences from a competitor is determined by two factors: the strategic importance of each of the product markets the focal firm shares with the competitor, and that competitor's market share in these markets. In our model, markets and market shares are replaced with positions and network values. We make the notion of pressure in a network operational through identifying the locations and sources of network value for each position. Precisely these 'markets', where network value is ob-

[1] Acknowledgment. I like to thank Prof. Robert Delver for several insightful comments regarding the notion of competitive pressure between firms.

tained and generated, can be at stake and determine the pressure a position may experience from other positions through their common neighbors. Pressures or threats that focal positions experience from competitors, and vice versa, are measures for the structural tension between pairs of positions in the network.

We show that network power of positions, which is defined as pressure that is experienced by other positions, does depend on network value. We also show that the overall pressure that a position experiences can be reduced by creating new links. For one specific pressure-map, it is even true that a position with twice as much network value as another position, has four times as much power: our square law of network power.

In Sect. 2, we determine the network value for each position in a network. Pressure-maps are explained and computed in Sect. 3. Finally, in Sect. 4, we introduce a square law for network power.

2. Network value

Values of positions are recursively related to the values of positions to which they are linked. To determine these network values, we have to solve this circular dependence of the value of a position on the values of adjacent positions. A well-known and often used method is the eigenvector approach: take the eigenvector of the adjacency matrix, corresponding to the principal (largest) eigenvalue. See (Bonacich 1972; Wasserman and Faust 1994; Laslier 1997; Monsuur and Storcken 2002, 2004).

Let $G = (V,E)$ be a undirected network, where V is a finite set of positions and E a set of links between pairs of positions. We assume that the network is connected, meaning that each pair of positions can be (indirectly) linked by a finite sequence of links from E. Let A be the adjacency matrix, defined by $A_{\mathrm{ij}} = 1$ if positions i and j are linked, and $A_{\mathrm{ij}} = 0$ if there is no (direct) link. We take $A_{\mathrm{ii}} = 1$ for all i, meaning that we assume that each position is adjacent to itself. Let $\underline{v}$ be the normalized eigenvector of A corresponding to the principal eigenvalue λ: $\underline{v} = \lim_{n\to\infty}\left(\frac{A^n}{\underline{e}^t A^n \underline{e}}\right)\underline{e}$. The components of $\underline{v}$ sum to one. Any nonnegative eigenvector of A is a multiple of $\underline{v}$. Then we let the network value c_i of a position i be its component of this vector $\underline{v}$. This means that the network value of a position is proportional to the sum of network values of adjacent positions.

For the graph of Fig. 1, the normalized eigenvector of the corresponding adjacency matrix is given by
[.0312, .0752, .0312, .119, .0939, .1074, .165, .1178, .0684, .1043, .0433, .0433] with eigenvalue 3.411238. The sum of all network values equals 1. For example, $c_g = 0.165$, meaning that 16.5% of the total network value generated by the (struc-

ture of the) graph is accumulated in position *g*. In social network analysis, this number can be interpreted as the status of position *g* within the network.

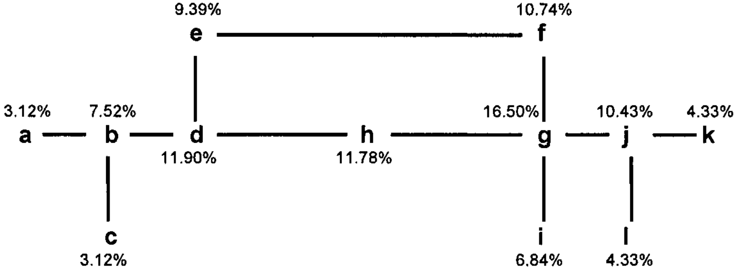


Fig. 1. A network and network values

To illustrate the mutual and recursive dependence, note that $A\underline{v} = \lambda\underline{v}$, showing that relative values of positions do not change if we assign to each position the sum of values of positions it is linked to. For the network of Fig.1 , we for example have, $\lambda c_g = c_f + c_g + c_h + c_i + c_j$. For later purposes, we also divide the eigenvector by λ, giving a vector *s* with components $s_x = c_x / \lambda$ and sum of components 1/λ.

[.0091, .022, .0091, .0349, .0275, .0315, .0484, .0345, .0201, .0306, .0127, .0127].

The network value of a position in a given network depends on the network values of positions it is linked to. In Fig.1, we for example have $c_g = s_f + s_g + s_h + s_i + s_j$. So position *f* contributes 0.0315/0.165×100% = 19.1% to the network value of *g*, while positions *g*, *h*, *i* and *j* contribute 29.3%, 20.9%, 12.2% and 18.5% respectively.

In the following section, we show how the network values can be used to determine the structural tension between positions.

3. Pressure-maps for networks

To introduce the notion of pressure within a network, we first consider the notion of competitive pressure between firms operating in the same industry. As mentioned before, competitive pressure is determined by two factors: the strategic importance of each of the markets the focal firm *b* shares with the competitor *a*, and that competitor's market share in these markets. For example, (d'Aveni 2001) uses the formula

$$p_{ab} = \sum_{m} (S_{am} / S_m) \times (S_{bm} / T_b)$$

to compute the magnitude of pressure felt by focal firm b from its overlaps with competitor *a*. The sum is over all product markets or segments *m*. In this formula, S_{am} is the unit sales of firm *a* in product market *m*; S_m is the total unit sales of all

competitors and the focal firm in the product market *m*; S_{bm} is the unit sales of firm *b* in product market *m*; T_b is the total unit sales of firm *b* across all its product markets. The second part of each term is the strategic importance of the invaded market *m* to focal firm *b*, while the first part is a measure of the size of the presence of the competitor in that market *m*, the market share of the competitor. All these pressures together, for each firm *a* and *b*, give a pressure-map. If p_{ab} is relatively large, then firm *a* is one of *b*'s top competitors, and an attack by firm *a*, such as introducing a new product, would be viewed as threatening by firm *b*.
Our networks do not have product markets where positions interact. But there is some analogy: take for example two positions, say *e* and *g* of Fig.1 with a common neighbor *f*. First, note that the network value of position *g* depends partly on the value of position *f*. So, to position *g*, the position *f* may be seen as a 'market' where network value is generated and accumulated: as computed above, the importance of position *f* to *g* is 19.1%. Secondly, position *e* contributes to the network value of position *f* in the sense that it has a 'share' of 0.0275/0.1074×100% = 25.6% in 'market' *f*: the presence of *e* in position *f* or the extent to which *e* is a key player in this market. To compute the pressure of *e* on *g* through position *f*, we may multiply the size of the presence and the importance to obtain 0.049. In general, we have:

Definition Given a network and two positions *a* and *b*, we let the pressure of position *a* on *b* be equal to

$$p_{ab} = \sum_{m \in N[a] \cap N[b]} \frac{c_a/\lambda}{c_m} \times \frac{c_m/\lambda}{c_b},$$

where, for any position *j*, c_j is the network value of position *j*, and N[*j*] is the set of positions adjacent to *j*, including *j* itself.

The first part of each term is the presence of the competitor *a* in the common neighbor *m*, the second term is the importance of the network value of *m* to the focal position *b*. For example, positions *f* and *g* have two common neighbors (*f* and *g* themselves). Therefore, the pressure of position *f* on *g* is equal to (.0315/.1074) × (.0315/.165) + (.0315/.165) × (.0484/.1650) = 0.112. Conversely, the pressure of *g* on *f* is equal to (0.0484/0.1074) × (0.0315/0.1074) + (0.0484/0.165) × (0.0484/0.1074) = 0.264. These number indicate the structural tension between the two positions.

Theorem 1 Let p_{ab} be the pressure that position *a* exerts on position *b*. Then

$$p_{ab} = \frac{\#\{N[a] \cap N[b]\}}{\lambda^2} \frac{c_a}{c_b},$$

where λ is the principal eigenvalue of the adjacency matrix.

Proof. By definition, we have $p_{ab} = \sum_{m \in N[a] \cap N[b]} \frac{c_a/\lambda}{c_m} \frac{c_m/\lambda}{c_b}$. This is equal to

$$p_{ab} = \frac{1}{\lambda^2} \sum_{m \in N[a] \cap N[b]} \frac{c_a}{c_m} \frac{c_m}{c_b} = \frac{\#\{N[a] \cap N[b]\}}{\lambda^2} \frac{c_a}{c_b}. \blacklozenge$$

In literature, one may find several studies concerning the relation between the principal eigenvalue λ and topological features of a network like density of links. For example, (Wang et al. 2004), study epidemic thresholds, which is a critical state beyond which infections become endemic. They present a model showing that this threshold equals 1/λ. In theorem 1, we showed that pressure of position *a* on *b* is equal to the quotient of their network values, adjusted by a factor $\frac{\#\{N[a] \cap N[b]\}}{\lambda^2}$. We call λ^2 the global damping factor. If λ is large, pressure is relatively small and tension between positions is small. This, for example, is the case if the network is dense. In these networks, positions have many connections through which they can generate network value. The term $\#\{N[a] \cap N[b]\}$, which is the number of common neighbors, indicating local interactions between *a* and *b*, we call a local expanding factor.

For the network of Fig.1, collecting all pressures p_{xy}, $x,y \in \{a,b,\ldots,l\}$, we obtain the following pressure-map in matrix representation:

$$\begin{pmatrix}
.172 & .071 & .086 & .023 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
.414 & .344 & .414 & .108 & .069 & 0 & 0 & .055 & 0 & 0 & 0 & 0 \\
.086 & .071 & .172 & .023 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
.328 & .272 & .328 & .344 & .218 & .095 & .062 & .174 & 0 & 0 & 0 & 0 \\
0 & .107 & 0 & .136 & .258 & .150 & .049 & .068 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & .078 & .196 & .258 & .112 & .078 & .135 & .088 & 0 & 0 \\
0 & 0 & 0 & .119 & .151 & .264 & .429 & .241 & .414 & .272 & .327 & .327 \\
0 & .135 & 0 & .170 & .108 & .094 & .123 & .258 & .148 & .097 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & .055 & .071 & .050 & .172 & .056 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & .083 & .109 & .076 & .131 & .344 & .414 & .414 \\
0 & 0 & 0 & 0 & 0 & 0 & .023 & 0 & 0 & .071 & .174 & .086 \\
0 & 0 & 0 & 0 & 0 & 0 & .023 & 0 & 0 & .071 & .086 & .174
\end{pmatrix}$$

The term $\#\{N[a] \cap N[b]\}$ is equal to A^2_{ab}, which is the entry on the location (a,b) of the matrix A^2. As the vector of network values also is an eigenvector of the matrix A^2, with eigenvalue λ^2, we obtain for fixed b: $\sum_a \#\{\mathrm{N}[a] \cap \mathrm{N}[b]\} c_a = \lambda^2 c_b$,

implying that $\sum_a \frac{\#\{N[a] \cap N[b]\}}{\lambda^2} \frac{c_a}{c_b} = 1$. Therefore, we have

Corollary 1 For each position b: $\sum_a p_{ab} = 1$. ♦

This shows that columns of the pressure-map sum to one. Therefore, the sum of pressures on b from other positions is equal to $1 - p_{bb}$. This number may be interpreted as a measure, say T_b, of the overall tension that a position experiences. As, by definition, $p_{bb} = \frac{\#\{N[b]\}}{\lambda^2}$, we have $T_b = 1 - \frac{\#\{N[b]\}}{\lambda^2}$. Therefore, the overall tension can be reduced by creating a new link to a position: $N[b]$ increases by 1, while λ does not increase substantial.

Corollary 2 For each pair of positions a and b with $\mathrm{N}[a] \cap \mathrm{N}[b] \neq \varnothing$, we have

$p_{ab} > p_{ba}$ if, and only if, $c_a > c_b$. ♦

Note that, in general, the tension that a experiences from the presence of b is not equal to the tension that b is facing from the presence of a.

Next, let a and b be a pair of positions. Suppose that a covers b. By that we mean that all positions adjacent to b are also adjacent to a and that, in addition, position a has at least one extra neighbor: $\mathrm{N}[b] \backslash \{b\}$ is strictly contained in $\mathrm{N}[a]$. In (Monsuur and Storcken 2004) it is shown that if a covers b, we have $c_a > c_b$. Therefore, we have

Corollary 3 If position b is covered by position a, then $p_{ab} > p_{ba}$. ♦

4. A square law for network power

As $\#\{\mathrm{N}[a] \cap \mathrm{N}[b]\} = A^2_{ab}$, we may introduce a whole family of pressure-maps:

$$p^1_{ab} = \frac{A_{ab}}{\lambda} \frac{c_a}{c_b}$$

$p^2_{ab} = \frac{A^2{}_{ab}}{\lambda^2} \frac{c_a}{c_b}$, the pressure-map we introduced in the previous section,

$$p^k{}_{ab} = \frac{A^k{}_{ab}}{\lambda^k}\frac{c_a}{c_b}, \text{ for arbitrary integers } k,$$

where $A^k{}_{ab}$ is the number of paths between positions a and b of length k. Using these pressure-maps, the pressure of a on b is equal to the quotient of their network value with adjustment factor $\frac{A^k{}_{ab}}{\lambda^k}$. So, the pressure and dependence is experienced not only through common neighbors, but also via chains of k links, $k \geq 2$. Note that, for arbitrary integer k and position b, $\sum_a p^k{}_{ab} = 1$.

Theorem 2 The pressure-map $p^{\propto}{}_{ab}$ is independent of b and is equal to $\frac{c_a{}^2}{c_a{}^2 + c_b{}^2 + \ldots + c_t{}^2}$.

Proof. Let $L := \lim_{i \to \infty} L_i$, where $L_i = \left(\frac{A}{\lambda}\right)^i$. It is well known that columns of L are positive multiples of the eigenvector c, see for example (Berman and Plemmons 1979). Moreover, as L_i is symmetric for each i, we know that L is symmetric. Altogether, this means that

$$L = \begin{pmatrix} kc_a & kc_b & kc_c & \ldots & kc_t \\ kc_b & k\frac{c_b}{c_a}c_b & k\frac{c_c}{c_a}c_b & \ldots & k\frac{c_t}{c_a}c_b \\ kc_c & k\frac{c_b}{c_a}c_c & k\frac{c_c}{c_a}c_c & \ldots & k\frac{c_t}{c_a}c_c \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ kc_t & k\frac{c_b}{c_a}c_t & k\frac{c_c}{c_a}c_t & \ldots & k\frac{c_t}{c_a}c_t \end{pmatrix},$$

where t is the number of positions and k some positive number. Observe that for each i, we have $L_ic - c = 0$, implying that also $Lc - c = 0$. This may be used to compute the value of k:

$$k = \frac{1}{c_a{}^2 + c_b{}^2 + \ldots + c_t{}^2}c_a.$$

Henceforth, we have

$$L = \frac{1}{c_a^2 + c_b^2 + \ldots + c_t^2}\begin{pmatrix} c_a^2 & c_b c_a & c_c c_a & \cdots & c_t c_a \\ c_a c_b & c_b^2 & c_c c_b & \cdots & c_t c_b \\ c_a c_c & c_b c_c & c_c^2 & \cdots & c_t c_c \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ c_a c_t & c_b c_t & c_c c_t & \cdots & c_t^2 \end{pmatrix}.$$

As $p^{\infty}{}_{ab} = \left(\lim_{i\to\infty} \frac{A^i{}_{ab}}{\lambda^i}\right)\frac{c_a}{c_b} = \left(\lim_{i\to\infty}\left(\frac{A}{\lambda}\right)^i\right)_{ab}\frac{c_a}{c_b}$, we have $p^{\infty}{}_{ab} = L_{ab}\frac{c_a}{c_b} = \frac{c_a c_b}{c_a^2 + c_b^2 + \ldots + c_t^2}\frac{c_a}{c_b} = \frac{c_a^2}{c_a^2 + c_b^2 + \ldots + c_t^2}$. This clearly is independent of position b.♦

This means that the pressure of a on b, using the pressure-map $p^{\infty}{}_{ab}$, is equal to the relative contribution of a to the total sum of squared network values of the network.

The pressure-map p^{∞} for the network of Fig. 1 is equal to:

[.009, .055, .009, .138, .086, .112, .265, .136, .046, .106, .018, .018]

From the theorem above, it is clear that we have a square law of network power:

Theorem 3 If the network value of position a is twice the network value of position b, then the pressure that position a can exert is four times as large as the pressure that b can exert, if we use the pressure-map p^{∞}. ♦

References

Berman A, Plemmons RJ (1979) Nonnegative matrices in the mathematical sciences. Academid Press, New York

Bonacich P (1972) Factoring and weighting approaches to status scores and clique identification. J. Math. Sociology 2: 113-120

Chen M-J (1996) Competitor analysis and interfirm rivalry: toward a theoretical integration. Academy of Management Review 21: 100-134

D'Aveni R (2001) Strategic supremacy: How industry leaders create growth, wealth, and power through spheres of influence. New York: The Free Press

Laslier J-F (1997) Tournament solutions and majority voting. Studies in Economic Theory 7. Springer

Monsuur H, Storcken T (2002) Centrality orderings in social networks. In Borm P, Peters H (eds) Chapters in game theory. Theory and Decision Library, Series C. Kluwer Academic Publishers, pp 157-181

Monsuur H, Storcken T (2004) Centers in connected undirected graphs: an axiomatic approach. Operations Research 52: 54-64

Wang Y, Chakrabarti D, Wang C, Faloutsos C (2004) Epidemic spreading in real networks: an eigenvalue viewpoint. Carnegie Mellon University, Mimeo.

Wasserman S, Faust K (1994). Social network analysis: methods and applications. Cambridge University Press, Cambridge.

Automated business diagnosis in the OLAP context

Emiel Caron, Hennie Daniels

[1]Erasmus University Rotterdam, ERIM Institute of Advanced Management Studies, PO Box 90153, 3000 DR Rotterdam, The Netherlands, phone +31 010 4082574, e-mail: ecaron@fbk.eur.nl; [2]Tilburg University, CentER for Economic Research, Tilburg, The Netherlands

Abstract. In this paper, we describe an extension of the OLAP (On-Line Analytical Processing) framework with automated causal diagnosis, offering the possibility to automatically generate explanations and diagnostics to support business decision tasks. This functionality can be provided by extending the conventional OLAP system with an explanation formalism, which mimics the work of business decision makers in diagnostic processes. The central goal of this paper is the identification of specific knowledge structures and reasoning methods required to construct computerized explanations from multidimensional data and business mo-dels. The methodology was tested on a case study involving the comparison of financial results of a firm's business units.

Keywords: Datamining, OLAP, Explanation, Business Intelligence.

1. Introduction

Today's OLAP (On-Line Analytical Processing) systems have limited explanation or diagnosis capabilities. The diagnostic process is now carried out manually by analysts, where the analyst explores the multidimensional data to spot exceptions, and navigate the data with operators like drill-down, roll-up, and selection to find the reasons for these exceptions. Such functionality can be provided by extending the conventional OLAP system with an explanation formalism. Here diagnosis is defined as finding the best explanation of unexpected behaviour (or symptoms) of a system under study [10]. This definition assumes that we know which behaviour we may expect from a correctly working system, otherwise we would not be able to determine whether the actual behaviour is what we expect it or not. The expected behaviour in an OLAP environment can be derived from some statistical model or can be domain knowledge from analysts. The objective of this paper is to extend the OLAP system with a diagnostic process. In short, we automate the current user-driven analysis of OLAP data, with an explanation formalism that finds exceptions, and finds out why these exceptions have emerged.

