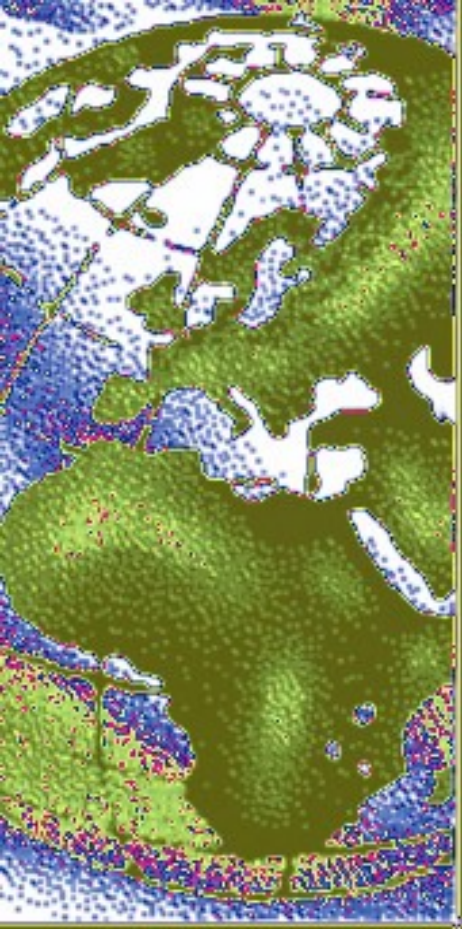


Martin Morlock,
Norbert Trautman
(Eds.)

Perspectiv
Operations

Essays in Honor

Martin Morlock, Christoph Schwindt, Norbert Trautmann,
Jürgen Zimmermann (Eds.)

Perspectives on Operations Research

GABLER EDITION WISSENSCHAFT

Martin Morlock, Christoph Schwindt,
Norbert Trautmann, Jürgen Zimmermann (Eds.)

Perspectives on Operations Research

Essays in Honor of Klaus Neumann

Deutscher Universitäts-Verlag

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

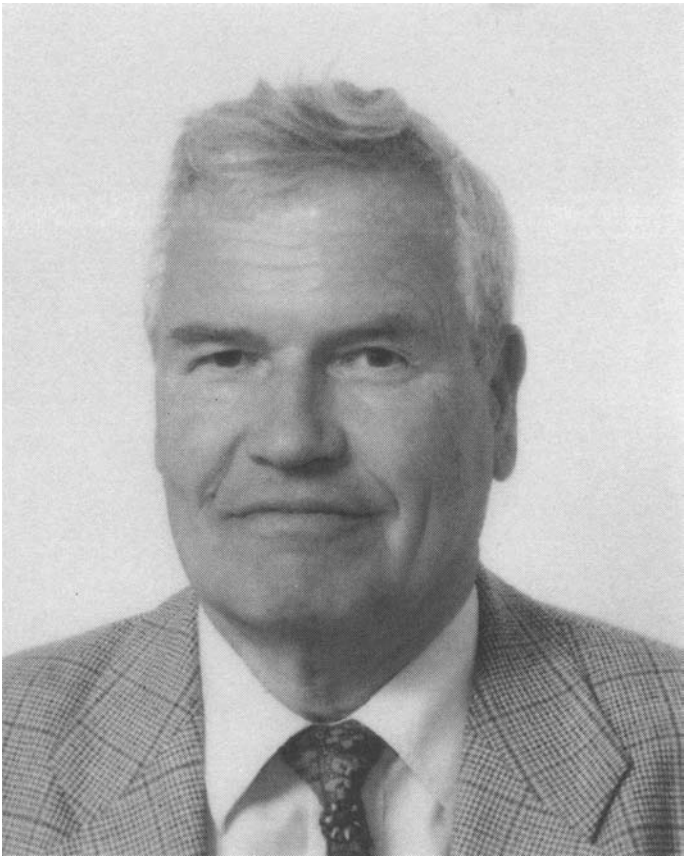
1. Auflage Februar 2006

Lektorat: Ute Wrasmann / Sabine Schöller

Der Deutsche Universitäts-Verlag ist ein Unternehmen von
Springer Science+Business Media.
www.duv.de

Umschlaggestaltung: Regine Zimmer, Dipl.-Designerin, Frankfurt/Main
Druck und Buchbinder: Rosch-Buch, Scheßlitz
Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier
Printed in Germany