Our exposition on causal explanation is largely based on Feelders and Daniels' notion of explanations [2, 3], which is essentially based on Humpreys' notion of aleatory explanations [6] and the theory of explaining differences by Hesslow [4]. Causal influences can appear in two forms: contributing and counteracting. Therefore, the following canonical form for causal explanations is proposed in [2, 3]:

$$\langle a, F, r \rangle \text{ because } C^{+}, \text{ despite } C^{-}. \tag{1}$$

where $\langle a, F, r \rangle$ is the event to be explained, C^{+} is non-empty set of contributing causes, and C^{-} a (possibly empty) set of counteracting causes. The explanation itself consists of the causes to which C^{+} jointly refers. C^{-} is not part of the explanation, but gives a clearer notion of how the members of C^{+} actually brought about E. The explanandum is a three-place relation between an object *a* (e.g. the ABC-company), a property *F* (e.g. having a low profit) and a reference class *r* (e.g. other companies in the same branch or industry). The task is not to explain why *a* has property *F*, but rather to explain why *a* has property *F when the members of r do not.* This general formalism for explanation constitutes the basis of the framework for diagnosis in an OLAP context developed in this paper.

To position this paper we mention some related work regarding the explanation of differences in multidimensional data. In [8] Sarawagi presented an operator for data cubes that lets the analyst get summarized reasons for drops or increases observed at an aggregated level. In [9] the authors developed a discovery-driven exploration paradigm that mines the data for exceptions and summarizes the exceptions at appropriate levels in advance. The discovery-driven method is guided by pre-computed indicators of exceptions at various levels of detail in the cube.

The remainder of this paper is organized as follows. Section 2 introduces our notation for the multidimensional model, followed by a description of normative models appropriate for diagnosis. In section 3 the explanation formalism is extended for multidimensional data in order to automatically generate explanations for symptoms derived from multidimensional data. Finally, the complete method is illustrated in a case study on sales data in section 4.

2. Notation, equations and normative models

Many different notations of OLAP concepts are found in the literature. Here we introduce a generic notation that is particular suitable for combining the OLAP concepts of measures, dimensions, and dimension hierarchies. A measure is defined as a function on multiple domains: $y^{i_1 i_2 \ldots i_n} : D_1^{i_1} \times D_2^{i_2} \times \ldots \times D_n^{i_n} \rightarrow \square$. Each domain D_i has a number of hierarchies ordered by $D_i^{i_{\max}} \prec D_i^{i_{\max}-1} \prec \ldots \prec D_i^0$, where D_i^0 is the highest level and $D_i^{i_{\max}}$ is the lowest level in D_i . A dimension's top level has a single level instance $D_k^0 = \{\text{All}\}$. For example, for the time dimension we could have the following hierarchy $\mathrm{T}^2 \prec \mathrm{T}^1 \prec \mathrm{T}^0$, where $\mathrm{T}^0 = \{\text{All-T}\}$, $\mathrm{T}^1 = \{2000{,}2001\}$, and $\mathrm{T}^2 = \{\text{Q1,Q2,Q3,Q4}\}$. Sometimes we will write T[Quar- ter] for T^2. A *cell* is $(d_1, d_2, \ldots, d_n)$ where d_i are elements of the domain hierarchy at some level, so for example (2000,Amsterdam,Beer) is a cell. Each cell contains data, which are the values of the measures *y* like, for example, $\text{sales}^{122}(2000,\ \text{Amsterdam}, \text{Beer})$. And if no confusion can arise we will leave out the upper indices indicating level hierarchies and write sales(2000,Amsterdam, Beer). By applying suitable equations we can alter the level of detail and map low level cubes to high level cubes and vice versa. Here we investigate the common situation where the aggregation relation is the summarization of measures in the dimension hierarchy. So *y* is an *additive measure* if in each dimension and hierarchy level of the data cube:

$$y^{i_1 \ldots i_{q-1} \ldots i_n}(\ldots, a, \ldots) = \sum_{j=1}^{m} y^{i_1 \ldots i_q \ldots i_n}(\ldots, a_j, \ldots), \tag{2}$$

where $a \in D_k^{q-1}$, $a_j \in D_k^q$, q is some level in the dimension hierarchy, and m represents the number of level instances in D_k^q. Sometimes we will use the term $y^{i_1 \ldots i_{q-1} \ldots i_n}(\ldots,+,\ldots)$ for the left-hand side of the above equation. An example equation is:

$$\text{sales}^{102}(2000, \text{All-Locations}, \text{Beer}) = \sum_{j=1}^{4} \sum_{k=1}^{20} \text{sales}^{212}(2000.\text{Q}_j, \text{Country}_k, \text{Beer}).$$

Furthermore, we assume that a business is given by a business model *M* representing relations between measures. These relations can be derived from many domains, like finance, accounting, logistics, and so forth. Relations are denoted by

$$y^{i_1 i_2 \ldots i_n}(d_1, d_2, \ldots, d_n) = f(\mathbf{x}^{i_1 i_2 \ldots i_n}(d_1, d_2, \ldots, d_n)), \tag{3}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$, and $y, x_1, \ldots, x_n$ are measures defined on the same domains. Business model equations hold on equal aggregation levels in the data cube, therefore we may leave out upper indices. In Table 1, an example of a business model from a sales database is given.

Table 1. Example business model *M*

1. Gross Profit = Revenues - Cost of Goods
2. Revenues = Volume · Unit Price
3. Cost of Goods = Variable Cost + Indirect Cost
4. Variable Cost = Volume · Unit Cost
5. Indirect Cost = 30% · Variable Cost

The normative model specifies the reference class that should be used to compare. It also specifies the measures with respect to which the comparison should be made. There are many ways to construct reference objects for multidimensional data. The simplest way is pairwise comparison [8], where a value of a measure *y* is compared with another in the data cube, the reference variable norm(*y*). For example, we can compare sales(2000,Germany,All-Products) with the sales of the previous year, norm(sales(1999,Germany,All-Products)), as an historical norm value. Other common norm values are the average or expected value of a cell. We use the notation: $\bar{y}(\ldots,+,\ldots) = (\sum_{j=1}^{M} y(\ldots,a_j,\ldots))/M$. And for the average over all domains we write $\bar{y}(+,+,\ldots,+)$. Expected values are based on statistical models. A huge variety of statistical models exists; see Tukey in [5] for an elaboration. Here we will use *simple additive model*, where we assume that the joint contribution of the aggregates is the sum of the separate contributions from each aggregate and $\varepsilon(d_1,d_2) \square N(0,\sigma^2)$. For a multidimensional data set with only two dimensions we write the estimate for the expected value as an additive function of three terms obtained from the possible aggregates of the table:

$$\hat{y}(d_1,d_2) = \bar{y}(d_1,+) + \bar{y}(+,d_2) - \bar{y}(+,+). \tag{4}$$

Now the expression, $\partial y = q$ ($q \in \{\text{high,low}\}$) specifies an event in the data cube, i.e. the occurrence of a quantitative difference between the actual and norm value, denoted by ∂y. If for the norm value the expected value is used, we can scale the residual with the standard deviation $\sigma(y)$. In that case, a cell is a symptom or surprise value [9] if $(y - norm(\hat{y}))/\sigma$ is higher than some threshold δ.

3. Methodology

Explanations of events are usually based on general laws expressing relations between events, such as cause effect relations. In the data cube, two types of relations are available for explanation generation namely: multiple additive relations – according to (2) – in the dimension hierarchies, and the business model relations – according to (3) – between the measures. We are interested in explaining the difference between object *a* and *r*. Contributing and counteracting causes that explain ∂y are determined by the calculation of a *measure of influence* [3, 7]. The correct interpretation of the measure depends on the form of the function *f*; the function has to satisfy the so-called *conjunctiveness constraint* [2, 3].

First we discuss the situation where explanation is sustained by relations in the business model, after that we elaborate on the situation where explanation is sustained by dimension hierarchies. The definition of the measure of influence is more general for business model relations because here explanation is sustained by additive or monotonic relations:

$$\inf(x_i, y) = f(\text{norm}(\mathbf{x}_{-i}), x_i) - \text{norm}(y) \tag{5}$$

where $f(\text{norm}(\mathbf{x}_{-i}), x_i)$ denotes the value of $f(\mathbf{x})$ with all measures evaluated at their norm values, except the measure x_i. And where *i* is the index of the vector $\mathbf{x}$ and all measures are on the same aggregation level. In words, $\inf(x_i, y)$ indicates what the difference between the actual and norm value of *y* would have been if only the measure x_i would have deviated from its norm value. In the dimension hierarchy the function *f* is additive by definition, therefore we define the measure of influence for an additive measure as:

$$\inf(y^{i_q}(\ldots, a_j, \ldots), y^{i_{q-1}}(\ldots, a, \ldots)) = y^{i_q}(\ldots, a_j, \ldots) - \text{norm}(y^{i_q}(\ldots, a_j, \ldots)) \tag{6}$$

When explanation is sustained by a business model equation the set of contributing (counteracting) causes C^+ (C^-) consists of measures x_i of $\mathbf{x}$ out of the business model with: $\inf(x_i, y) \times \Delta y > 0 \;\; (< 0)$. In words, the contributing causes are those variables whose influence values have the same sign as ∂y, and the counteracting causes are those variables whose influence values have the opposite sign. And if explanation is sustained by the dimension hierarchy, the set of contributing (counteracting) causes C^+ (C^-) consists of the set of child instances a_j of dimension level i_q out of the hierarchy of a specific dimension with $\inf(y^{i_q}(\ldots, a_j, \ldots), y^{i_{q-1}}(\ldots, a, \ldots)) \times \Delta y > 0 \;\; (< 0)$.

Because every applicable equation gives an explanation, the number of generated explanations for a single symptom can be quite large. Especially when explanations are chained together to form a tree of explanations we might obtain intractable branches. In order to leave insignificant influences out of the explanation we elaborate on two generic concepts. Firstly, small influences are left out in the explanation by a filter. In [2, 3] the set of causes is reduced to the so-called parsimonious set of causes. The parsimonious set of contributing causes C_p^+ is the smallest subset of the set of contributing causes, such that its influence on *y* exceeds a particular fraction (T^+) of the influence of the complete set. The fraction T^+ is a number between 0 and 1, and will typically 0.8 or so. The second way to reduce the number of explanations is the introduction of a *measure of specificity* for each applicable equation. This measure quantifies the "interestingness" of the explanation step. The measure is defined as:

$$\text{specificity} = \frac{\#\ \text{possible causes}}{\#\ \text{actual causes}}. \tag{7}$$

The number of possible causes is the number of right-hand side elements of each applicable equation, and the number of actual causes is the number of elements in the parsimonious set of causes. We prefer explanation steps with a relatively high specificity value.

In the previous parts, we have discussed "one-level" explanations, because they are based on a single relation from the business model or dimension hierarchy. For diagnostic purposes, however, it is meaningful to continue an explanation of $\partial y = q$, by explaining ∂y of its contributing causes. In *multi-level explanation* this process is continued until a parsimonious contributing cause is encountered that cannot be explained further due to:

- the business model, because the business model equations do not contain an equation in which this contributing cause appears on the left-hand side.
- the dimensions hierarchies, because the hierarchies do not contain an additive measure in which this contributing cause appears on the left-hand side.

The result of this process is an *explanation tree of causes*, where y is the root of the tree with two types of children, corresponding to its parsimonious contributing and counteracting causes respectively.

4. Case study: sport equipment sales data

We use a dataset (called "GOSales") obtained from the Cognos OLAP product PowerPlay [1] as a case study for our method. The data consist of 42.063 records and three dimensions; see Fig. 1. The measures (and relations) of the sales data are defined in Table 1. In this case study we present examples of the explanation process namely: the multi-level explanation of a symptom in the product dimension, and explanation of a symptom in the business model M of the data cube.

Product	Location	Time
Product (115)	Name (88)	Month (36)
ProductType (21)	Position (70)	Quarter (12)
ProductLine (5)	City (28)	Year (3)
	Country (20)	

Fig. 1. Dimensions and hierarchies of the GOSales dataset. The numbers within brackets denote the cardinality of that level of the dimension.

Suppose that an analyst starts exploring the cube in the Year×Country×All-Products plane and problem identification yields the symptom S={∂profit(2001, Spain,All-Products)="*low*"}. Under the assumptions of the additive model we calculated the expected values for the context using (4) and standardized the residuals. Here we choose $\delta = 1.28$ corresponding to a probability of 90% in the normal distribution, and find that the standardized residual for Spain in the year 2001 is larger than the threshold value. We filter out insignificant influences from the explanations, therefore we take $T^{+} = T^{-} = 0.75$. Explanation generation may start in the product dimension for the detected symptom, where explanation is sustained by additive relations. First the decrease in profit on the All-Products level is examined on the ProductLine level of the dimension hierarchy. Therefore, profit^{110} (2001,Spain,All-Products) is the root of the explanation tree. The norm values for explanation generation are all based on the expected values for the entries of the dimension level ProductLine in the context Year×Country×ProductLine. Computation of the influences of the individual variables for the additive equation above with (6) yields the results in Table 2. From the data in the table it can be concluded that C_p^+ = {profit(.,.,Personal Accessories), profit(.,., Golf Equipment), profit(.,.,Mountaineering Equipment)}, since only these three relatively large causes are needed to explain the desired fraction of inf(C^+, profit(.,., All-products). Obviously, C_p^- = { }. The parsimonious causes are explained further on the level ProductType, the data for comparison of the entries on the level ProductType of the ProductLine Personal Accessories are presented in Table 3.

Table 2. Data for explanation of S = {∂profit(2001,Spain,All-Products) = "low"}

Profit(2001,Spain,All-Products)	Norm	Actual	Inf
Profit(.,.,All-Products)	242,169.03	145,976.67	
Profit(.,.,Camping Equipment)	6,488.07	-8,684.36	-15.172.43
Profit(.,.,Personal Accessories)	46,610.41	22,521.12	-24,089.29
Profit(.,.,Outdoor Protection)	17,807.01	10,033.18	-7,773.83
Profit(.,.,Golf Equipment)	99,048.87	79,928.64	-19,120.23
Profit(.,.,Mountaineering Equip.)	72,214.67	42,178.09	-30,036.58

Table 3. Data for explanation of S = {∂profit(.,.,Personal Accessories) = "low"}

Profit(2001,Spain,Personal Accessories)	Norm	Actual	Inf
Profit(.,.,Personal Accessories)	46,610.41	22,521.12	
Profit(.,.,PA.Watches)	24,409.07	13,345.47	-11,063.60
Profit(.,.,PA.Eyewear)	4,686.22	2,302.91	-2,383.31
Profit(.,.,PA.Knives)	6,664.91	2,535.36	-4,129.55
Profit(.,.,PA.Binoculars)	4,792.63	1,385.56	-3,407.07
Profit(.,.,PA.Navigation)	6,057.59	2,951.82	-3,105.77

From the data in Table 3 it follows that C_p^+ = {profit(.,.,Watches),profit(.,.,Knives),profit(.,.,Binoculars),profit(.,.,Navigation)} and C_p^- = { }. Here the relatively large cause Watches contributes significantly to the low profit for Personal Accessories, therefore explanation generation continues downwards in the Product level. The parsimonious causes (Knives, Binoculars, and Navigation) are presented on the aggregated level, to avoid too much detail for the analyst.

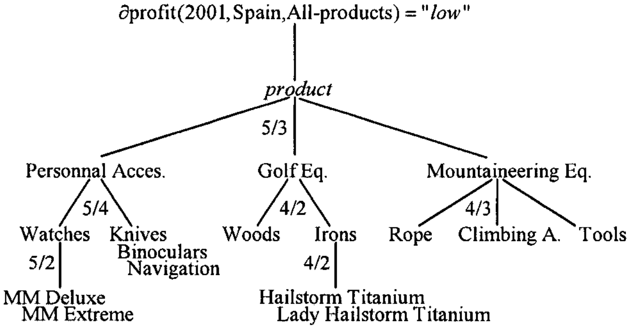


Fig. 2. Diagnosis S={∂profit(2001,Spain,All-P.)="low"} in the product dimension

Now the previous examples of one-level explanations are combined to a complete diagnosis in the product dimension. Fig. 2 summarizes the results of the multi-level diagnosis, where the lines indicate contributing causes and the numbers indicate the specificity value of the explanation step. The specificity values are determined using (7). Moreover, explanation trees can be constructed in the same way for the time and location dimension.

The symptom S={∂profit(2001,Spain,All-Products)="*low*"} can also be explained in the business model *M* of the data cube. Hence the corresponding equation in Table 1 is: $\text{profit}^{110} = \text{revenues}^{110} - \text{costs}^{110}$. The norm values for the measures revenues(.,.,.) and costs (.,.,.) are based on the context Year×Country×All-Products. The influences of the individual measures in the equation for profit are calculated by applying (5). From the data in Table 4 it follows that: profit(.,.,.)="*low*", because $C_p^+ = \{\text{revenues}(.,.,.)\}$, despite $C_p^- = \{\text{costs}(.,.,.)\}$. Explanation generation may continue for the contributing cause revenues, but is omitted here because of space limitations.

Table 4. Data for explanation of S = {∂profit(2001,Spain,All-Products) = "low"}

(2001,Spain,All-Products)	Norm	Actual	Inf
Profit(.,.,.)	242,169.02	145,976.67	
Revenues(.,.,.)	2,269,708.27	1,698,895.10	-570,813.17
Costs(.,.,.)	2,027,539.25	1,552,918.43	474,620.82

References

1. Cognos Incorporated, Cognos Series 7, Cognos PowerPlay, http://www.cognos.com, (2004).
2. Daniels HAM, Feelders AJ (2001) Theory and methodology: a general model for automated business diagnosis. European Journal of Operational Research, 130: 623-637
3. Feelders AJ (1993) Diagnostic reasoning and explanation in financial models of the firm. PhD thesis, University of Tilburg
4. Hesslow G (1984) Explaining differences and weighting causes. Theoria, 49:87-111
5. Hoaglin DC, Mosteller F, and Tukey JW (1988) Exploring data tables, trends and shapes. Wiley series in probability
6. Humphreys PW (1989) The chances of explanation. Princeton University Press, Princeton, New Jersey
7. Kosy DW, Wise BP (1984) Self-explanatory financial planning models. In: Proceedings of AAAI-84, Los Altos, CA, Morgan Kaufmann, pages 176-181
8. Sarawagi S (1999) Explaining differences in multidimensional aggregates. In: Proceedings of the 25th VLDB Conference, Edinburgh, Scotland, pages 42-53
9. Sarawagi S, Agrawal R, and Megiddo N (1998) Discovery-driven exploration of OLAP data cubes. In: Proceedings of the 6th Conference on EDBT, Valencia, Spain
10. Verkooijen W (1993) Automated financial diagnosis: a comparison with other diagnostic domains. Journal of Information Science, 19, pages 125-135

User-Oriented Filtering of Qualitative Data

Carsten Felden, Peter Chamoni

Universität Duisburg-Essen, Lotharstraße 65, 47048 Duisburg,

{c.felden/chamoni}@uni-duisburg.de

Abstract. Several business intelligence concepts show that it is possible to build a data warehouse based on heterogeneous quantitative and qualitative data. But the danger of flooding the management with information still remains. Business information should be made available according to the personal need of a manager by a self-defined push mechanism. The concept of active data warehousing is going to be expanded in this way, first. Messages about new relevant information should be created for quantitative and qualitative data as well. Due to this, there is a need for an enterprise specific ontology. This ontology works as an intermediate between current information and user profiles. Just interesting information are passed to the user. The second approach clusters qualitative data in favour of visualization. The usage of increasing cell structures provides a self organizing map. Decision makers will get the opportunity to search in document volumes, which are clustered.