ISBN 3-8350-0234-1



Preface

This collection of essays is dedicated to Professor Klaus Neumann, Head and Chair of the Institute for Economic Theory and Operations Research WiOR at the University of Karlsruhe. On the occasion of his emeritation, disciples, colleagues, scientific companions, and friends coming from different fields have contributed their perspectives on Operations Research to form a broad view on the discipline. The papers are organized in four parts on optimization, OR in production and service management, OR in logistics, and interdisciplinary approaches. We thank all the authors for their participation in publishing this volume. Mrs. Ute Wrasmann from Deutscher Universitäts-Verlag deserves credit for her interest and assistance on this project. Finally, we would like to express our gratitude to PTV Planung Transport Verkehr AG in Karlsruhe and to numerous former WiOR colleagues for their financial support.

Klaus Neumann was born in Liegnitz (Silesia) in 1937. From 1955 to 1961 he studied mathematics at the Technical Universities of Dresden and Munich. His first paper on analog computers and dynamic programming was published less than two years later. In 1964 he obtained a Ph.D. in mathematics under the supervision of Josef Heinhold in Munich. After a two-year stay in industry, he returned to his alma mater, working on the fields of dynamic optimization and control theory. In 1968 he was conferred the venia legendi for mathematics from the Technical University of Munich with a habilitation thesis on optimization subject to nonholonomous constraints. The same year he moved to the University of Karlsruhe, where he took up the head of the computer center. Since 1970 he is full professor of Operations Research at the School of Economics and Business Engineering in Karlsruhe.

Klaus Neumann has strongly influenced the development of Operations Research in Germany over more than four decades. For generations of German-speaking students his seminal trilogy *Operations-Research-Verfahren* has been the OR textbook of choice. His books on Operations Research and Production and Operations Management published in the 1990s remain a major reference in the field. Scientific monographs on dy-

namic programming (1969), control theory (1969), stochastic project networks (1979 and 1990), and project scheduling (2003) are evidences of his fruitful research, which has repeatedly been supported by the German Research Foundation DFG and by industry. The main achievements of this research are outlined in the first chapter of this book. From 1970 to date Klaus Neumann has supervised more than 30 doctoral and habilitation candidates. He held visiting professorships at the Universities of California at Berkeley and Riverside, Stanford, Florida, Waikato at Hamilton, Kunming, and Beijing Institute of Technology. Since 1972 he has been editor of several scientific series and journals like *Mathematical Systems in Economics*, *Methods of Operations Research*, and *Mathematical Methods of Operations Research*. In addition, he has been chairman and (and still is) member of the program committee of numerous scientific conferences such as EURO WG PMS, IEPM, IKM, or MISTA.

All over the years, students and colleagues at WiOR have not only benefited from Klaus Neumann's comprehensive scientific knowledge and expertise. We all have been influenced by his cultivated personality and generosity. Memorable excursions, wine tastings, and exquisite dinner receptions at his home in Conweiler have set very high cultural standards at our institute. We wish Klaus all the best for the future.

Gießen, Clausthal-Zellerfeld, Bern
November 2005

Martin Morlock
Christoph Schwindt
Norbert Trautmann
Jürgen Zimmermann

Contents

II Operations Research in Production and Service Management

III Operations Research in Logistics

IV Interdisciplinary Dimensions

Overview of Klaus Neumann's Research

Martin Morlock

Department of Economics
University of Gießen

Christoph Schwindt and Jürgen Zimmermann

Institute for Management and Economics
Clausthal University of Technology

Norbert Trautmann

Departement für Betriebswirtschaftslehre
University of Bern

1 Introduction

In this paper we give a short overview of the research conducted, initiated, and supervised by Klaus Neumann from the early sixties up to present. Of course, we do not claim exhaustiveness of our review. The major themes of research can be clustered into the three main areas sketched in Sections 2 to 4:

- Control Theory and Dynamic Programming (1960s and 1970s)
- GERT Networks (1970s to 1990s)
- Resource-Constrained Project Scheduling (since 1990s)

In any of those fields, Klaus Neumann has significantly influenced the development of OR in Germany and beyond. From the very beginning, his research has combined solid mathematical foundation and applicability of theoretical results. The relevance of his achievements to the treatment of real-world problems has been reflected in many applied research and development projects. A selection of the projects that have been carried out in cooperation with different industrial partners is sketched in Section 5.