1 Introduction

The information to decision makers is nowadays afflicted with the problem of information overload [14]. The World Wide Web (WWW) offers the opportunity, to integrate textual information (qualitative data) from enterprise external sources. This leads back to the problem of information overload.

In this paper we propose the enhancement of an existing market data information system (MAIS) with internal and external data by two methods in favour of text documents analysis. The first approach realizes the automation of individual information filtering in the sense of an active data warehouse to reduce the information overload. Due to this, the user profile plays an important role, because an individual filtering of information for decision makers can be made according to his profile. This enables decision makers to fulfil their tasks faster, because of the selection and individual representation of decision relevant information. The second approach supports the more efficient information search of a user by clustering text documents, to enable a more powerful retrieval. The results of our findings are summarized in a conclusion.

2 Conceptual Frameworks

There is an implicit need of fast decision-relevant information support. This leads to information systems which can recognize situations automatically and enforce user actions. Thus the data warehouse nowadays changes into an active data warehouse which provides a high availability of business-critical information.

2.1 Basic concept

There already exist approaches to integrate quantitative and qualitative data in a data warehouse. This has to be understood as a basis for the further development, whereas an active data warehouse should also refer to qualitative data [4]. The underlying information system was realized in a project of the VEW Energie GmbH and the University of Duisburg.

There are several approaches for the distinction of interesting and uninteresting internet documents to be mapped into a data warehouse. For this reason, non-overlapping classes are used, in order to make a clear separation. The examination of the documents and the setting up of classification criteria has to be differentiated. Alternative automatic procedures are statistic methods, e.g. the Rocchio algorithm, Support Vector Machine (SVM), or artificial neural networks [18]. In context of MAIS development, a neural network was used for relevance determination. The integration process is illustrated in the following figure 1.

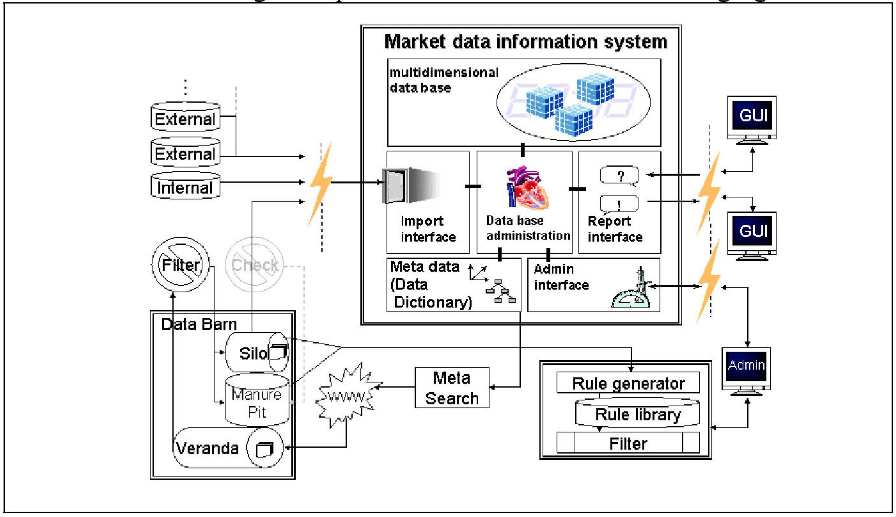


Fig. 1. Market data information system [7]

The set of dimensions of the implemented multidimensional data model are the basis of the retrieval process to find external data. If a coded retrieval query refers to a dimension itself, the result has to be linked to this dimension. If the coded query refers to the attributes of a dimension, the qualitative data have to be linked to the respective OLAP-slice [3]. This integration process is independent of any business scenario.

By using meta data from the data warehouse a web extracting and loading tool is applied in order to find and transfer information. All internet pages, which are found, will be stored in the so-called data barn. The data barn makes a transient and persistent storage for the processing of this pages available. A central problem for the integration is the extraction of information from the internet and the generation of rules for the classification of data. This classification is done by a Multilayer Perceptron [2]. The input of the MLP is an information surrogate which consists of a vector capturing a subset of meta data, quality demands and conceptual extension. The applied rules are stored in the rule library. Although an evaluation of the text documents is made by the neural network, all documents cannot be examined by the decision makers. Thus the individual supply with interesting information for managers moves into the focus of further activities. These transactions are accomplished by an active data warehouse automatically, whereby the reactive component is to be understood as an extension of the MAIS.

2.2 Active Data Warehouse

As described in the literature, an active data warehouse is a collection of data, which describes facts and business-relevant events. The user-specific behaviour is implemented in the data base. This database is active in the sense of supervising potentially interesting events. If a defined situation occurs, the system will implement appropriate actions. This data base behaviour can be specified by data base triggers in the form of an *event-condition-action-rule* (ECA) [5]. These triggers are currently implemented only for quantitative data. However, also qualitative data include information with high importance for decision makers. Due to this it is necessary, to expand the idea of an active data warehouse to qualitative data.

3 Analysing qualitative data

Founded on the above remarks we are going to develop a binary classification in this chapter. To do this, an ontology is developed and put into reason. The basic market data information system is not able to group similar documents, in order to attain further information to a certain topic area. An appropriate procedure to cluster the integrated documents is developed as well.

3.1 Ontology development

The development of an ontology should be based on a data dictionary. This ensures the term clarity within the data base and to individual data fields. It additionally contains meta data about the data base structures [7]. The indexing results of the integration process of the market data information system are considered additionally for the further approach. An ontology is defined as a formal and explicit specification of a hierarchically arranged conceptualization [8]. For creation of so called topic areas and rules, the concepts can be differentiated using the five market forces of competition according to Porter [16]. Beyond that, environmental referred concepts are present [20]. They have to be connected to the enterprise referred concepts. This task is accomplished by the Latent Semantic Analysis (LSA), on basis of the document collection, to determine term correlations in documents [9]. This ontology operates as an intermediate between users and documents in the database and is used for a user profile.

3.2 User profiling

The model of a user profile is based on the users' business role which is similar to a user-oriented filtering system. This model is built on a set of term vectors which can be used to map new text documents and generate recent and fitting information. The profiling can be differentiated into abstract and personal profiling [1].

Abstract profiling means that those terms are identified by documents of a set of training data which belong to a non-individualized class of people. This class represents e. g. the individual role of a person inside an organization. Such a role consists of at least one or more topic areas of the developed ontology [6]. Personal profiling means the individual collection of terms, which are rated as problem-characteristic by the decision maker. These terms can be arranged directly or determined automatically by selected documents.

Finally, the user has a list of all topics in his profile. Each topic requires a minimum similarity value additionally given by the user. This value is the starting signal for the operation of the ECA mechanism. Particularly important circumstances are described by rules, for which concepts are likewise selected. The concepts of a rule are also stored in a vector. A rule requires no minimum similarity value, because these rules must always be completely fulfilled.

It has to be decided whether the structure of the profile is accomplished explicitly by the user or implicitly by observation of the user behaviour. The individual user profiles are the result of the process implemented above.

3.3 Document analysis

The document analysis of the data warehouse consists of the computation of the individual relevance of the imported text documents. Due to this, a new document is examined for individual topic area vectors, first. Afterwards, the rules of the relations for each user profile are examined. Each vector of a topic area is compared with the index of a document, and the cosine coefficient is formed as follows [19].

$$\cos(w_i, q\ ) = \frac{\sum_{k=1}^{n} w_{i,k} q_k}{\sqrt{\sum_{k=1}^{n} w_{i,k}^2}\sqrt{\sum_{k=1}^{n} q_k^2}}$$

For each document D we provide a document index as a vector $d=(d_1, \ldots, d_t)$ of t terms representing the content of D. A profile P describes a user role and can also be stated as a term vector $p=(p_1, \ldots, p_n)$. The boolean vector $q=(q_1, \ldots, q_n) \in \{0, 1\}^n$ indicates the existence of any term p_j in a query Q ($q_j=1$ *if* $p_i \in Q$ else $q_i=0$). The boolean vector $w_i=(w_{i,1}, \ldots, w_{i,n}) \in \{0, 1\}^n$ indicates the coincidence of term p_j in a document D_i ($w_{i,j}=1$ if $p_j=d_k$ for any $k \in \{1, \ldots, t\}$ else $w_{i,j}=0$). The cosine measure gives the similarity between vector w_i and vector q. Due to the reason that the data warehouse documents are already examined by an artificial neural net, the cosine coefficient is sufficient, in order to determine topic area similarities.

The result of this partial analysis is the accordingly determined cosine values for the topic areas. Thus remain the rules, which have to be examined afterwards [17]. It is an obstacle that during the indexing all information about the descriptors' arrangement are lost. For this reason the indices are not used for the rules' examination. With the examination of a rule, three sequential sentences are considered. If the sentences are relatively short, the possibility exists that the interesting concepts are part of a following sentence. Beginning with the first three sets of a document it is examined whether the concepts are contained in the sentences or not. The sequence of the concepts within the sentences is unimportant. The first sentence is removed after the examination and the following sentence is included.

The examination is accomplished for all rules of a profile. All results as well as those of the topic area examination are stored in the data warehouse. The results contain the position in the document, at which the examination was successful. This is apart from the rule and the topic designation, too. A trigger is implemented for each profile after complete analysis of a document (Event). If one threshold value is exceeded at least or a rule fulfilled (condition), a user message with the appropriate information is dispatched (action).

If a user is informed about a relevant text document, the analysis can be supported by clustering text documents, so that the user can find context related information easier.

3.4 Clustering

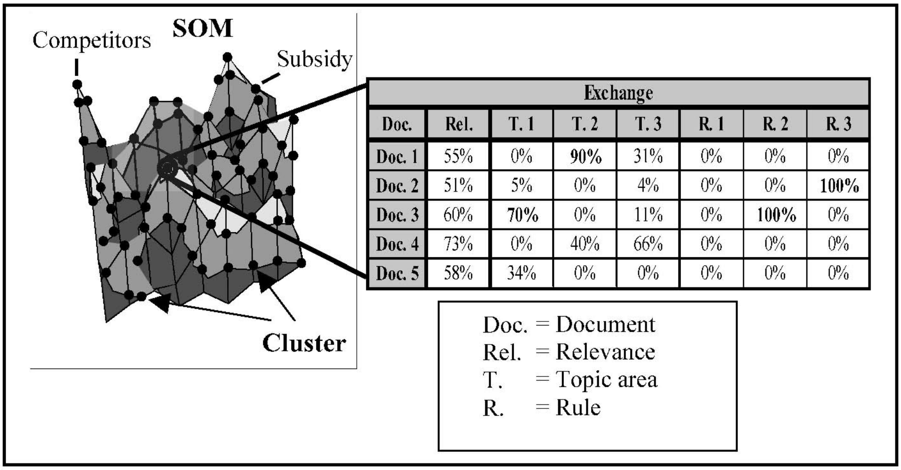
In order to offer a visual analysis of relevant documents, clustering can be done by a self organizing map (SOM), because this makes a visualization of the results possible during creation time [15].

A SOM is produced once by building increasing cell structures and renewed in defined time intervals [10]. First of all, the number of neuron entrances has to be fixed, which is determined by the number of provided concepts. There is a need to put these concepts into a firm order, so that a concept is always placed at the same entrance of the neuron. The weights g_{ij} are determined as follows [21]:

$$g_{ij} = \frac{freq_{il}}{\max_l freq_{lj}} \times \log \frac{D}{do_i}$$

$freq_{ij}$ is the frequency of a descriptor in the document, $max_l freq_{lj}$ the number of all descriptors in a document of j. Additionally, the number of all documents D has to be divided by do_i, which is the number of documents, in which the descriptor d_i arises. The absolute frequency, with which a descriptor arises in a document, is assigned to the appropriate concept. If a further descriptor is found in a document, the value is added to the already existing frequency. In order to avoid uncontrolled shifts, the weight of the individual dimensions is updated before map creation as well as in certain time intervals. The SOM is based on the Euclidean distance. The weighting of the individual concepts is considered to show content wise similarities.

A representative selection of documents is done according to the number of collected text document within the data warehouse. The net is developed and the shown procedure works until a defined abort criterion is valid (maximum number of neurons). These neurons represent a defined number of clusters for the map. If the net is completed, all neurons, which fall below a minimum value, are removed afterwards [10]. All documents of the data warehouse are finally assigned to the neurons of the provided map in accordance with the grade of similarity [13]. The result is a two-dimensional map, whereby the clusters are arranged in triangles [11]. The frequency of assigned documents is reflected by the height of a triangle. Figure 2 shows the clustering of documents by an example.



Exchange							
Doc.	Rel.	T. 1	T. 2	T. 3	R. 1	R. 2	R. 3
Doc. 1	55%	0%	**90%**	31%	0%	0%	0%
Doc. 2	51%	5%	0%	4%	0%	0%	**100%**
Doc. 3	60%	**70%**	0%	11%	0%	**100%**	0%
Doc. 4	73%	0%	40%	66%	0%	0%	0%
Doc. 5	58%	34%	0%	0%	0%	0%	0%

Doc. = Document
Rel. = Relevance
T. = Topic area
R. = Rule

Fig. 2. Self organizing map

The clusters are represented by black dots. Due to clarity, just five documents are indicated within the table. The table shows the relevance value of the neural network in favour of each document. The individually determined values of the topic areas and of the rules of the user profile are stored as well. The emphasized percentages fulfil the user criteria.

4 Conclusions

An approach for the accomplishment of the information flooding, with consideration of individual preferences and an approach for the representation of qualitative data were presented in this paper. All developments are based on the market data information system. Individual preferences of the users found consideration. The concept of an active data warehouse was selected for the automatic information supply of the decision makers. But the problem exists that the used triggers just process quantitative data, traditionally, because they have to examine data for certain criteria. Due to the reason that qualitative data are unstructured, no clear criteria could be intended for these. The usage of meta data, which are produced by text mining for each text document, helps to overcome this obstacle.

The individual information filtering is done by the automation of individual information filtering in the sense of an active data warehouse. These user profiles describe contents of the interesting qualitative data. A development of the user profiles by the users themselves is done by the structuring and visualization of concepts according to Porter's competition forces.

Problems arise in particular in case of the ontology development. First of all, the execution of the LSA needs a lot of computer resources for the formation of concepts. The necessary parameters are difficult to determine. The prototype revealed that a disadvantage of this approach is in the lack of interpreting the LSA results. Each word-word-relation contains also characteristics of all other dimensions. That leads to the problem that a clear reconstruction of the meaning of individual words is possible. Here it applies to examine further concepts like rule induction algorithms or explicit background algorithms [12] in place of the LSA for applicability.

But apart from this, reactive information supply is facilitated for the user by a clustering of all documents. This supports the search of further information. The representation of the document area is made by three-dimensional plots. Thereby the user can explore the document area independently and cover its additional need for information. Fast findings of similar information can create a solid basis for decision making. The shown approaches make a meaningful contribution for the accomplishment of the increasing data volume in the daily work of managers.

References

[1] Abramowicz, W.; Kalczyński, P.; Weçel, K.: Filtering the Web to Feed Data Warehouses. Springer, London et al. 2002.

[2] Bishop, C. M. (1995): Neural Networks for Pattern Recognition, Oxford, 1995.

[3] Codd, E.; Codd, S.; Salley, C.: Providing OLAP (On-line Analytical Processing) to User-Analysts. An IT Mandate. White Paper. Arbor Software Corporation. 1993.

[4] Colomb, R. M. (2002): Information Retrieval – The Architecture of Cyberspace, London, 2002

[5] Elmasri, R./Navathe, S. B. (2003): Fundamentals of database systems, 4 edition, München, 2003.

[6] Ellingsworth, M.; Sullivan, D.: Text Mining Improves Business Intelligence and Predictive Modeling in Insurance. In: DM Review 13 (2003) 7, S. 42 - 44.

[7] Felden, C.: Konzept zum Aufbau eines Marktdateninformationssystems für den Energiehandel auf der Basis interner und externer Daten. DUV, Wiesbaden 2002.

[8] Fensel, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer, Berlin, Heidelberg 2001.

[9] Foltz, P.; Kintsch, W.; Landauer, T.: The Measurement of Textual Coherence with Latent Semantic Analysis. Discourse Process. Http:// www.knowledge-technologies.com/papers/dp2.foltz.pdf, 1998, Abruf am 2003-12-15.

[10] Fritzke, B.: Wachsende Zellstrukturen. Ein selbstorganisierendes Neuronales Netzwerkmodell. Arbeitsberichte des Instituts für mathematische Maschinen und Datenverarbeitung (Informatik), 25. Bd., Nr. 9. Erlangen 1992.

[11] Fritzke, B.: Vektorbasierte Neuronale Netze. Habilitationsschrift. Shaker, Aachen 1998.

[12] Hotho, A.; Staab, St.; Stumme, G.: Wordnet improves Text Document Clustering. In: Proceedings of the Semantic Web Workshop at SIGIR-2003, 26th Annual International ACM SIGIR Conference, July 28-August 1, 2003, Toronto, Canada.

[13] Kamphusmann, T.: Text-Mining. Eine praktische Marktübersicht. Symposium, Düsseldorf 2002.

[14] Kohonen, T.: Overture. In (Seiffert, U.; Jain, L. Hrsg.): Self-Organizing Neural Networks. Recent Advances and Applications. Studies in Fuzziness and Soft Computing, Nr. 78. Physica, Heidelberg et al. 2002; S. 1 - 12.

[15] Kohonen, T.; Kaski, S.; Lagus, K.; Salojärvi, J.; Honkela, J.; Paatero, V.; Saarela, A.: Self organizing of a massive document collection. In: IEEE Transactions on Neural Networks, Vol. 11, No. 3, Mai 2000, S. 574 - 606.

[16] Porter, M.: Wettbewerbsstrategie (Competitive Strategy). Methoden zur Analyse von Branchen und Konkurrenten. 10. Aufl. Campus, Frankfurt et al., 1999.

[17] Pullwitt, D.: Integrating Contextual Information to Enhance SOM-based Text Document Clustering. Graduiertenkolleg Wissenspräsentation, Institut für Informatik, Universität Leipzig. Http://www.informatik.uni-leipzig.de/˜pullwitt/papers/nnsi.pdf, 2001, Abruf am 2003-12-13.

[18] Sebastiani, F.: Machine Learning in Automated Text Categorization. In: ACM Computing Surveys, Vol. 34, No. 1, März 2002, S. 1 - 47.

[19] Sullivan, D.: Document Warehousing and Text Mining. Techniques for Improving Business Operations, Marketing, and Sales. Wiley, New York et al. 2001.

[20] Welge, M.; Al-Laham, A.: Strategisches Management. Grundlagen - Prozess - Implementierung. 3. Aufl. Gabler, Wiesbaden 2001.

[21] Zavrel, J.: Neural Information Retrieval. An Experimental Study of Clustering and Browsing of Document Collections with Neural Networks. Http://ilk.kub.nl/~zavrel/zavrel.scriptie.ps.Z, 1995, Abruf am 2003-12-13.

Vorstellung einer erweiterbaren Selbstlernumgebung „Operations Research“ als Beispiel für exploratives, Web-basiertes E-Learning im Hochschulumfeld

Michael Lutz, Johannes Kern

Fachhochschule Augsburg, Fachbereich Informatik, Baumgartnerstr. 16, D-86161 Augsburg, (E-Mail: michael.lutz@fh-augsburg.de, mail@johannes-kern.de)

Zusammenfassung. Im Rahmen der High-Tech-Offensive Bayern (HTO) wurde an der Fachhochschule Augsburg für die Virtuelle Hochschule Bayern (VHB) eine Web-basierte „Selbstlernumgebung Operations Research (OR)“ entwickelt. Ziel war es, vornehmlich Studierenden der Fachrichtungen Wirtschaftsinformatik und Informatik, eine OR-Vorlesung virtuell bzw. online anzubieten, mit der sie sich die grundlegenden OR-Verfahren im Selbststudium aneignen können.

Das unter diesen Voraussetzungen entwickelte Lernprogramm wird seit gut einem Jahr intensiv von Studierenden verschiedener Hochschulen genutzt. Wesentliche Besonderheit dieses Lernprogramms ist der explorative Charakter:

Neben dem eigentlichen Lehrbuch, das im Wesentlichen aus einem Online-Skriptum besteht, gibt es eine Sammlung von über 60 Java-Applets. Diese simulieren die theoretisch vorgestellten Algorithmen anhand von frei wählbaren Werten, indem sie jeden Schritt erklären. In diesen Simulationen kann der Lernende schrittweise die Algorithmen an eigenen Beispielen selbst „entdecken“.

Das umgesetzte Konzept ermöglicht den Studenten auch einen iterativen Lernprozess und die Möglichkeit einer effizienten Wissensauffrischung. Die bisherigen Erfahrungen mit dem Lernprogramm und die durchgeführten Studentenevaluationen werden vorgestellt und grob analysiert.

Die Selbstlernumgebung im Gesamten entspricht den Vorstellungen der deutschen HRK (Hochschulrektorenkonferenz), die in Selbstlernumgebungen Lehrangebote sieht, die eigenständig und selbstbestimmend und praktisch ohne Betreuung von den Lernenden bearbeitet werden und die damit auf die Lehrbedürfnisse, Lerngeschwindigkeiten und das individuelle Zeitbudget Rücksicht nehmen.

Eine weitere Besonderheit ist die einfache Erweiterbarkeit dieses Web-basierten Lernprogramms. Die flache Struktur erlaubt es weitere Texte und Applets zu integrieren. Um eine einfache Erstellung von interaktiven Übungen zu ermöglichen, wurde ein Autorenwerkzeug entwickelt, das es ermöglicht, ohne jegliche Java Kenntnisse, Java-Übungs-Applets mit automatischer Auswertung zu erstellen. Zum Abschluss des Vortrags wird dieses Werkzeug, das mittlerweile auch in anderen E-Learning-Projekten eingesetzt wird, in seinen elementaren Funktionen vorgestellt.

1. Verwendete Technologien

Vor Beginn der Entwicklungstätigkeiten wurde beschlossen, das Lernprogramm auf einfachen, statischen HTML-Seiten basierend zu erstellen. Die Gründe, keine der teilweise sehr mächtigen E-Learning-Plattformen zu verwenden, lagen vor allem in der schwierigen Wartbarkeit (besonders nach Ausscheiden des Entwicklerteams, das überwiegend aus studentischen Hilfskräften bestand). Von Beginn an wurde daher die volle Aufmerksamkeit einem didaktisch gut durchdachten Lerninhalt und dazu passenden Visualisierungen gelegt.

Unter diesen Gesichtspunkten entstand eine einfach wartbare und leicht erweiterbare Lernumgebung. Diese besteht aus statischen HTML Seiten, in die eine große Anzahl von Plattform und Browser unabhängigen Java Applets integriert sind. Die Texte des Lernprogramms entstammen größtenteils dem seit vielen Jahren ausgereiften Skriptum zur Präsenzvorlesung „OR“. Die Java Applets mussten komplett neu entwickelt werden.

Die Wahl von Java als Basis zur Realisierung der weitergehenden E-Learning-Konzepte hat sich als sehr praktikabel erwiesen: Mit Ausnahme einiger weniger Problemfälle, lief die gesamte Lernumgebung problemfrei.

Die Anwender vermissten auch die weiterführenden Funktionen der angesprochenen E-Learning-Plattformen – wie dies schon vermutet wurde – nicht. Auf der Administrationsseite hat es sich bewährt, ein einfaches und statisches Web-Konzept zu verwenden.

2. Überblick über die Lernumgebung

Das Besondere dieser OR-Lernumgebung ist die Verknüpfung eines traditionellen Lehrbuches mit animierten, interaktiv zu bedienenden Beispielrechnern. Zudem kann jeder Benutzer die Algorithmen nach seinen Vorlieben erlernen:

- Klassisch durch das Lehrbuch.
- Lehrbuch kombiniert mit Beispielrechnern und Demoversionen.
- Ohne Lehrbuch - nur durch selbständiges „entdecken“ der Algorithmen (besonders attraktiv für die Wissensauffrischung).

Zusätzlich können die Beispielrechner und interaktiven Aufgaben auch zur Überprüfung des Lernfortschritts bzw. zum Auffinden der selbst gemachten Fehler genutzt werden.

Abbildung 1 zeigt das Navigationsmenü der Lernumgebung, über das u.a. die folgenden Bestandteile der Lernumgebung erreichbar sind:

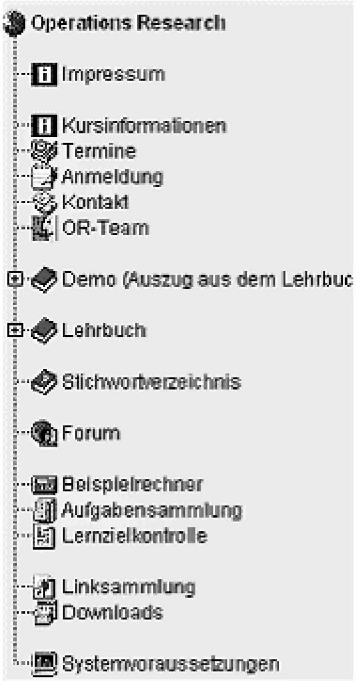


Abb.1 - Navigation

- **Kursinformation:** Sie gibt dem Lerner einen Leitfaden zur Bearbeitung des Stoffes. In einem Zeitplan wird vorgegeben, welche Texte und Rechner explorativ zu bearbeiten sind und welche Aufgaben der Lernzielkontrollen durchzuführen sind.

- **Aufgabensammlung:** Wesentlicher Bestandteil sind interaktive Aufgaben, die mit dem eigens entwickelten Tool erstellt wurden (siehe Abschnitt 3).

- **Lernzielkontrolle:** Sie enthält Beispielaufgaben von Klausuren, teilweise mit Musterlösungen, teilweise mit Hinweisen auf die Benutzung der Rechner.

- Außer den erwähnten Aspekten enthält die Selbstlernumgebung weitere Unterpunkte wie **Literatur-** und **Linklisten** und ein **Forum.**

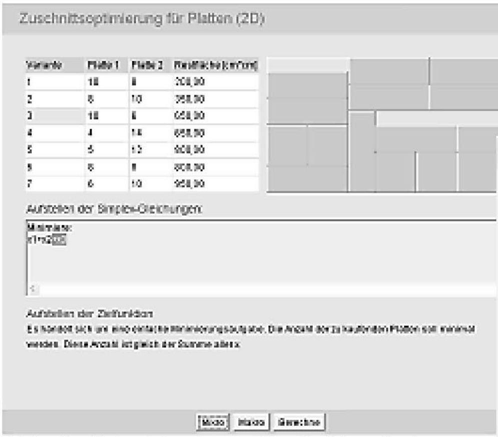
Aufbau des Lehrbuches

Das Lehrbuch wurde primär für Studierende erstellt, die noch über wenig OR-Wissen verfügen. Es trägt dieser Zielgruppe Rechnung und behandelt einführend die wichtigsten Kapitel des OR. Die verschiedenen Verfahren werden hergeleitet und theoretisch begründet; Beispiele zeigen Anwendungsmöglichkeiten auf. Von den Seiten des Lehrbuchs führen Links zu den jeweils zugeordneten Beispielrechnern. Folgende Themen werden u.a. im Lehrbuch behandelt: Lineare, nichtlineare und ganzzahlige Optimierung, Transportverfahren, Tourenplanung, Fertigungsablaufsteuerung, genetische und evolutionäre Verfahren, Warteschlangen und Simulation.

Beispielrechner

Praktisch jeder Algorithmus kann durch Beispielrechner nachvollzogen werden. Grafische Visualisierungen illustrieren in vielen Fällen die Verfahren. Nach der Wahl von selbst gewählten Ausgangswerten kann der Lernende den Ablauf eines Algorithmus Schritt für Schritt verfolgen. Auf diese Weise „entdeckt“ der Benutzer den Algorithmus. Kombiniert mit dem Lehrbuch ist diese „explorative“ Lernweise modern, attraktiv und effizient.

Abbildung 2 zeigt den Beispielrechner „Zuschnittsoptimierung 2D“. Die Bedienelemente „Mikro“, „Makro“ und „Berechnung“ sind in allen Applets einheitlich gestaltet. Über diese steuert der Lernende, wie schnell er durch das Lernprogramm schreiten möchte.



Zuschnittsoptimierung für Platten (2D)

Variante	Platte 1	Platte 2	Restfläche [cm²/cm]
1	18	8	200,00
2	8	10	350,00
3	18	6	650,00
4	4	14	650,00
5	5	12	600,00
6	8	8	800,00
7	6	10	950,00

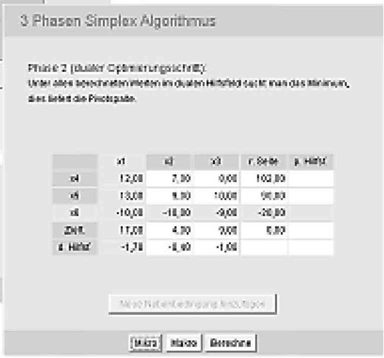
Aufstellen der Simplex-Gleichungen:

Minimiere:
x1+x2

Aufstellen der Zielfunktion

Es handelt sich um eine einfache Minimierungsaufgabe. Die Anzahl der zu kaufenden Platten soll minimal werden. Diese Anzahl ist gleich der Summe aller x.

Abb. 2 - Beispielrechner „Zuschnittsoptimierung



3 Phasen Simplex Algorithmus

Phase 2 (dualer Optimierungsschritt):
Unter allen berechneten Werten im dualen Hilfsfeld sucht man das Minimum, dies liefert die Pivotspalte.

	x1	x2	x3	r. Seite	p. Hilfsf.
x4	12,00	7,00	0,00	102,00	
x5	13,00	8,00	10,00	90,00	
x6	-10,00	-18,00	-9,00	-20,00	
Zielf.	11,00	4,00	9,00	0,00	
d. Hilfsf.	-1,70	-0,40	-1,00		

Abb. 3 - Mikro-Schritt im Dualen Simplex

Mit den „Mikro-“ Schritten kann der Benutzer den Algorithmus ausführlich dokumentiert verfolgen. Abbildung 3 zeigt ein durch den Anwender frei gewähltes Beispiel für den dualen Simplexalgorithmus. Dieser kann Schritt für Schritt durchlaufen werden: Zuerst wird die Pivotzeile markiert; beim nächsten Klick wird die Pivotspalte markiert, usw. Zu jedem Arbeitsschritt wird auch ein kurzer Hinweistext eingeblendet.

Diese Vorgehensweise demonstriert die zuvor erwähnte „entdeckende“ Lernweise. Auch ein iterativer Lernprozess ist dadurch denkbar (lesen, experimentieren, lesen, verstehen, ...), aber nicht zwingend – ganz nach Belieben des Lernenden.

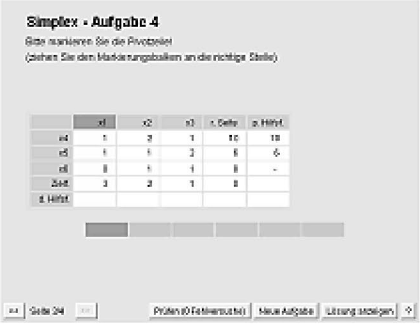
Die folgende Aufstellung liefert einen kurzen Überblick über einige der entwickelten Beispielrechner:

- Simplexverfahren (mit Sensitivitätsanalyse und revidierte Simplexmethode)
- Transportalgorithmen (Vogelsche Approximationsmethode, Modiverfahren)
- Tourenplanung (Saving und Sweep Verfahren)
- Ganzzahlige Optimierung (Gomory)
- Zuschnittsverfahren und optimale Stauraumnutzung
- Fertigungsablaufsteuerung (Johnson)
- Genetische und evolutionäre Verfahren
- Nichtlineare Optimierung (Gradientenverfahren, Kelley, Wolfe)
- Warteschlangen (Markov-Ketten, M/M/x-Systeme, offene Netze)

- Simulation (Zufallszahlengeneratoren, allgemeine Simulationssysteme)
- Netzplantechnik

Aufgabensammlung und Lernzielkontrolle

Die Aufgabensammlung enthält interaktive Aufgaben, durch die der Benutzer seinen Wissenstand automatisch überprüfen kann (Abbildung 4a und 4b).

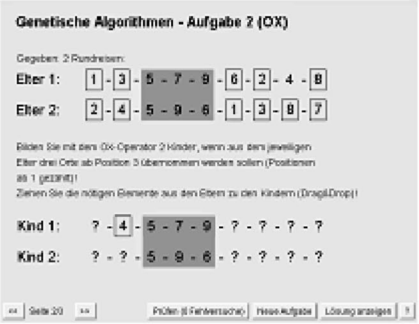


Simplex - Aufgabe 4

Bitte markieren Sie die Pivotzeile
(ziehen Sie den Markierungsbalken an die richtige Stelle)

	x1	x2	x3	r. Seite	p. Hilfst.
x4	1	2	1	10	10
x5	1	1	2	6	6
x6	0	1	1	0	-
Ziel	3	2	1	0	
z. Hilfst.					

Abb. 4a - Drag&Drop Aufgabe zum Simplex-Verfahren



Genetische Algorithmen - Aufgabe 2 (OX)

Gegeben: 2 Rundreisen:

Elter 1: 1 - 3 - 5 - 7 - 9 - 6 - 2 - 4 - 8
Elter 2: 2 - 4 - 5 - 9 - 6 - 1 - 3 - 8 - 7

Bilden Sie mit dem OX-Operator 2 Kinder, wenn aus dem jeweiligen Elter drei Orte ab Position 3 übernommen werden sollen (Positionen ab 1 gezählt)!
Ziehen Sie die nötigen Elemente aus den Eltern zu den Kindern (Drag&Drop)!

Kind 1: ? - 4 - 5 - 7 - 9 - ? - ? - ? - ?
Kind 2: ? - ? - 5 - 9 - 6 - ? - ? - ? - ?

Abb. 4b - Drag&Drop Aufgabe zu Genetischen Algorithmen

Neben dieser Aufgabensammlung werden auch noch eine Reihe von **Lernzielkontrollen** mit Lösungsvorschlägen angeboten.

3. Entwicklungstool „Rapid Assessment Designer (RAS)“

Zur effizienteren Erstellung der Applets wurde ein Editor für Übungsaufgaben entwickelt. Dieses Autorenwerkzeug kann auch anderen Hochschulen zur Verfügung gestellt werden. Es wurde so konzipiert, dass es gut erweiterbar ist und damit auch weitere Aufgabentypen unterstützen kann.

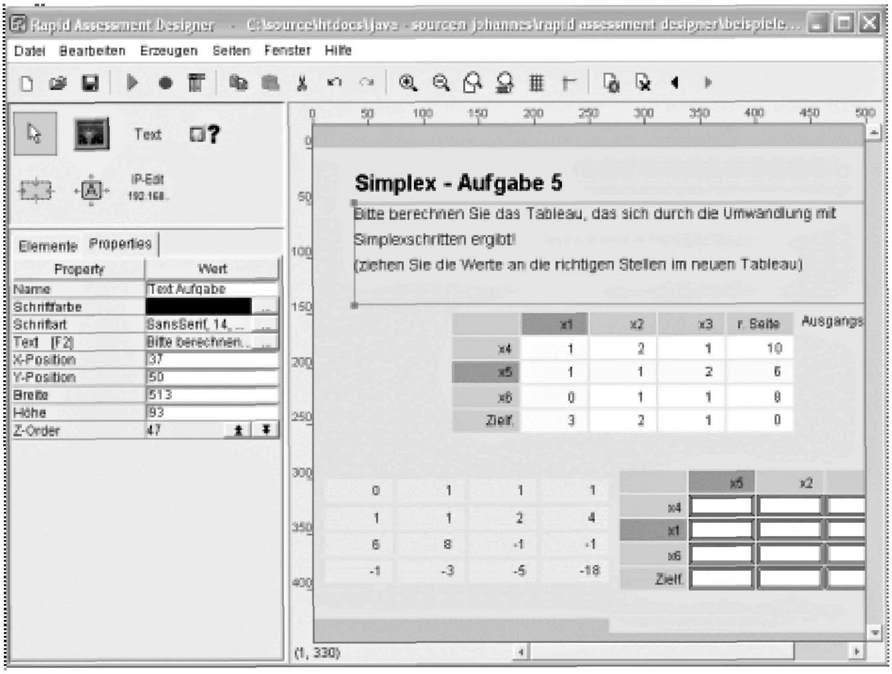
An der FH Augsburg wird der „Rapid Assessment Designer“ mittlerweile mit sehr gutem Erfolg auch für die „Selbstlernumgebung Datenkommunikation“ und in einem Lernprogramm „Kryptographie“ eingesetzt. Die Aufgaben aus Abbildung 4a und 4b wurden beispielsweise mit diesem Tool erstellt.

Features des RAS

Mit dem Entwicklungstool RAS können interaktive Übungsaufgaben (wie z.B. Drag&Drop Aufgaben, Multiple-Choice oder Lückentextaufgaben) vollautomatisch erstellt werden. Die Benutzerführung orientiert sich an bekannten Entwicklungs- und Zeichenprogrammen. Vom Benutzer werden praktisch keine

Programmierkenntnisse erwartet. Als Ergebnis werden Betriebssystem und Browser unabhängige Java-Applets geliefert, die mit fast jeder Lernplattform kombiniert werden können. Nur zur Einbindung der komplett automatisch erstellten Übungen sind HTML-Grundkenntnisse des Anwenders erforderlich.