2 Control Theory and Dynamic Programming

Among the various approaches existing at the beginning of the 1970's in quantitative economic science, only linear programming has been successful on a broad front. For this simply structured class of static optimization

problems, a commonly accepted and transparent model as well as efficient solution algorithms could be developed and applied due to the enormous advances in computer technology.

However, a multitude of practical problems in management and economics is not static in nature, but concern the analysis and optimal solution of time-dependent (decision) processes. Such problems are well-known as control problems (particularly in technology). To find an optimal solution to such problems, mainly two different approaches have been investigated: *control theory* and *dynamic programming.*

Control theory in continuous time is based substantially on an analytic approach referring to the *Pontrjagin maximum principle* and *transversality conditions.* Fundamental to dynamic programming is the so-called *Bellman optimality principle*, which was developed in the 1950's by the American mathematician Richard Bellman (cf. Neumann 1969a). In particular Neumann contributed several publications to the spreading of those two optimization techniques and to their application. Together with Bauer (1969), he was one of the first who explained in a very lucid way these two fundamental approaches and their relationship. For the acceptance and successful use of dynamic models, both their theoretical foundation and the development of numerical methods were essential. Major contributions to the latter topic, as well as descriptions of relevant applications, can be found for example in Neumann (1969a) and (1975a).

Initial considerations were concerned with the question whether analog or digital computers should be used for the numerical solution of dynamic optimization problems, especially for dynamic optimization problems in continuous time (cf. Neumann and Neumann 1963). Rapid progress in the digital computer technology soon decided in favor of the digital computers. In the following, research in the areas of control theory and dynamic programming concentrated on the development of solution procedures for different problems with a great diversity of applications and on their theoretical foundations (cf. e.g., Neumann 1965a, 1965b, 1968, 1969b, 1969c, 1970a, and 1971a). In addition, for applying the dynamic optimization principle, which represents a universally applicable instrument, large numerical problems had to be tackled. Dynamic programming mainly suffers from the curse of dimensionality. This means that the search process exploiting Bellman's optimality principle in higher-dimensional state and control spaces results in exponentially growing computational requirements. Some efficient procedures reducing the costs of computation by using approximating approaches and appropriately adapted gradient methods are, for example, presented in Neumann (1970b, 1975b).

A substantial strength of the Bellman optimality principle and its suc-

cessful use appears if the problem decomposes into many similar and interdependent sub-problems. These sub-problems are exposed to coincidental influences and the solution of the total problem can be built up from optimal solutions of the sub-problems. This is for instance the case for *Markov decision problems.* These problems belong to the field of stochastic dynamic programming and cover economic questions for which stochastic influences are relevant. The aspect of risk, connected with economic acting, plays a more and more important role in decision making (cf. Neumann and Morlock 2002).

Finally, a class of problems which are relevant to practice and for which stochastic dynamic programming proved suitable are known as *decision activity networks.* This is a very clear planning instrument for the representation and handling of stochastic network project control and scheduling problems, which, since the middle of the 1970's, are studied in numerous publications (cf. Neumann 1977a). In the following section, those networks are treated in more detail.

3 GERT Networks

Project planning, scheduling, and control are widely used in practice to accomplish outcomes under critical time constraints and given limited resources. Classical network techniques like CPM, MPM, or PERT are used for projects whose evolution in time can be uniquely specified in advance (cf. Neumann and Morlock 2002). Unfortunately, in practice this condition is frequently not fulfilled. Consider for instance an inspection that takes place during a production process and which reveals that a product does not conform to a set of given specifications. Thus it must be repaired or replaced, i.e., we have to return to a preceding stage of the production process. Since only a certain percentage of tested products does not comply with the specifications, this feedback loop occurs with a probability of less than one. To deal with these more general projects, whose evolution in time cannot be anticipated precisely (stochastic evolution structure of the project) and where feedback is permitted, so-called GERT networks with an activity-on-arc representation have been introduced (cf. Neumann 1971b, 1976, and 1977b).