Eines der wesentlichen Merkmale des Editors ist die einfache Benutzerführung. Arbeitsfläche, Neu-Bereich und Properties (Eigenschaften) wurden getrennt; die Oberfläche ähnelt damit bekannten Entwicklungswerkzeugen wie Borland Delphi, Microsoft Visual C++, Visual Basic oder ähnlichen Programmen. Wer bereits mit diesen Programmen gearbeitet hat, wird mit dem Editor schnell vertraut sein. Abbildung 5 zeigt den Hauptbildschirm und die Arbeitsfläche dieses „Malprogramms für interaktive Übungsaufgaben“:



Property	Wert
Name	Text Aufgabe
Schriftfarbe	■
Schriftart	SansSerif, 14, ...
Text (F2)	Bitte berechnen...
X-Position	37
Y-Position	50
Breite	513
Höhe	93
Z-Order	47

Simplex - Aufgabe 5

Bitte berechnen Sie das Tableau, das sich durch die Umwandlung mit Simplexschritten ergibt!
(ziehen Sie die Werte an die richtigen Stellen im neuen Tableau)

Ausgangs

	x1	x2	x3	r. Seite
x4	1	2	1	10
x5	1	1	2	6
x6	0	1	1	8
Zielf.	3	2	1	0

0	1	1	1
1	1	2	4
6	8	-1	-1
-1	-3	-5	-18

	x5	x2
x4		
x1		
x6		
Zielf.		

Abb. 5 – Der „Rapid Assessment Designer“

4. Evaluation

Die Lernumgebung wird natürlich auch von ihren Benutzern evaluiert. Hier die wesentlichen Ergebnisse (Auszug) der Evaluation im WS 2003/2004 (im Mittel 55 Antworten):

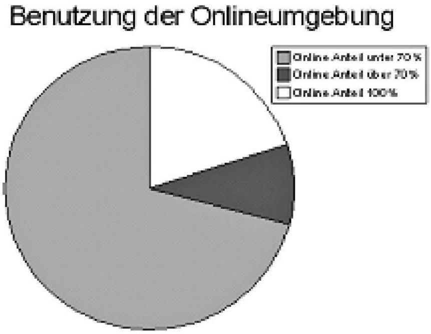
Frage	Bewertung (Schulnote 1-5)
Die Inhalte der Lernumgebung sind gut gegliedert	1,53
Die Inhalte sind klar formuliert und verständlich	1,91
Die Beispielrechner sind hilfreich	1,33

Im Zuge dieser Evaluation wurde von den Studenten aber auch Kritik bzw. Verbesserungsvorschläge an einigen Details geäußert:

- Bei den Beispielrechnern mit freier Eingabe wurden die teilweise recht engen Restriktionen bei den definierbaren Parametern bemängelt. (Bemerkung: Diese sind für eine vernünftige Visualisierung aber zwingend nötig.)
- Bei vielen der Beispielrechner fehlt eine „zurück" Funktion. Mit den „Mikro-" und „Makro-" Schritten kann man den Algorithmus nur vorwärts verfolgen.
- Für einige, weitere Themengebiete sollten noch Beispielrechner und interaktive Übungen entwickelt werden.

Die Prüfungsergebnisse ergaben keine signifikanten Ergebnisse beim Notendurchschnitt; alle Benutzergruppen schnitten in etwa gleich ab. Das Angebot, die Online-Umgebung zusätzlich zu nutzen, wurde vielfach auch von den Studenten, die regelmäßig die Präsenzvorlesung besuchten, angenommen.

Die Tendenz der letzten Semester lässt vermuten, dass die Beteiligung am „Blended Learning" und die Nutzung der Onlinelernumgebung noch deutlich zunehmen wird.



Die Benutzer der Onlineumgebung wurden zu den Vor- und Nachteilen gegenüber der Präsenzlehre befragt. Hier der Auszug einiger Zitate der Studenten, die repräsentativ scheinen:

PRO: „Eine Anwesenheit in der Vorlesung ist nicht erforderlich, die Lernzeiten können z.B. auf den Abend verschoben werden, wenn tagsüber gearbeitet werden muss."

„Lernen, wann ich will."

„Das selbständige Lösen der Probleme erfordert mehr Zeitaufwand, bedeutet aber ein deutlich intensiveres Lernen."

KONTRA: „Es fehlt das direkte, schnelle Feedback mit einem Ansprechpartner. Dies lässt sich auch durch E-Mail und Foren nur bedingt lösen."

„Keine persönliche Ansprache, weniger Kontakt zu anderen Studierenden."

„Ich persönlich nehme viel aus dem Vortrag des Lehrenden mit, das erspart mir viel Nacharbeitungszeit daheim, was beim bloßen Lesen nicht so schnell hängen bleibt."

5. Fazit

Die Selbstlernumgebung wird von den Studierenden sehr gut angenommen, wobei sie häufig als Ergänzung der Präsenzvorlesung benutzt wird. Reines E-Learning hingegen wird nur von wenigen Studenten bevorzugt.

Auf der technischer Seite stellt die Betreuung des Lernprogrammes durch unsere Administratoren keine signifikante Belastung dar. Auch die ständige Erweiterung des Lernprogrammes ist bis dato problemlos verlaufen.

Mit dem bisherigen Nutzen des OR-Lernprogramms sind wir hoch zufrieden. Die Belastungen, die während der Entwicklungszeit zusätzlich entstanden sind, machen sich nun bezahlt.

Wir werden die Lernumgebung um weitere Applets (z.B. für die dynamische Optimierung, neuere Verfahren der linearen Optimierung) in nächster Zukunft erweitern und vorhandene Beispielrechner entsprechend den Wünschen der Studenten verbessern. Auch die interaktiven Übungsaufgaben werden noch weiter ausgebaut.

Literatur

Downloadmöglichkeit für das Entwicklungstool („Rapid Assessment Designer“)

http://www.fh-augsburg.de/informatik/projekte/rad
http://www.Johannes-Kern.de

Selbstlernumgebung „Operations Research“ (samt Literatur- und Linklisten)

http://or-online.informatik.fh-augsburg.de

Partially Integrated Airline Crew Scheduling for Team-oriented Rostering

Markus P. Thiel, Taïeb Mellouli, Yufeng Guo

Decision Support and Operations Research Laboratory, and
International Graduate School of Dynamic Intelligent Systems,
University of Paderborn, Warburger Str. 100, D-33100 Paderborn, Germany
Email: {thiel; mellouli; guo}@dsor.de

Summary. Crew scheduling for airlines requires an optimally scheduled coverage of flights with regard to given timetables. We consider the crew scheduling and rostering process for airlines, where crew members are stationed unevenly among home bases. In addition, their availability changes dynamically during the planning period due to pre-scheduled activities, such as office and simulator duties, vacancy, or requested off-duty days.
Besides this highly complex setting, crew satisfaction becomes more and more important: Recent approaches focus on a variety of quality-of-life criteria like the consideration of individual preferences. Hence most of them are applied to individual rosters; certain aspects especially for the appropriate handling of teams are neglected. Therefore the introduced concept of Team-oriented Rostering addresses the upcoming demand for a reduction of the usually high amount of team changes among crew members within daily or day-by-day team compositions. It relies on the optimal set of roster combinations for the scheduled crew members within the assignment step.

Keywords: airline, crew scheduling (CSP), integrated crew scheduling, crew assignment (CAP), crew rostering (CRP), team-oriented rostering (ToRP), team changes

1 Introduction

The airline Crew Scheduling Problem (CSP) is well-known as one of the most difficult combinatorial problems. Its task is to assign all flights of a given timetable together with further activities to a limited number of crew members stationed at one or several home bases. Besides the consideration of all given activities, operational cost has to be minimized, and workload should be evenly distributed among home bases and crew members.

An airline crew typically receives a monthly or semi-monthly schedule which has to fulfill numerous work rules and regulations. There is a bundle of rigid rules imposed by civil aviation authorities, union contracts, and company policies (e.g. [1,4,6,7]). Less rigid rules considering crew satisfaction and personal preferences can be applied as well.

Due to its complexity the CSP is typically divided into two sequential subproblems [1]: Firstly, in the airline Crew Pairing Problem (CPP) a set of pairings is generated that minimizes operational cost in such a way that each flight belongs to exactly one pairing (set of flight leg which starts and ends at the crew member's home base). Secondly, the airline Crew Assignment Problem (CAP) or airline Crew Rostering Problem (CRP) assigns generated pairings together with other scheduled activities, training, vacations, and requested off-duty periods. In order to build legal crew schedules or rosters for each crew member, all company rules and regulations must be satisfied. For a recent annotated bibliography on the state-of-art on scheduling and rostering we refer to [2].

In our research on the airline CSP, see [3,5], we propose a partially integrated procedure to solve the airline crew scheduling problem, thus making a contribution towards an exact optimal solution of the fully integrated CSP. We have investigated models that generate not just pairings, but *pairing chains*, in the first step, taking guaranteed individual scheduled activities of crew members into account. A pairing chain is a sequence of pairings which covers the scheduled time period, incorporating weekly rests so that all valid rules and regulations have been taken into account. The main benefit can be seen in the earlier consideration of the daily available crew capacity for each home base in the first step, thus significantly reducing the need for expensive changes in the assignment step.

In this paper we want to present how specific requirements for the newly introduced Team-oriented Rostering approach can be address within the above mentioned integrated crew scheduling approach, especially its underlying special network structures for the pairing phase.

The paper itself is structured as follows: We give a brief introduction to the general Team-oriented Rostering Problem, and some characteristics for cockpit crew in particular. In Section 3 we present the so-called aggregated time-space networks that we apply for our pairing generation and describe how this can be modified to meet the additional requirements for the team-oriented Rostering approach. We close with a brief result review in the conclusion.

2 Team-oriented Rostering

Hereby, we briefly introduce the special characteristics of the Team-oriented Rostering Problem (ToRP) in general and in particular for cockpit crew: The goal of Team-oriented Rostering is expressed by the creation of automated crew schedules that reveal a certain team-orientation. This team-orientation intends to grant higher crew satisfaction in terms of quality-of-life criteria. Its basic idea is in addition to the objectives of the airline CRP the consideration of team-orientation by avoidance of frequent changes in the composition of a servicing or operating onboard team. The approach itself is situated as an enhancement to the personalized rostering concept widely applied in Europe, which aims to fair-and-equal share of workload. This is in contrast to well-known bidding systems or preferential bidding approaches which generate their crew schedules based on strict (or released) seniority [4].

Why is team-orientation this much important? It is known that crew satisfaction is highly dependent on the colleagues he/she works with [6]. In current approaches some crew members may prefer to exclusively work with the same colleague(s) over a long time period (e.g. *married couples* or *must-fly-together restrictions* [4,6]). The realization of such constraints is comparably simple, but this it is almost impossible to implement without great financial losses, because of different, non-overlapping scheduled activities at most airlines. Therefore, teams should be kept as flexible as possible. On the other hand, aircraft security but also quality-of-service for passengers are directly at risk in cases of disharmonies among operating cockpit and servicing cabin crew. Especially team changes have been identified to have a negative impact on the individual crew satisfaction, e.g. being left alone in a non-domicile town after work or giving up harmonizing working teams.

Before discussing some more details, we give some necessary definitions used in our approach: A *team* is to be understood as the grouping of crew members with, if required, different crew functions and quantities in a way that a single or a series of flight legs is adequately staffed. Crew members of such a team may origin from different home bases, but they all share the minimum qualification for the fleet to be operated. A *shared flight activity (SFA)* is defined to be the smallest unit that is considered in this approach: Such an activity is serviced by a team without any team change. It could be a single or multiple flight legs, flight duties (set of flights operated within one day), or even a complete pairing. Such pieces-of-work are extracted from the generated pairings of the CPP. A *team change* occurs if at least one crew member is scheduled to service the next flight activity together with a different team composition (other colleagues). Such team changes occur due to the obeyed rule sets (e.g. a crew member has reached his maximum of daily working hours), or by very strict fair-and-equal share of workload; but so far, the main reason for team changes lies in the fact that (due to the dominating cost minimization objective) they are simply not considered at all.

Since the described ToRP approach tries to minimize the number of team changes, their occurrences are considered via so-called *team change penalties*. Those penalties are usually chosen as positive values. In contrast to this, negative team change penalties (or bonuses) can be applied for benefits of servicing as a team while, e.g., saving operational cost by sharing a taxi. We also differentiate between different *types of a team change.* This consideration evaluates with regard to *when* and *where* the team change occurs. They can happen within the day, over night, both at the home base or outside, or after the weekly rest period at the home base. Since changes within the day are the most undesired, especially in combination with an outside location, we propose a clear hierarchy among those listed instances with decreasing penalty values for each type.

The main disadvantage of our ToRP approach can be seen in the fact that a resulting crew schedule focuses in addition on the minimization of team-changes is most likely more cost intensive than the one without team-orientation. In general, there is a trade-off between the minimization of operational cost and the minimization of team changes: Assigning team change penalties leads at a certain point to outweigh operationally less expensive rosters in preference to those with higher team-orientation, e.g. fewer team changes. Due to the penalization of team changes among roster combinations the aim of the ToRP is hereby defined as the search for an appropriate set of individual rosters (one roster for each crew member) that all given flights are covered properly at minimum of cost with a socially and economically reasonable reduction of team changes.

A more detailed problem analysis can be found in [8]. Additionally we present and discuss two alternative IP-formulations for the ToRP: The *Extended Rostering Model* penalizes team changes by additional columns and rows for selected roster combinations, whereas the *Roster-Combination Model* treats operational cost and team change penalties simultaneously within the proposed model.

Throughout our computational experiments and as mentioned above, the ToRP approach should be based on the assignment of shared flight activities, not pairings. In Fig. 1 we show the differences between both terms: Although there are two (one-day) pairings for captains and two for first officers, their assignment leads directly to four team changes among all crew members within this single day.

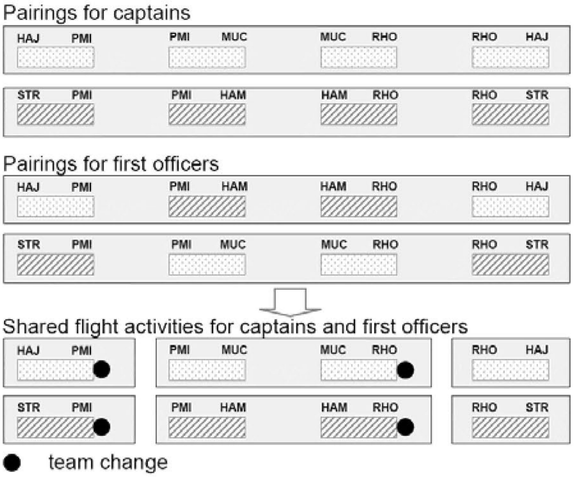


Fig. 1. Team changes within pairings

This is a typical issue within the pairing generation phase: During the creation of the above mentioned pairings it is very likely to have a certain asynchrony for the executing crew team. Therefore, assigning pairings at this point is not sufficient for the minimization of team changes on the one hand, because otherwise only those between pairings can be taken into account, all changes inside pairings would be kept as fixed. On the other hand, the main drawback of working with probably quite short SFAs increases the combination basis for the roster generation, by which we might loose most benefits from the optimization in the pairing generation step. (In the example we have now six SFAs over both crew functions instead of two separate assignment problems with only two pairings each.)

3 Time-space Networks for Paring Chain Generation

In contrast to the well-known set partitioning approach for CPP, see e.g. [1,2,3], we have investigated solving the above airline crew pairing chain problem (CPCP, a strict extension of the standard CPP) in one step using a dedicated time-space network (TSN). The model which we discuss in [3,5] is able to consider multiple home bases with time-dependent crew capacities and allows an even partitioning of pairings and flight time among home bases and crew members. Because of the powerful extension of CSP's first step, namely the so-called pairing chain generation, the results are expected to be highly useful and adoptable for the second scheduling step: the assignment or rostering.

In this paper we present the basic structure of the TSN and its extension regarding the requirements for an appropriate generation of shared flight activities for the subsequent Team-oriented Rostering step as described above.

Aggregated Time-Space Networks: A Basic Flow Formulation

We assume that a timetable T is given with a set of flights, involving a set of airports A. For each airport $k \in A$, let E^k be the list of all arriving flight legs of T at airport k, and S^k the list of all starting flight legs of T from airport k. Thus, each flight leg i of T occurs twice, namely, as an end event (ending flight leg) in $E^{dest(i)}$ and as a start event (starting flight leg) in $S^{orig(i)}$.

Preprocessing: We aggregate end and start nodes at each airport in order to be able to reduce the network size without loss of generality. For each airport $k \in A$, flight legs in the lists E^k and S^k are chronologically sorted with respect to their end times and start times, respectively. Let us partition E^k and S^k into *end blocks* and *start blocks*, respectively, such that end times of E_l^k flights are less than or equal to start times of S_l^k flight legs which in turn are strictly less than end times of E_{l+1}^k flights. (We assume that the end times include the required minimum turnover time as well.) This partition is unique, so let w_k be the resulting number of end blocks, which is equal to the number of start blocks, at airport $k \in A$. Each end block E_l^k $(l = 1,\ldots,w_k)$ precedes the corresponding start block. Only E_1^k and/or S_{wk}^k may be empty for some airports k.
Afterwards, the network can be built according to the following structure:

Network structure: For each airport $k \in A$ and $l = 1,\ldots, w_k$, each ending flight leg of E_l^k is compatible with each of the starting flight legs of S_l^k. Representing flight legs by arcs and crew members by flow units, this compatibility is established through the *connection node* n_l^k, see Fig. 2. Observe that each ending flight leg of E_l^k is also compatible with each starting flight leg of $S_{l'}^k$ for each $l' > l$. This is established by traversing the connection nodes n_l^k, n_{l+1}^k, … , $n_{l'}^k$ along horizontal *waiting arcs* following the time axis.

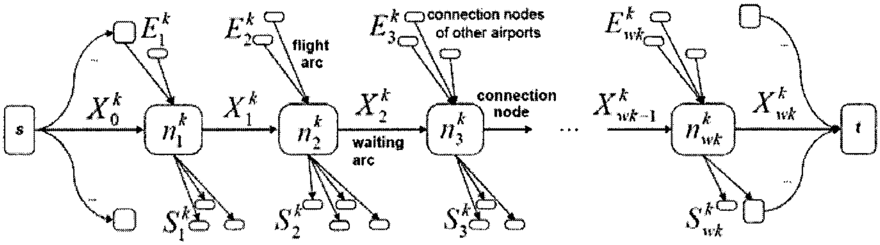


Fig. 2. Aggregation of flight legs and connection line of the aggregated TSN

For each airport $k \in A$, let us call the line constituted by the connection nodes together with the waiting arcs a *connection line* CL^k. Thus, a connection line CL^k regulates all possible flight leg connections at airport k; *direct* connections (for aggregated ending flight legs and aggregated connection flight legs) through one connection node, and *indirect* connections through several connection nodes linked by *waiting arcs*. Here, crew members at the same airport, waiting at the same time for indirect connections between two connection nodes, are aggregated. Thus, an *aggregated time-space-network* has been constructed.

Disaggregation: In general, such networks allow numerous ways of disaggregation, even with identical total operational cost for crew. After having computed a certain flow on a TSN, the freedom one can take when building a concrete schedule becomes clear: Suppose that a computed optimal flow delivers flow values $X_0^k = 3, X_1^k = 2$, and $X_2^k = 0$ at a airport $k \in A$ for ending and starting flight legs as in Fig. 2. That is, three crew members are available at airport k at the beginning and two other crew members enter airport k through flight legs of E_1^k before flight legs of S_1^k start. At this moment, three arbitrary crew members from the waiting five may be chosen to serve the flight legs of S_1^k. The remaining two crew members together with the two entering airport k through flight legs of E_2^k are then used to serve the flight legs of S_2^k .

Extension of aggregated TSN: As described in Section 2, our ToRP approach requires the consideration of SFAs. Therefore, we introduce a model extension to the above given TSN. Currently, the presented pairing chain generation works independently for each crew function which results in a set of pairings with little synchrony. As a consequence the information of the generated pairing chains for the first crew function should be utilized within the optimization runs of the remaining crew functions. Such hierarchical proceeding allows the introduction of additional bonus arcs representing the avoidance of team changes within a pairing. As given in Section 2, those bonuses (negative penalties) should be granted based on the described levels in the team change classification, e.g. team changes within the day, over night, outside home base etc. It is important to mention that those bonus values should increase by the duration of the granted synchronous (sub)pairings, nevertheless they should not exceed a certain value at which the sum of bonuses outbalances the selection of optimal pairing chains for the examined crew function. (Such high bonuses may absorb undesirable additional operational cost, e.g., for hotel and taxing.) Additional bonus arcs that cover more than one pairing are not required since all chains will be decomposed into pairings in the rostering phase. Fig. 3 shows the application of such additional bonus arcs which represents the connection between three airports by two flight legs and one waiting arc.

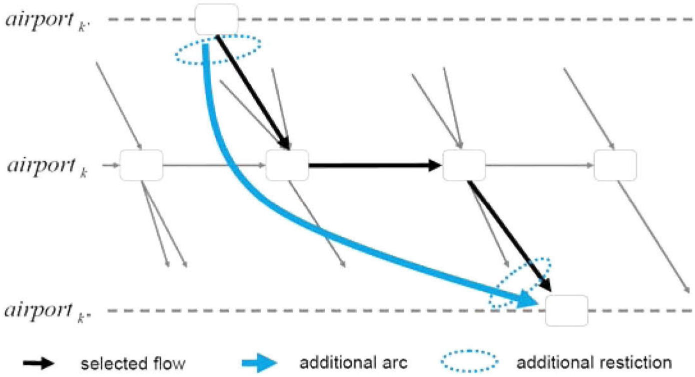


Fig. 3. Bonus arcs for avoided team changes regarding the pre-selected (sub-)pairings

As pointed out in Fig. 3 the compliance to the inflow-outflow requirement for each connection node requires one additional restriction of each bonus arc. It has to ensure that, for the case that the bonus arc is chosen, the sum over all flow units for the considered flight legs must be zero.

4 Conclusion

In this paper we presented the Team-oriented Rostering approach which aims to reduce the overall number of team changes among crew members during their working period. As the basis for this approach we apply our partially integrated crew scheduling formulation which we described in [3,5]. Due to the fact that we experience multiple team changes within the generated pairings, we proposed an extension to the (state-expended multi-commodity) aggregated time-space network optimization model in this paper. Via additional arcs we developed a useful way to build sufficient shared flight activities as they are required for the ToRP approach. Although we follow the idea of partial integration among the two scheduling steps, we have to suggest a strict hierarchical paring generation for the involved crew functions. Further improvements should pay attention to its simultaneous consideration (at least partially) which is so far impossible to realize within the proposed time-space network due to its high aggregation level.

As a result the extensions to the network model due to the preference of defined (sub)parts of one crew function's pairings allow and (for proper bonus setting) do not enforce quite synchronic pairings within their generation phase. The number of the improved shared flight activities indicated a greatly notable reduction of asynchrony of parings as they resulted from the multi-optimum solutions that we observed before. By this, shared flight activities that consist only of a single flight occur quite rarely, and from the technical point of view these little modifications allow to solve larger instances or to reduce computational time because of smaller model sizes. Testing on several instances of our project partner, a European tourist airline, indicated that the solution quality in terms of operational cost and crew satisfaction is not afflicted by the above mentioned model extension.

References

1. Barnhart C, Cohn AM, Johnson EL, Klabjan D, Nemhauser GL, Vance PH (2003) Airline Crew Scheduling. In: Hall RW (ed) Handbook of Transportation Science, Kluwer Scientific Publishers
2. Ernst AT, Jiang H, Krishnamoorthy M, Owens B, Sier D (2004) An Annotated Bibliography of Personnel Scheduling and Rostering. Annals of Operations Research 127, 21-144, 2004.
3. Guo Y, Mellouli T, Suhl L, Thiel MP (2003) A Partially Integrated Airline Crew Scheduling Approach with Time-Dependent Crew Capacities and Multiple Home Bases. Working papers WP0302, Decision Support & OR Lab, University of Paderborn, submitted to EJOR
4. Kohl N, Karisch SE (2004) Airline Crew Rostering: Problem Types, Modeling, and Optimization. Annals of Operations Research 127:223–257, Kluwer Academic Publishers
5. Mellouli T (2003) Scheduling and Routing Processes in Public Transport Systems. Professorial dissertation, University of Paderborn
6. Strauss C (2001) Quantitative Personaleinsatzplanung im Airline Business. Peter Lang Publishing Group, Frankfurt am Main
7. Suhl L (1995) Computer-aided scheduling: an airline perspective. Deutscher Universitätsverlag, Wiesbaden
8. Thiel MP (2004) Team-oriented Rostering for Cockpit Personnel, Working papers WP0406, Decision Support & OR Lab, University of Paderborn, submitted to CASPT 2004 in San Diego

Multi Objective Pinch Analysis (MOPA) for Integrated Process Design

J. Geldermann, H. Schollenberger, M. Treitz, O. Rentz

French-German Institute for Environmental Research (DFIU / IFARE)
University of Karlsruhe (TH), Hertzstr. 16, D-76187 Karlsruhe
Tel: +49-721-608-4583; Fax: +49-721-758909,
{jutta.geldermann; hannes.schollenberger; martin.treitz; otto.rentz}@wiwi.uni-karlsruhe.de

Abstract

The combination of process integration and Operations Research enables an integrated technique assessment and a subsequent process design. The application of Multi Objective Pinch Analysis (MOPA) permits the identification of overall saving potentials for energy, water and Volatile Organic Compounds (VOC). In this paper the general concept of MOPA is described and its application is shown in its first steps for a bicycle coating plant in Chile.

Introduction

Most of the common VOC-reducing measures (e.g. for paint and adhesive applications) are installed as secondary measures (e.g. thermal incineration), process-integrated measures (e.g. reduced-overspray methods, powder coating) or as product-integrated measures (e.g. high-solid coatings). Further optimisation potentials can be identified by integrating additional unit operations (e.g. heat exchangers, condensers, etc.) into the plant layout and linking mass and energy flows between different production plants. Besides VOC emissions, energy and water consumption are also of significant interest. Therefore, a three dimensional problem must be solved in order to identify the overall set of feasible solutions.

In this paper, the coating of bicycle frames is investigated in order to demonstrate the underlying methodology. In the next section, the pinch analysis and its role in integrated process design as Multi Objective Pinch Analysis (MOPA) is described. In the third section, the example of bicycle frame coating is calculated, while in the last section conclusions are drawn.

Multi Objective Pinch Analysis for Integrated Process Design

The pinch analysis is an approach for integrated process design. First applications were exclusively addressed to energy optimisation [7; 9] and as a consequence thereof, to the design of heat exchanger networks (HEN). In the last two decades applications targeting wastewater minimisation and VOC recovery from waste gas streams have been developed [1; 3; 12; 13]. Additionally, the algorithms for solving the design problems have been developed further, and in some cases have been adapted from the field of Operations Research (e.g. the transport algorithm [2; 5]).

Past investigations mainly focused on one target, whereas the approach presented in this paper aims at a Multi Objective Pinch Analysis (MOPA) which is characterised by simultaneous consideration of energy, VOC and water. The results of the different pinch analyses are combined using Multi Criteria Analysis approaches.

Energy

The basic idea of energy pinch analysis is a systematic approach to the minimisation of lost energy in order to come as close as possible to a reversible system. In its first step the pinch analysis yields the best possible heat recovery. Further recovery can only be achieved by changing conditions or structures of the investigated system (e.g. flow rates, pressures etc.) [11].

The pinch analysis requires the combination of hot and cold process streams to composite curves and the description of the respective temperature - enthalpy relationships. Additionally, a minimum temperature gradient ΔT_{min} must be set representing the driving force of the heat transfer. The pinch point, where the distance between the hot and cold composite curves is equal to ΔT_{min}, denotes the optimal internal heat transfer between the hot and cold flows [9].

The pinch point is always a corner or end point of at least one of the composite curves. Its temperature is less than ΔT_{min} above or below any of the temperatures spanned by the other composite curve. At the pinch point, either the slope of the cold composite curve becomes flatter or remains equal, or the slope of the hot composite curve becomes steeper or remains equal above the pinch point, or both [2].

The result of the pinch analysis is the energy saving potential for the considered set of processes representing the target for the subsequent design process. Furthermore, information is obtained on the amount of heat exchange required between the appropriate streams minimising the use of hot and cold utilities. Depending on the chosen design constraints, which reflect technical and chemical requirements, the actual savings are determined resulting in an economically feasible solution. Consequently, the layout planning is driven by a trade-off between equipment (e.g. capital for heat exchanger) and operating cost (energy utility cost).

Volatile Organic Compounds (VOC)

Besides heat exchange, another application of pinch analysis is on VOC or multicomponent VOC since their separation from waste gas is usually carried out via thermal condensation [3; 10]. This permits a translation of organic solvents reclamation into a heat exchange problem. By using phase diagrams the targeted VOC concentration can be described by temperatures of the gaseous waste stream [4]. Thus, the required temperature intervals for applying the pinch analysis can be obtained [11]. When using multistage condensation for VOC recovery the waste gas can be pre-cooled by the cold cleaned gas stream. The approach of incorporating the cleaned gas stream can also be applied when other emission reduction measures are used (e.g. thermal incinerators [5]).

The application of pinch analysis and subsequent integrated design are used to find the most cost-effective solution. Hence, the amount of solvent to be recovered can be determined [10]. The total costs in the case of VOC are a combination of those for purchased solvents and those for condensation. The recovered solvent can be reused in the same process or it can even be sold (e.g. for cleaning applications), depending on the option[1] finally selected.

The properties of the employed solvents (VOC) determine the temperature intervals used in the pinch analysis. In the first step, a total recovery of consumed solvents is considered. In a subsequent step, a feasible economic solution is obtained by a techno-economic assessment of available technologies that must be translated into an adapted design and results in a new pinch analysis.

The technical constraints that must be considered in the design phase are the maximum and minimum solvent concentration given by the process itself. For example, in order to allow quick drying and to limit the lasting effects of the solvent on the object, the concentration of the air inside the drying oven should not exceed a certain value. Additionally, safety requirements must be carefully followed due to the explosion risk originating from the solvents. When using solvent condensation as a recovery measure dehumidification and the type of cooling agent and their related constraints must also be considered when looking for the most economically feasible solution. This solution, obtained from the results of the VOC pinch analysis, then provides the amount of solvent that can be recovered and the network design needed for its realisation.

[1] The reuse of the recovered VOC depends on the purity of the condensed solvent and the quality requirements of the investigated process.

Water

In addition to heat exchange and VOC recovery the pinch analysis can also be applied to calculate water and wastewater savings. Two different problem types can be addressed: single or multiple parameters. Single parameter problems are comparable to the ones solved with the energy pinch. They consider the mass transfer of one contaminant from a rich to a lean stream, equivalent to the hot and cold streams of the energy pinch.

Instead of regarding temperature-enthalpy relationships the water pinch regards concentration-mass load curves. The mass load is calculated by multiplying the concentration with the water flow of the stream. A linear behaviour is assumed for the transfer which is a good approximation for diluted streams (e.g. water used for washing). All streams of the investigated system are combined to a limiting composite curve describing the “worst” water quality acceptable. The freshwater curve describes the water supply of the system. The slope of the curve is a measure for the flow rate. The water supply and limiting curve match at the pinch point and the resulting slope defines the minimum water flow rate needed [12].

Practical problems often require the consideration of several contaminants or parameters. If it is not possible to aggregate multiple contaminants to one parameter (e.g. Chemical Oxygen Demand - COD) in order to translate the problem into a single parameter one, an iterative process must be applied to find the overall pinch for all “water-based” flows in the system [8].

Furthermore, water regeneration can be depicted with the water pinch analysis. This leads to changes in the description of the water supply line which shows changes in flow rates resulting from the composition of the regenerated and the fresh water flow [12].

Besides the different parameters to be considered (single or multi parameter, water regeneration), a distinction between water as a utility (utility water pinch analysis) and water as a substance required in the process itself (process water pinch analysis) must also be made. A new method addressing the problem of several aqueous streams relevant for one operation including water losses has been developed [6].

Similar to the energy pinch, the solution obtained with the water pinch must be translated into a feasible process design. Therefore, strategies such as bypassing of unnecessary water in order to always ensure use of the minimum flow rate can be applied. Typical freshwater savings identified and realised with water pinch analysis and the subsequent process design are in the range of 15-40 %.

Integrated approach

In the approaches of the pinch analyses discussed so far, only single energy and mass flows were regarded. Through the parallel reduction of water use, energy consumption and VOC emissions, a multi-criteria process design problem for company-wide facility planning must be solved, in order to implement efficient recycling cascades. Resulting conflicting solutions must be evaluated based on a multi-criteria approach. For this purpose convenient OR models (for instance the Outranking-Method PROMETHEE) can be employed and modified. In addition, this will also introduce new possibilities for cleaner technologies and environmental protection by optimally combining process integrated emission reduction measures and end-of-pipe technologies and also creating a challenge for research in the field of Operations Research.

Case Study: Coating of Bicycle Frames

The case study analyses a typical bicycle production facility within an industrialising country. A baseline scenario using independent fresh air sources for each oven, is compared to a scenario using the generated waste heat of the production process. Calculations are done using the classical transport algorithm as described in Geldermann et al. [5].

The coating of bicycle frames is important for protecting the frame against corrosion, chemical deterioration, etc., and for maintaining the best possible optical quality. The case study illustrates a reference installation producing 200 bicycles per day, with the process schema given in Fig. 1. The coating starts with a pretreatment step (degreasing, phosphating, passivation) which is not considered in this calculation. Nevertheless, it requires a subsequent drying step. Consecutively, a two-layer coating consisting of a filler and a two paint application steps are executed. Consequently, four drying processes are required generating energy demand and solvent emissions.

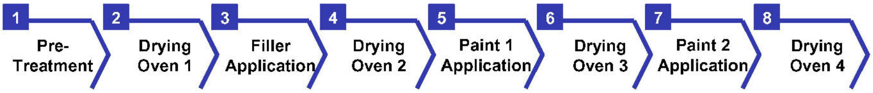


Fig. 1. Process Steps of Bicycle Coating

The drying processes are characterised by their required process temperatures (oven 1: 155 °C; oven 2: 90 °C; oven 3: 125 °C; oven 4: 150 °C) and length of the drying oven (16m, 31m, 13m, 14m). Both the drying air of the process requiring heating and the cooling air of the counter-current flow are taken from the outside air supply (assumed to be 20°C) with a flow rate per meter of drying oven of 500 Nm^3/h. In the baseline scenario the waste gas is released after the drying step without any cleaning. In the investigated scenario the waste gas is cooled down to a temperature of 40°C and released also without any cleaning activity.

Auxiliary processes such as the preheating of drying ovens, the cleaning of tools, or the mixing of the filler or paint exist in addition to the main painting process, but are not included in this case study. Nevertheless they may also provide significant potential for the conservation of resources.

For the filler application a multi-component solvent with a mass proportion of 2:1:1 between Isobutyl Acetate (CAS[2] 110-19-0), Naphthalene (CAS 91-20-3) and Toluene (CAS 108-88-3) is assumed. For the two paint applications a multi-component solvent of Butyl Acetate (CAS 123-86-4), Naphthalene (CAS 91-20-3) and the paint thinner Methyl Isobutyl Ketone (CAS 108-10-1) with a 2:1:1 mass proportion is assumed. The quantities of released solvents (oven 2: 0.53 kg/h; oven 3: 0.60 kg/h and oven 4: 0.32 kg/h) are calculated using the lasting dried paint of the frame (20 g of filler, 15 g of paint 1, 8 g of paint 2) and the respective solvent content (filler: 40%, paint 1 and 2: 50%). The required temperature-dependent heat capacity for the calculation is calculated by the following polynomial equation:

$$C_p = \alpha + \beta \cdot T + \gamma \cdot T^2 + \delta \cdot T^3 + \varepsilon \cdot T^4$$

where C_p is the heat capacity in kJ/kmol and T is the temperature in K. The coefficients are taken from the CHERIC-Database[3]. Furthermore, the temperature dependent density of air and the solvent-air-mixture is calculated by the relation of air pressure divided by the gas constant times the temperature in Kelvin.

It should be noted that in the following calculations only the collected VOC are considered, but due to the different production environments the diffuse solvents might also be considerable high. The mass flow of air within the drying ovens is supplemented by the released mass of solvents.

Table 2 summarises the findings as the basis for the calculation. The table shows the detailed data used as the basis of the composite curves. The artificial cold and hot process streams shown in the table originate from one or more existing process streams of the drying process.

[2] Chemical Abstracts Service registry numbers are copyright by the American Chemical Society.

[3] Chemical Engineering Research Information Center (CERIC): Korea Thermophysical Properties Data Bank; http://www.cheric.org/research/kdb/

Table 1. Scenario Using Waste-Heat

	Temp. Inlet [°C]	Temp. Outlet [°C]	C_p [kJ/(kg·K)]	Density [kg/m^3]	Flow Rate [m^3/h]	ΔH [kJ/h]
Cold Process Streams	-∞	20	-	-	0	0
	20	30	1.0184	1.1843	15.500	186,944
	30	80	1.0239	1.0758	15.500	853,671
	80	90	1.0301	0.9858	15.500	157,398
	90	115	1.0340	0.9398	15.500	376,554
	115	125	1.0381	0.8980	15.500	144,493
	125	140	1.0412	0.8703	8.000	108,738
	140	145	1.0437	0.8493	8.000	35,457
	145	150	1.0450	0.8392	8.000	35,079
	150	155	1.0463	0.8294	8.000	34,712
Hot Process Streams	30	20	-	-	0	0
	40	30	-	-	0	0
	90	40	1.0259	1.0440	15.500	830,056
	100	90	1.0324	0.9589	15.500	153,445
	125	100	1.0364	0.9154	15.500	367,629
	135	125	1.0407	0.8757	8.000	72,907
	150	135	1.0438	0.8493	8.000	106,380
	155	150	1.0462	0.8294	8.000	34,709
	160	155	-	-	0	0
	+∞	160	-	-	0	0

The baseline scenario requires a heating duty of 4,255,877 kJ/h. Since all waste heat is released without being used considerable amounts of energy and solvents are released into the environment. Given the thermodynamic optimal solution requiring only around 367,920 kJ/h heating utility, a significant energy saving potential can be identified. The heat transfer table (cf. Table 2) shows which flows to connect. For example the heating duty of the cold stream 2 is completely satisfied by the hot utility. The cold process flows 3 to 8 are partially and the streams 9 and 10 are completely fulfilled by the hot utility. In a subsequent step the target must be evaluated considering the available technological options.