The essential features of **GERT networks** as compared to CPM or PERT networks are more general arc weights, cycles to represent feedback, and six different types of nodes. These node types arise from combining three different *node entrances* corresponding to the logical operations "and", "inclusive-or", and "exclusive-or" as well as two possible *node exits*, which determine whether exactly one ("stochastic exit") or all ("determin-

istic exit") emanating activities must be performed if the corresponding node is activated. For each arc (activity) there is a conditional execution probability given that the corresponding initial event has occurred and a conditional distribution function for the duration of that activity given that the activity is carried out. For an in-depth treatment of the theory of GERT networks, we refer to Neumann and Steinhard (1979a) and Neumann (1989, 1990).

In CPM, MPM, or PERT network techniques, the **temporal analysis** of the project includes the determination of the earliest and latest start times of the project activities, the earliest and latest occurrence times of certain project events, as well as the computation of the project duration or its distribution. For GERT networks these concepts have been discussed by Neumann (1979a) and Neumann and Steinhard (1979a). However, in the case of GERT networks the meaning of those concepts is quite different because project events may occur several times and their computation is much more complicated. Therefore, the temporal analysis of GERT networks usually only considers quantities that are associated with the terminal events of the project such as the probability that certain terminal events will occur (a GERT network generally has more than one sink) and the respective (conditional) distribution function (cf. Neumann 1979b, 1990). For general GERT networks the temporal analysis is usually very time consuming because it requires the evaluation of multiple integrals (cf. Neumann 1984b). For special GERT networks such as so-called *EOR networks*[1] or *reducible GERT networks*[2] results from Markov renewal processes can be exploited for the temporal evaluation of the network, which simplifies the determination of the activation distributions (cf. Fix and Neumann 1979, Neumann and Steinhard 1979b, and Neumann 1985).

Besides the temporal analysis of stochastic projects, the **cost minimization** of such projects is of great interest. In the case of GERT networks different types of costs are incurred by the execution of activities and the occurrence of events. For EOR networks the cost minimization problem again leads to a Markov renewal decision process and can thus be modeled and solved as stochastic dynamic programming problem (see Neumann 1981, 1984a and Foulds and Neumann 1989). A different approach to solving the cost minimization problem, which leads to an optimal control problem, has been proposed by Delivorias et al. (1984).

If scarce resources (e.g., machines) are required for performing the proj-

[1] EOR networks are GERT networks whose nodes have an "exclusive-or" entrance.

[2] A GERT network is called reducible, if all nodes with "and" or "inclusive-or" entrances are part of special subnetworks which can be reduced to structures containing only "exclusive-or" nodes with a stochastic exit.

ect activities, so-called **GERT scheduling problems** have to be solved, whose type depends on the structure of the underlying production processes (cf. Neumann 1999). In particular single machine, parallel machine, flow shop, and job shop scheduling problems with GERT network precedence constraints arise in practical applications involving product variants. For single machine scheduling problems with stochastic precedence constraints a dynamic programming approach can be found in Neumann (1990). Polynomial algorithms for single machine scheduling problems with precedence constraints given by an EOR network are developed by Bücker et al. (1994). Heuristic procedures for parallel machine problems with GERT precedence constraints are discussed in Foulds et al. (1991) and Neumann and Zimmermann (1998). Neumann and Schneider (1999) deal with minimizing the expected makespan of flow shop and job shop scheduling problems with EOR network precedence constraints. A comprehensive summary on scheduling problems with GERT precedence constraints is given by Neumann (1990, 1999).

4 Resource-Constrained Project Scheduling

In this section we consider the planning of projects for which the evolution structure, activity durations, and resource data can be estimated in advance with sufficient accuracy. In this case we may consider the predictive data as being deterministic and take uncertainty into account by constructing robust plans or dynamically reacting on disruptions during the implementation. Project scheduling as part of project planning is concerned with computing time intervals for the execution of project activities in such a way that the precedence relationships between activities are satisfied and an objective function formulating the planning goal is minimized or maximized. In resource-constrained project scheduling, the latter problem amounts to allocating scarce resources over time to the execution of the activities. Different types of resources have been considered in the literature. The availability of *renewable resources* like personnel, machines, or equipment at a given time solely depends on the activities being in progress. Examples of *cumulative resources*, whose availability depends on the complete project history, are funds, materials, or storage space.