Table 2. Heat-Transfer Table

		Hot Process Streams										Hot
		1	2	3	4	5	6	7	8	9	10	Utility
Cold Process Streams	1											
	2											186,944
	3			830,056								23,615
	4				153,445							3,953
	5					376,629						8,925
	6						72,907					71,586
	7							106,380				2,358
	8								34,709			748
	9											35,079
	10											34,712
Cold Utility												

Conclusions

In the calculated example energy recovery was addressed. However, instead the substance properties of the solvents or their economic value could also be of interest. This leads to a multi-criteria problem. Depending on the emphasis given to the material properties of the solvents or the financial options, different rankings can be obtained.

In this paper the principles of Multi Objective Pinch Analysis were demonstrated, specifically its role in the context of process design. It consists of a combination of pinch analysis with different targets (energy, wastewater, VOC) and a following multi-criteria analysis. A set of optimal solutions is delivered, which span the solution space as a pareto surface. Identifying applicable solutions on this pareto surface requires detailed economic and technological information (e.g. prices; type of heat exchanger; exchanger surface; type of VOC-condenser; capacities; water treatment systems etc.). In a multi-criteria decision process the preferences, according to the different resources, conclusively determine the selection of a set of required technologies. These specific technologies implemented in the subsequent process design eventually define the savings being realised. In an iterative process the new design can be evaluated by MOPA.

The mathematical formulation of the problem allows the additional depiction of further constraints (e.g. technical restrictions and/or chemical behaviour of the involved substances). The additional application of environmental assessment methods in the multi-criteria analysis, especially in the case of VOC recovery, may deliver additional information about the best technological option. In the future, not only will process changes through new investments be included but also retrofitting. In this case, the trade-off between equipment (e.g. heat exchanger) and operating cost is even more complex as investments have already been made.

This work is part of the project „PepOn: Integrated Process Design for Inter-Enterprise Plant Layout Planning of Dynamic Mass Flow Networks" funded by the Volkswagen-Stiftung. We thank the VolkswagenStiftung for their excellent support of our research.

References

[1] Alva-Argáez, A., Kokossis, A. C., Smith, R. (1998): Wastewater minimisation of industrial systems using an integrated approach; Computers and Chemical Engineering; vol. 22; pp. S741-S744

[2] Cerda, J., Westerberg, A., Manson, D., Linnhoff, B. (1983): Minimum utility usage in heat exchanger network synthesis - A transportation problem; Chemical Engineering Science; vol. 38; no. 3; pp. 373-387

[3] Dunn, R. F., El-Halwagi, M. M. (1994): Optimal design of multicomponent VOC-condensation systems; J. of Hazardous Materials; vol. 38; no. 1; pp. 187-206

[4] Geldermann, J., Schollenberger, H., Rentz, O. (2004): Integrated Scenario Analysis for Metal Surface Treatment ; Int. J. of Integrated Supply Management; (accepted)

[5] Geldermann, J., Treitz, M., Rentz, O. (2004): Integrated Technique Assessment based on the Pinch-Analysis Approach for the Design of Production-Networks; European Journal of Operational Research; (accepted)

[6] Hallale, N. (2002): A new graphical targeting method for water minimisation; Advances in Environmental Research; vol. 6; pp. 377-390

[7] Kobayashi, S., Umeda, T., Ichikawa, A. (1971): Synthesis of optimal heat exchange systems - an approach by the optimal assignment problem in linear programming; Chemical Engineering Science; vol. 26; pp. 1367-1380

[8] Koufos, D., Retsina, T. (2001): Practical energy and water management through pinch analysis for the pulp and paper industry; Water Science and Technology; vol. 43; no. 2; pp. 327-332

[9] Linnhoff, B., Flower, J. R. (1978): Synthesis of Heat Exchanger Networks; AIChE J.; vol. 24; pp. 633

[10] Parthasarathy, G., El-Halwagi, M. M. (2000): Optimum mass integration strategies for condensation and allocation of multicomponent VOCs; Chemical Engineering Science; vol. 55; pp. 881-895

[11] Umeda, T., Harada, T., Shiroko, K. (1979): A Thermodynamic Approach to the Synthesis of Heat Integration Systems in Chemical Processes; Computers and Chemical Engineering; vol. 3; pp. 273-282

[12] Wang, Y. P., Smith, R. (1994): Wastewater Minimisation; Chemical Engineering Science; vol. 49; no. 7; pp. 981-1006

[13] Zhelev, T. K., Semkov, K. A. (2004): Cleaner flue gas and energy recovery through pinch analysis; Journal of Cleaner Production; vol. 12; pp. 165-170

Revenue Management in a Make-to-Order Environment

Stefan Rehkopf and Thomas Spengler

Institute for Economics and Business Administration, Department of Production Management, TU Braunschweig, Katharinenstraße 3, D-38106 Braunschweig, Germany, {s.rehkopf, t.spengler}@tu-bs.de

Abstract. Manufacturing companies in a make-to-order environment sell customer specific products. Usually, a company offers more than one product, faces a stochastic demand, and a time-varying product price. The latter cannot entirely be set autonomously by the company due to competitors. The decision the company has to make for an incoming order is whether to accept or reject the order, depending on the remaining capacity, the contribution margin of the order and the orders expected for the future. In our contribution we will discuss if the methods for the airline revenue management are applicable to the described problem. After giving an overview about the requirements a decision support system ought to fulfill to improve the order selection process in a make-to-order environment, we will discuss the differences of such a system to existing revenue management approaches in service industries. Subsequently, we give a mathematical formulation for the described problem and apply it to a simplified practical example.

1 Make-to-Order Manufacturing, Introduction and Problem Definition

Manufacturing companies in a make-to-order environment sell products specific to consumer requirements. In general, an incoming order initiates the production process, i.e. the procurement of specialized materials and component parts, the in-house fabrication of parts, the production of subassemblies and the final assembly. Often the product itself or a variant of the product has been manufactured before, and hence, the bill of materials as well as measures for the expected processing time of the order at the resources are available [4].

We will focus in our contribution on the production processes in the iron and steel industry as an example of a make-to-order production process. Our reference production structure originates from the production structure of the Salzgitter AG, Germany. The steel making process [10] in general starts with the pretreatment of iron ores in a sintering plant which agglomerates the fine iron ores to sinter. In a blast furnace, the next production step, the ore is reduced to hot metal by adding coke, coal, oil or gas. The hot metal is then refined in a converter to liquid steel by an oxygen top-blowing process, subsequently ready to be cast and rolled. Nowadays, post-treatment is generally applied after refining, as an increasing demand in terms of quality and the ongoing development and diversity of steel grades produced do not allow sufficient accuracy when melting the steel in the refining vessel. This so-called secondary metallurgy aims at enhancing the quality of the metal by precisely alloying and lowering undesired elements or compounds. After this process liquid steel with the desired grade is available for transforming through a continuous casting process to solid cuboid-shaped slabs. The slabs are customized (in terms of steel grade and dimensions) semi-finished products and are the starting product for further processing in different types of rolling mills. Moreover they are tradable products with a market existing between different steel companies. Hence there is a possibility to procure additional slabs in order to increase production capacity. The finished products which are produced through different rolling processes are classified according to the shape of their cross-sections as long products and flat products. The long products being produced in our reference production structure are sectional steel products and steel beams with different geometry and application areas. The flat products produced by two different hot and one cold rolling mill are consequently subdivided into a hot-rolled and a cold-rolled fraction. The hot wide strip as a hot-rolled finished product marks the input for the cold rolling mill and the spiral tube welding plant. Some of the cold strip gets further refinement in the metallic and non-metallic coating plants to obtain corrosion-resistant and surface protected steel strips and sheets.

As a result of the analysis of the reference production structure we have found the following conclusions:

- The produced slabs are make-to-order semi-finished products in terms of steel grade and dimensions.
- There are multiple finished products produced from slabs, themselves constituting raw material for different industries (car industry, building industry, etc.).
- The processing times of orders at certain resources depend on specific order characteristics (steel grade and dimensions) and are known (as an average) before start of production.
- The company cannot entirely set prices autonomously because of competition.

The problem arising for the company is which orders to accept and which orders to reject in order to maximize profit or contribution margin respectively. Therefore we see a practical need to develop a decision support system with the purpose of improving mid-term sales planning and primarily short-term sales planning in advanced planning concepts [7]. This decision support system ought to improve sales planning thereby explicitly considering interdependencies of different orders in terms of production capacity incorporating the (stochastic) demand to come and the contribution margin to reach with the accepted orders.

Due to the parallelism of the described setting and the setting in the "classical" revenue management approach, we are interested in whether the methods of classical revenue management approaches are applicable to the introduced decision making process.

2 Analysis of the Applicability of Revenue Management

Harris and Pinder [2], Klein [3], McGill and van Ryzin [5], Tscheulin and Lindenmeier [9], and Weatherford and Bodily [11] have identified several common characteristics of environments revenue management methods can be successfully applied to.

These characteristics can be subdivided into characteristics concerning the capacity and characteristics concerning the demand. We will give a brief introduction to these characteristics and will discuss if our problem satisfies these characteristics. Based on this we will discuss whether existing revenue management concepts are applicable to our problem.

2.1 Characteristics Concerning the Capacity

Characteristics that have been identified concerning the capacity in service operation environments are the fixed amount available, the high share of fixed costs and the perishability of capacity.

Fixed capacity

In all successful application areas of revenue management it is almost impossible to adjust the prevalent capacity to demand because of significant costs and/or a significant time lag to do so. This fixed capacity is often accompanied by stochastic demand, which makes balancing availability of products and fulfillment of demand difficult. Therefore it is important to optimally utilize this fixed capacity.

In the described problem setting we do obtain fixed capacity, too. But in addition to the hosted capacity we have the possibility to buy slabs in order to extend the capacity of the steel mill. Weatherford and Bodily [11, p. 832] argue that the presence of fixed capacity is not necessary to practice revenue management but variable capacity is significantly more complex to manage in this context. As an example for not entirely fixed capacity they mention the hotel industry where revenue management has been successfully implemented in practice. In this industry the point customers check out is stochastic therefore the capacity to be rented must be forecasted.

High fixed costs
In all successful applications of revenue management in the service industry there are relatively low variable costs for production, which allow a wide range of prices over which selling the service is better than letting the capacity be idle. In general this leads to high contribution margin for additional orders. The significant costs in this applications are fixed costs (e.g. the purchase of a car for a car rental company).

In the problem setting described we do obtain significant variable costs depending on the steel grade. This phenomena is due to the alloying of steel in accordance with the customer needs as prices for and ratios of alloying elements differ. That means that maximizing the expected revenue as a central objective of revenue management must be replaced by maximizing the expected contribution margin. Nevertheless revenue management in general seems to be practicable. Although the rewards of an effective revenue management are higher when high fixed costs are combined with high contribution margins [11].

Perishability of capacity
An additional characteristic of settings in which revenue management is established is the perishability of capacity. There is a certain date at which the product or service becomes available and after which it is either no longer available (typically for the service industry) or it depreciates (e.g. fashion goods). The product or service cannot be stored, unless accepting significant costs or the aging of the product. If the storage of the product is possible than one has to consider inventory control strategies in addition.

It is of course possible to store steel products but the make-to-order characteristic of the presented production structure does not permit a considerable production in advance. As a result manufacturing capacity specific to customer orders is perishable, too.

2.2 Characteristics Concerning the Demand

The characteristics that have been identified concerning the demand in service operation environments are the possibility of segmenting customers, the stochastic demand and the possibility of advance sales/bookings of capacity.

Possibility of segmenting customers
Revenue management is most effective when demand can be segmented and price sensitivity varies between market segments [2]. The variable or characteristic used to segment the market must truly differentiate the resulting products (fencing mechanism). The mechanisms generally used to segment customers in revenue management situations are the time of purchase and the quantity discounting practice [11].

In the described problem setting the segmentation of customers is not critical, because of the different and distinctive products. Nevertheless production of all relevant products employs the same resources during production and thus segments the capacity itself.

Stochastic demand
Fluctuations in demand create problems for efficient capacity management. In this situation revenue management is able to improve decisions of which orders to accept/reject [2].

The steel price determination in European Community has been investigated by Richardson [6]. He identifies three distribution channels for steel products: direct contract market (e.g. car industry), steel stockholder and a spot market. While the contract market accounts for only a small share of the total sales of each relevant product (about a quarter in the UK) and the stockholder market contracts are short for a large chunk of the market, the spot market price reflects more accurately the process of pricing in the market [6]. In times of high demand, spot prices move above contract prices; when demand is weak, the relationship reverses. As a result of our research and of the research of Richardson we notice a volatile market with respect to pricing of the different products and demand resulting.

Possibility of advance sales/bookings
The possibility of advance sales or bookings respectively is a key precondition for effectively applying revenue management in practice. This leads to one central issue of revenue management: How much capacity to reserve for late-arriving high-margin demand.

This is one of the main decisions to make in the described problem setting, too. But the identification of the most profitable orders is less straight forward, because of the multistage problem mentioned and the order-specific capacity utilization.

2.3 Conclusions

The application of revenue management concepts to the described make-to-order production system seems to be possible though more complex as compared to the service industry. The complexity is due to the explicit consideration of variable costs. Since variable costs are significant in the described setting an appropriate objective of a decision support system is the maximization of the (expected) contribution margin. To determine the contribution margin of potential orders the availability of a sophisticated cost accounting system is necessary. Furthermore the capacity utilization in the described problem setting is different from the "classical" areas of revenue management application. While in "classical" revenue management approaches the capacity utilization is always one unit, in the illustrated context capacity utilization depends on the order characteristics and therefore is a random parameter itself. Another challenge is the necessary forecasting of demand to come in an appropriate manner i.e. concerning different products, steel grade and dimensions. Further research on this topic will show if it is possible to aggregate the demand in order to allow a practical forecasting with respect to frame a sufficient decision support.

3 Mathematical Formulation

Our focus in this contribution is on effectively allocating capacity to a stochastic demand. Since no-shows and cancellations do not occur in our production environment we are not interested in overbooking strategies for the decision support system.

Our problem is a network problem where we are interested in applying existent revenue management methods of seat inventory control. In general we define a similar problem definition as Bertsimas and Popescu [1]. We host a network with e edges, the resources, which are used to produce p products. The initial resource capacity is given by a vector $\mathbf{N} = (N_1,\ldots, N_e)$ and the vector $\mathbf{n}$ denotes capacity left. The network is described by an $e \times p$ matrix $\mathbf{A}$, with entries a_{ij} the resource utilization of product j on resource i. In this first approach we do not consider resource utilization as a random parameter depending on the order characteristics. The column vector $\mathbf{A}^j$ indicates the capacity utilization of one product j at the resources. The p-vector $\mathbf{M} = (M_1,\ldots, M_p)$ denotes the contribution margin M_j gaining from selling a product j. The described problem is a problem with an infinite production time horizon that we decompose in intervals of constant time length with an appropriate interval of one week. Therefore we obtain a finite booking horizon problem lasting T periods in advance to production. The time line is furthermore discretized, so as to allow at most one request per time period. Time is counting backwards, i.e. $t = T$ is the beginning of our booking horizon and $t = 0$ is the end of the reservation period. The random vector of cumulative demand to come at time t is denoted by $\tilde{\mathbf{D}}^t$, that is, $\tilde{D}_j^t$ is a random variable representing the number of class j requests to come from time t until the end of the reservation period. The expected demand to come is denoted by the vector $\mathbf{D}^t = E[\tilde{\mathbf{D}}^t]$.

The decision to accept or reject an incoming order is a function of the current network configuration $\mathbf{n}$, the time-to-go t, the order specific contribution margin M_j and further information from demand forecasts. If we accept an order the state of our network becomes $(\mathbf{n}\text{-}\mathbf{A}^j, t\text{-}1)$, if we reject the order, only the time component of the state is changed into $t\text{-}1$.

The optimal policy for our network problem is provided by a stochastic dynamic programming model which can be found in Bertsimas and Popescu [1] and Talluri and van Ryzin [8]. It is optimal to accept an order when the corresponding contribution margin exceeds the opportunity cost of accepting the order which is calculated by the dynamic programming model. We are interested in methods that are applicable to our problem environment and therefore analyze integer and linear programming models as the deterministic analogues of the stochastic dynamic programming model. These models use only the expected demand information and are usually much easier to solve.

The integer programming model computes the optimal allocation $\mathbf{y}^*$ of available capacity $\mathbf{n}$ to the expected demand $\mathbf{D}^t$, by maximizing the total contribution margin subject to capacity and demand constraints.

For all $\mathbf{n} \leq \mathbf{N}$ and $t \leq T$, we have ($\mathbf{M}^T$ denotes the transpose of vector $\mathbf{M}$)

$$\begin{aligned} \mathrm{IP}\,(\mathbf{n}, \mathbf{D}^t) = {} & \max \mathbf{M}^T \mathbf{y} \\ \text{s.t. } & \mathbf{A}\,\mathbf{y} \leq \mathbf{n} \\ & \mathbf{y} \leq \mathbf{D}^t \\ & \mathbf{y} \geq \mathbf{0} \text{ and integer} \end{aligned}$$

The linear programming relaxation LP $(\mathbf{n}, \mathbf{D}^t)$ of this problem computes the fractional solution to our problem by relaxing the integrality constraints. The dual problem to the LP problem is given by

$$\begin{aligned} \mathrm{LPD}\,(\mathbf{n}, \mathbf{D}^t) = {} & \min \mathbf{n}^T \mathbf{v} + (\mathbf{D}^t)^T \mathbf{u} \\ \text{s.t. } & \mathbf{A}^T \mathbf{v} + \mathbf{u} \geq \mathbf{M} \\ & \mathbf{u}, \mathbf{v} \geq \mathbf{0} \end{aligned}$$

Where $\mathbf{v}$ determines the vector of shadow prices corresponding to the resources and $\mathbf{u}$ determines the vector of shadow prices corresponding to the demand.

The corresponding approximations of the opportunity cost OC of an order j are

$$\mathrm{OC}_j^{\mathrm{IP}}(\mathbf{n},t) = \mathrm{IP}(\mathbf{n},\mathbf{D}^{t-1}) - \mathrm{IP}(\mathbf{n} - \mathbf{A}^j, \mathbf{D}^{t-1})$$

$$\mathrm{OC}_j^{\mathrm{LP}}(\mathbf{n},t) = \mathrm{LP}(\mathbf{n},\mathbf{D}^{t-1}) - \mathrm{LP}(\mathbf{n} - \mathbf{A}^j, \mathbf{D}^{t-1})$$

$$\mathrm{OC}_j^{\mathrm{LPD}}(\mathbf{n},t) = (\mathbf{v}^{(\mathbf{n},t)})^T\, \mathbf{A}^j$$

Where the last one is the bid-price control which is a popular method in network revenue management [1], [8].