For what follows, we suppose that the execution modes defining the resource requirements of each activity have been fixed and that the activities must not be interrupted during their execution. A solution to such a single-mode scheduling problem is usually represented as a vector of activity start times, which is called a schedule. Furthermore, we assume that the precedence relationships between activities are given as *minimum and*

maximum time lags between the start times of activities. The activities and time lags can be modeled as an MPM network, possibly containing cycles. Minimum and maximum time lags allow to formulate many constraints arising in practical applications of project scheduling like release dates, deadlines, quarantine and shelf life times, or overlapping activities (see Franck et al. 1997 and Neumann and Schwindt 1997, 1998 for applications of project scheduling models in production planning). Minimum and maximum time lags greatly add to the complexity of resource-constrained scheduling problems since in difference to the case of ordinary precedence constraints, the problem of finding a feasible schedule is already NP-hard even if the project only contains renewable resources.

An **overview** of models and methods for project scheduling is given by Brucker et al. (1999), which also provides a three-field classification scheme for project scheduling problems. Many of the results on project scheduling in MPM networks mentioned in this section are presented in more detail in a review by Neumann et al. (2002b) and the monograph by Neumann et al. (2003a).

Exact and heuristic algorithms for project scheduling are based on the exploration of finite sets containing efficient points of the feasible region. The type of schedules to be investigated depends on the objective function under consideration. Based on a **structural analysis** of the feasible region, Neumann et al. (2000) have proposed a classification of objective functions and corresponding efficient points. The analysis shows that basically, efficient points can be enumerated in two alternative ways. If the temporal scheduling problem arising from deleting the resource constraints can be solved efficiently, the classical approach consists in using some relaxation-based generation scheme branching over alternatives to resolve resource conflicts. Examples of objective functions for which temporal scheduling can be done efficiently are the makespan (project duration) and the sum of discounted cash flows associated with the project activities (net present value of the project). If already the temporal scheduling problem is NP-hard, an optimal schedule can be computed with a constructive generation scheme, which iteratively establishes binding temporal or precedence constraints. Resource leveling problems, where the objective is to smooth the resource utilization over time, belong to this second class of problems.

For solving the **project duration problem** with renewable resources, both the constructive and the relaxation-based approache have been used. Priority-rule based methods exploiting the cyclic structure of the MPM project network have first been presented by Neumann and Zhan (1995) and Brinkmann and Neumann (1996). In Franck et al. (2001), the performance of different priority-rule based methods, local search procedures,

and truncated branch-and-bound algorithms based on resource relaxation have been compared with respect to accuracy and computation time. A branch-and-bound algorithm for the project duration problem with cumulative resources can be found in Neumann and Schwindt (2002).

Schedule-construction algorithms for the **net present value problem** have been devised by Neumann and Zimmermann (2000). A relaxation-based branch-and-bound algorithm for this problem has been developed by Neumann and Zimmermann (2002). In this algorithm, the temporal scheduling problems are solved by efficient primal and dual vertex-following algorithms.

Brinkmann and Neumann (1996) and Neumann and Zimmermann (1999, 2000) have treated several variants of the **resource leveling problem**. Depending on whether the maximum resource usage or the variability in resource utilization shall be minimized, different sets of tentative activity start times are investigated. According to the principle of the constructive schedule-generation scheme, the sets are chosen in way ensuring that in each iteration some temporal or precedence constraint becomes binding. Order-based neighborhoods for project scheduling problems with general nonregular objective functions like the net present value of resource leveling functions can be found in Neumann et al. (2003b).

5 Selected Applications

In what follows, we briefly discuss some selected applications of the research that has been described in the preceding sections. Together with further applications, they all are the result of applied research projects carried out in cooperation with partners from different industries.

A six-year research and development project building on the achievements in the field of resource-constrained project scheduling was concerned with **short-term production planning in the process industries**. In those industries, final products arise from several successive chemical or physical transformations of bulk goods, liquids, or gases processed on *processing units* such as reactors, heaters, or filters. Each transformation process may consume several input products and may produce several output products, whose amounts may be chosen within prescribed bounds. Perishable products must be consumed within a given shelf life time, which may be equal to zero. In addition, the storable intermediate products must be stocked in dedicated *storage facilities* like tanks or silos. Further peculiarities encountered in the process industries are cyclic product structures and sequence-dependent cleaning times on processing units.