Example

Consider a simple network with three edges and four nodes. The edges represents a continuous caster and subsequently two hot rolling mills, supplied with slabs by the continuous caster. We are selling only two products: hot wide strip (p_1) and hot rolled heavy plate (p_2) with a contribution margin of $M_1 = 100$ and $M_2 = 90$, respectively. The capacity left at our resources $\mathbf{n}$ and the resource utilization matrix $\mathbf{A}$ are given by

$$\mathbf{n} = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} \qquad \mathbf{A} = \begin{pmatrix} 4 & 3 \\ 2 & 0 \\ 0 & 1 \end{pmatrix}$$

Suppose we are at *t* = *2* and the expected demand for hot wide strip to come is denoted by $D_1^2 = 3$ and for the hot rolled heavy plate by $D_2^2 = 3$. Obviously it is optimal to accept no order for the heavy plate and one order for the hot wide strip due to the capacity restriction at the caster. The optimal contribution margin to achieve is 100.
Let us assume that at *t* = *2* one order for heavy plate occurs and at *t* = *1* another one for hot wide strip. The corresponding approximations of the opportunity cost OC of this order pattern at *t* = *2* are

$$OC_2^{IP}((4,5,6)^T, 2) = IP((4,5,6)^T, \mathbf{D}^1) - IP((1,5,5)^T, \mathbf{D}^1) = 100 - 0 = 100$$

$$OC_2^{LP}((4,5,6)^T, 2) = LP((4,5,6)^T, \mathbf{D}^1) - LP((1,5,5)^T, \mathbf{D}^1) = 120 - 30 = 90$$

$$OC_2^{LPD}((4,5,6)^T, 2) = 30 \cdot 3 + 0 \cdot 0 + 0 \cdot 1 = 90$$

The LP Approximation and the bid-price control will accept the order and the IP approximation will reject the order, which is the optimal strategy. At *t* = *1* the IP approximation calculates the same opportunity cost and the order will be accepted. Therefore only the IP approximation finds the optimal solution in the described example.

4 Perspective

The presented practical problem has to be further investigated. In order to verify the mathematical formulation and results real world stochastic demand data has to be incorporated and the benefit of the implementation has to be pointed out. Moreover an extended model explicitly considering the resource utilization as a random parameter depending on the order characteristics has to be formulated.

References

1. Bertsimas, D., Popescu, I. (2003): Revenue Management in a Dynamic Network Environment. Transportation Science, 37, 257-277
2. Harris, F. H., Pinder, J.P. (1995): A Revenue Management Approach to Demand Management and Order Booking in Assemble-to-Order Manufacturing. Journal of Operations Management, 13, 299-309
3. Klein, R. (2001): Quantitative Methoden zur Erlösmaximierung in der Dienstleistungsproduktion. Betriebswirtschaftliche Forschung und Praxis, 53, 245-259
4. Kolisch, R. (2001): Make-to-Order Assembly Management, Berlin et al.
5. McGill, J., van Ryzin, G. (1999): Revenue Management: Research Overview and Prospects. Transportation Science, 33, 233-256

6. Richardson, P. K. (1998): Steel Price Determination in the European Community. Journal of Product & Brand Management, 7, 62-73
7. Stadtler, H., Kilger, C. (2000): Supply Chain Management and Advanced Planning, Berlin et al.
8. Talluri, K., van Ryzin, G. (1998): An Analysis of Bid-Price Controls for Network Revenue Management. Management Science, 44, 1577-1593
9. Tscheulin, D., Lindenmeier, J. (2003): Yield Management – Ein State-of-the-Art. Zeitschrift für Betriebswirtschaft, 73, 629-662
10. Verein Deutscher Eisenhüttenleute (1992): Steel Manual, Düsseldorf
11. Weatherford, L., Bodily, S. (1992): A Taxonomy and Research Overview of Perishable-Asset Revenue Management: Yield Management, Overbooking and Pricing. Operations Research, 40, 831-844

Hierarchical Multilevel Approaches of Forecast Combination

Silvia Riedel[1,2] and Bogdan Gabrys[2]

[1] Lufthansa Systems Berlin GmbH, Fritschstrasse 27-28, 10585 Berlin, Germany
[2] School of Design, Engineering & Computing, Bournemouth University
Poole House, Talbot Campus, Poole, BH12 5BB, United Kingdom

Abstract. In this paper the approach of combining predictions is used to benefit from the advantages of forecasts predicting on different levels, to reduce the risks of high noise terms on low level predictions and overgeneralization on higher levels. The presented experimentally compared approaches of combining seasonal airline demand forecasts differ concerning input decomposition, multilevel structures, combination models and kinds of aggregation. Significant forecast improvements have been obtained when using multilevel, hierarchical structures.

1 Introduction

A typical approach to building a forecast model consists of a phase of data analysis, determination of a level of forecasting and history building, model creation, determination of appropriate preprocessing, parameter calibration and validation. The model, once built, is updated on a regular basis by model rebuilding or updating the history based on the most recent data. But even if the calibration has been done well, it is likely that the real relationship between given inputs and the values to predict is so complex that it can not be modelled perfectly. The input information is restricted to noisy data measured only for the hopefully most important influencing features and interpreted only at the current subspace representing the level of history building. The restriction of history pool and other parameter settings as well as a lot of preprocessing measures finally lead to a loss of information, too. This problem is becoming even more relevant if the underlying processes and data change over time and the chosen settings are not optimal any more.

To overcome the above mentioned problems there is a necessity for an approach which is not only taking into account the available information in a sophisticated manner but which is also able to adapt automatically to new situations.

We first discuss combining approaches which enable the modelling of complex functional relationships based on different less complex ones, but which also offer possibilities for adaptive parameter settings and preprocessing. Then the problem of choosing the level of forecasting and history building is investigated. We propose to extend the use of combination approaches to the combination of multilevel predictions and discuss related tasks. We finally discuss experimental results which show a clear forecast improvement while using hierarchical multilevel combination strategies.

2 Combination Approaches

Forecast combination approaches are today a scientifically acknowledged procedure to model complex functional relationships in producing not one optimal forecast, but a number of forecasts and combining them for the final prediction. It can be considered as a black box where the inputs are individual forecasts and sometimes additional information and the output is a final forecast (see figure 1). The existing combination approaches are differing in the description of the functional relationship which represents the black box. Beside the *simple average model*, which gives the same weight to all individual forecasts, there are two common groups of linear combination models, in which the relationship is a simple weighted linear sum of the individual forecasts and individual forecast performance is taken into account to calculate the weights. While *rank based models* describe forecast performance based on ranks of past performance without interpreting statistical properties of forecast errors, *variance / covariance based models* and *ordinary least squares regression based models* use error variance and covariance information to represent forecast performance. (for details see [1] and [2])

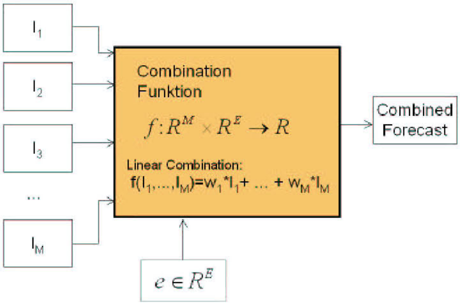


Fig. 1. Forecast Combination as a black box.

A more complex and flexible group of combination models are nonlinear combination models. There exist a wide variety of mostly application specific approaches. They differ in the selection of external input information as well as in the general approach. Typical nonlinear approaches include neural networks [3], (fuzzy) expert systems [4] or functional approaches.

At the beginning of the discussion related to combination approaches the scientists were surprised that combined forecasts outperformed individual forecasts. If it is possible to generate a combined forecast outperforming each individual forecast, it is also possible to model that relationship directly in one forecast model. So combination can be seen as a possibility to model complex functional relationships on the basis of different less complex ones.

Different approaches have been developed to explain the performance of the combined forecasts based on error variances and covariances of the individual forecasts [5]. It has been shown theoretically and experimentally that the best results can be achieved if the different individual forecasts are divers in the sense that they are able to provide diverse knowledge based on different input information in terms of different available sources of information, different preprocessing or history pools, different functional or stochastic approaches or different parametrization of the models. So combination can also be used in order to profit from different parameter settings or different views of historical data. It is even possible to use it to calibrate preprocessing like stabilization measures. Note that the advantage here is that the combined forecast is profiting not only from one optimal solution like one optimal parameter set, but from a whole set and that it is an adaptive procedure meaning that, e.g., a parameter set would automatically adapt to new situations.

3 Determination of the Appropriate Forecast Level

Real world forecasting problems are often not related to an isolated prediction, but to the task of predicting the future situation in an application defined space. For the application of revenue management forecasts this means a whole network of flights, routings or itineraries which again are containing different fareclasses, point of sales or others. A crucial problem is to determine appropriate levels on which forecasts should be produced, on which the models are calibrated or structural characteristics are determined. If a level is chosen too fine, there is a high risk of undesirable small numbers predictions in connection with large noise terms in the data. Important characteristics or relations may not or only badly be detected, which leads to unstable or structurally poor forecasts. If on the other hand the chosen level is too general, important characteristics related to special parts in the input space may be ignored. The task is getting even more difficult due to the fact that data is mostly composed by different independent components, which may be characterized by a completely different behavior regarding structure, correlation and noise characteristics. So it is likely that seasonal behavior should be modelled on another level than a schedule change.

In practice this problem is often resolved with trial and error approaches. If the level of calculation is determined based on static test data, the choice is rarely revised. This may become a problem when the behavior changes over time. So for instance it can be good to learn seasonal behavior for a certain booking class on the class level, but if suddenly the amount of bookings in this class is significantly reduced because a new competitor is on the market, the seasonal behavior should better be learned on the compartment level in the future. Automatic approaches to solve the problem should ideally have the following characteristics:

- The forecasts profit from specialists knowledge (fine level forecasts).
- The forecasts profit from general structural knowledge (higher levels).
- Fine level noise influence is reduced.
- The selection is done on decomposed data.
- The selection of the levels is adapting automatically to changes in the data.

4 Hierarchical Multilevel Combining Approaches

A similar situation compared to the choice of the functional approach, parameter settings or preprocessing can be expected related to the choice of the forecast level. Each forecast level is containing information based on which functional relationships can be determined, but it is likely that none of those models is optimal since it is not taking into account all available information. Low level forecasts are potentially missing general structure information. High level forecasts are not taking into account the special characteristics related to the concrete part of the input space, or the representation of these characteristics is contained in the forecast model in a completely different manner than having built the model directly on the finer level.

We propose to extend the use of combining models to generate forecasts which use really all available information, meaning also different representations from different levels, and which are also able to adapt automatically to new situations present at different levels. The following questions have been answered:

- How to determine low level predictions based on high level predictions taking into account that predictions produced on different levels can be characterized by different statistical properties?
- To represent a higher level, is it better to use a larger amount of low level data covering a bigger part of the space or is it better to aggregate data to higher subspaces?
- Should all forecasts including different levels as well as different methods be combined in one combining process or is it better to use multi step approaches?

4.1 Decomposition and Comparability between different Levels

The question of how to aggregate and split forecasts between levels has been extensively discussed. Our experience related to airline demand data is that summation of low level forecasts to a higher level produces quite good high level forecasts, but on the other hand the splitting of high level forecasts to lower levels performs poorly. We have tackled this problem by profiting from the decomposition approach. We propose to predict the general level of demand as a stable absolute representation of the different levels without taking into account multilevel approaches. All other components can be predicted in terms of a factor related to the general level, which can be transferred to other levels without any need of splitting the forecasts (see figure 2).

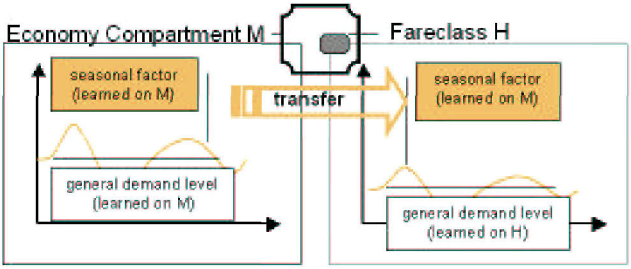
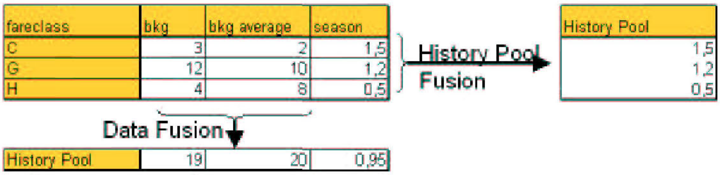

Economy Compartment M
Fareclass H
seasonal factor (learned on M)
transfer
seasonal factor (learned on M)
general demand level (learned on M)
general demand level (learned on H)

Fig. 2. Transferring forecasts between forecast levels taken as an example for the two levels: compartment M and fareclass H.

4.2 History Pool Fusion (HPF) versus Data Fusion (DF)

There are two options to take a wider range of information into account : the pool (number) of historical data used to learn a special behavior is extended (History Pool Fusion (HPF)) or the level, on which the input data and forecasts are generated, is changed (Data Fusion (DF)). Figure 3 illustrates how history pool fusion and data fusion of three fareclasses would work for seasonal factors (here calculated in dividing a booking value by a value representing the yearly average). Whereas HPF simply takes into account the seasonal factors of all three fareclass, DF produces the sum of the inputs and is calculating one factor on the higher level.



fareclass	bkg	bkg average	season
C	3	2	1,5
G	12	10	1,2
H	4	8	0,5

History Pool Fusion
Data Fusion

History Pool
1,5
1,2
0,5

History Pool	19	20	0,95

Fig. 3. History Pool Fusion versus Data Fusion illustrated at the basis of seasonal forecasts for three fareclasses.

4.3 Structures of Multilevel Combinations

When using multilevel forecasts a crucial problem is that the number of forecasts to combine increases significantly. what leads to a much higher risk of instabilities based on noisy data or overfitting problems. A possibility to handle that problem is not to combine all forecasts but to use multi step approaches combining a number of forecasts and giving the result to a next combination step. In our first approach (Method Step (MS)) we start combining the forecasts produced by the same method on different levels and then we combine the resulting multilevel forecasts. The second approach is a hierarchical top down approach (Hierarchy Step (HS)) meaning that first the forecasts of higher levels are combined and then the forecasts of lower levels, but using also the combined forecast from the higher level. Figure 4 illustrates the hierarchical approach.

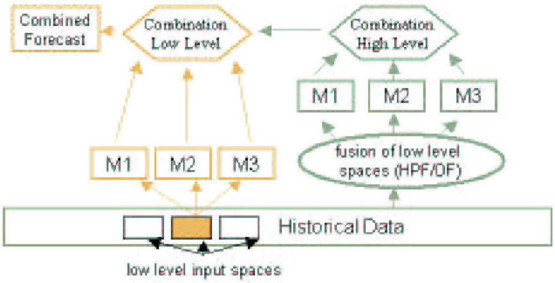
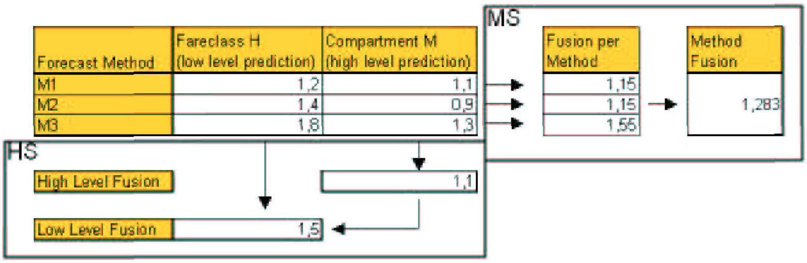


Fig. 4. Hierarchical multilevel forecast combination. M1 to M3 are representing different individual forecast methods.

Figure 5 compares the two approaches as an example. The matrix contains example predictions M1 to M3 of seasonal factors produced on two levels: the compartment level and the level of one fareclass of this compartment. In the MS approach (on the right side of the matrix), first all rows representing the individual methods are combined (simple average model). With the hierarchical approach HS (below the matrix), first the values of the second column (higher level) are combined, the second step is to combine the low level, including the result of the first combination.

5 Experiments

We have carried out a number of experiments to compare combinations based on decomposed versus not decomposed data, different structures, combination models and aggregations. The experiments have been restricted in the sense that up to now only two levels have been taken into account using three



Forecast Method	Fareclass H (low level prediction)	Compartment M (high level prediction)
M1	1,2	1,1
M2	1,4	0,9
M3	1,8	1,3

Fig. 5. Multi- step approaches: method step versus hierarchy step structures.

different forecast models to predict seasonal behavior of booking demand for Lufthansa on Itinerary Fareclass Point of Sale level. Five different combining models have been compared (for details see [2] and [6]):

- (av) : the simple average model
- (outp) : the outperformance model as an example of rank based models
- (opt) : the optimal model with assumption of independence as variance based model
- (sigm) : a nonlinear functional approach expecting that different methods work well for low seasons and different for high seasons. The parameters of the sigmoid functions used are learned using evolutionary strategies.
- (anfis) : the nonlinear neuro-fuzzy approach ANFIS

Table 1 shows the resulting percentage improvement in comparison to the model which is currently in use and which is already containing simple linear combinations. It can be seen that without decomposition it was not possible to achieve any improvements using combining strategies. Concerning the choice of the combination model the results show that improvements can be increased using nonlinear combination models, but that there are also more relevant losses based on a higher risk of instabilities. Another interesting result is that both, History Pool Fusion and Data Fusion, lead to improvements, but the use of both of them have had a detrimental effect on the forecast quality. One reason for it could be the fact that in this case the number of forecasts is highly increased, which could lead to instabilities during the combination process. The same reason can be responsible for the poor results if no multi step approaches are used. An indicator for this reasoning is the result of the simple average combination model which is stable per definition and does not show the same loss of improvement. Even if method steps seam better than no multi step approaches, the largest significant improvement can be seen in using hierarchy steps together with data fusion.

Table 1. Experimental results using different structural approaches and different combination models. The results are represented as an improvement percentage compared to the model which is currently in use.

decomposed	no	yes	yes	yes	yes	yes	yes	yes	yes	yes
fusion	no	HPF	HPF	HPF	DF	DF	DF	Both	Both	Both
structure	no	no	MS	HS	no	MS	HS	no	MS	HS
av	-5	-3	-3	-3	-4	-4	-4	-3	-3	-3
outp	-1	-10	+2	+3	-1	+3	+4	-9	-2	+1
opt	-2	-8	+2	+3	+1	+2	+5	-12	-1	-2
sigm	-22	-15	+3	+5	-3	+5	+9	-25	-4	+2
anfis	-19	-16	+4	+3	+1	+4	+6	-16	-3	+1

6 Conclusions and Future Work

We have seen that hierarchical multilevel combination models can be a powerful approach in order to build a high quality and adaptive forecast system. We have compared models differing concerning decomposition, multilevel structures and kinds of aggregation and have achieved an improvement of forecast quality up to 9 percent. The best results have been achieved using hierarchically aggregated data and multi step combinations. In this case forecasts were nonlinearly combined at each level of the hierarchy and the results from higher level were used as input to the lower level.

In the near future, encouraged by these results, more complex multilevel structures will also be investigated in relation to multi step approaches concerning the single level tasks of forecast combination.

References

1. Russell, T.D. and E.E. Adam (1987). An empirical evaluation of alternative forecast combinations. European Journal of Operations Research **33**, 1267-1276
2. De Menezes, Lilian M. et al. (2000). Review of guidelines for the use of combined forecasts. Management Science **120**, 190–204
3. Shan, M.S. et al. (1999). Improving the accuracy of nonlinear combined forecasting using neural networks. Expert Systems with Applications **16**, 49–54
4. Fiordaliso, A. (1998). A nonlinear forecasts combination method based on Takagi- Sugeno fuzzy systems. International Journal of Forecasting **14**, 367–379
5. Bunn, E.W. (1985). Statistical efficiency on the linear combination of forecasts. International Journal of Forecasting **1**, 151–163
6. Jang, J.-S.R. (1993). ANFIS: Adaptive Network- based Fuzzy Inference Systems. IEEE Transactions on Systems, Man and Cybernetics **23**, 665–685