For the case of batch production, Neumann et al. (2002a) present a new

solution approach, which can solve much larger practical problems than the methods known at this time. The new approach decomposes short-term planning for batch production into *batching* and *batch scheduling*. Batching converts the primary requirements for products into individual batches, where the objective is to minimize the resulting workload. The batching problem is formulated as a mixed-integer nonlinear program. The latter problem is transformed into a mixed-binary linear program of moderate size, which can be solved by standard MILP software. A solution to the batch scheduling problem allocates the batches to scarce resources such as processing units, workers, and intermediate storage facilities, where some regular objective function like the makespan is to be minimized. The batch scheduling problem is modeled as a resource-constrained project scheduling problem, which can be solved by new efficient truncated branch-and-bound or priority-rule based methods. The performance of the new solution procedures for batching and batch scheduling is demonstrated by solving several instances of a case study from process industries. Recently, the truncated branch-and-bound algorithm for the batch scheduling problem has been generalized to the case of continuous material flows (cf. Neumann et al. 2005).

Schwindt und Trautmann (2003) study a real-world scheduling problem arising in aluminium industry. They consider the production of *rolling ingots*, i.e., ingots of a certain aluminium alloy in rectangular form. These ingots are the starting material for the rolling of sheet, strip, and foil. It is shown how to model this scheduling problem as a resource-constrained project scheduling problem using minimum and maximum time lags between operations, different kinds of resources, and sequence-dependent changeovers. A solution procedure of type branch-and-bound is presented.

Now we turn to a project scheduling application from the area of **service operations management**. Car manufacturers increasingly organize visit programs for the customers that pick up their new cars at the factory. Such a program consists of a broad range of *event-marketing activities* and is designed to establish an emotional relationship between the customer and the brand. Mellentien et al. (2004) study the problem of scheduling all program activities of one day in such a way that the sum of the customers' *waiting times* during their visit is minimized. In service operations management, short customer waiting times are considered to be a key performance indicator of customer satisfaction.

Eventually, resource-constrained project scheduling has been applied to the problem of managing **research and development projects** in the pharmaceutical industries. Kolisch et al. (2003) study the problem of scheduling research activities in *drug development*. A particularity of this

problem is that the manpower requirements of the activities may vary over time, the requirement profiles being subject to decision.

In a current research project, quantitative methods for decision support in the **service industries** are developed. Schön-Peterson (2003) has developed various models and solution methods for the pricing of telecommunication services.

References

[1] Bauer H, Neumann K (1969) Berechnung optimaler Steuerungen. Lecture Notes in Operations Research and Mathematical Systems, Vol. 17. Springer, Berlin

[2] Brinkmann K, Neumann K (1996) Heuristic procedures for resource-constrained project scheduling with minimal and maximal time lags: The resource-levelling and minimum project-duration problems. Journal of Decision Systems 5:129–155

[3] Brucker P, Drexl A, Möhring R, Neumann K, Pesch E (1999) Resource-constrained project scheduling: Notation, classification, models and methods. European Journal of Operational Research 112:3–41

[4] Bücker M, Neumann K, Rubach T, (1994) Algorithms for single-machine scheduling with stochastic outtree precedence relations to minimize weighted expected flow-time or maximum expected lateness. Mathematical Methods of Operations Research 39:321–348

[5] Delivorias PN, Neumann K, Steinhardt U (1984) Gradient-projection and policy-iteration methods for solving optimization problems in STEOR networks. Zeitschrift für Operations Research 28:67–88

[6] Fix W, Neumann K (1979) Project scheduling by special GERT networks. Computing 23:299–308

[7] Foulds LR, Neumann K (1989) Introduction to temporal analysis, cost minimization, and scheduling of projects with stochastic evolution structure. Asia-Pacific Journal of Operational Research 6:167–191

[8] Foulds LR, Hoffmann D, Neumann K, (1991) Stochastic identical parallel machine scheduling with OR precedence constraints. Asia-Pacific Journal of Operational Research 8:1–25

[9] Franck B, Neumann K, Schwindt C (1997) A capacity-oriented hierarchical approach to single-item and small-batch production planning using project-scheduling methods. OR Spektrum 19:77–85

[10] Franck B, Neumann K, Schwindt C (2001) Truncated branch-and-bound, schedule construction, and schedule-improvement procedures for resource-constrained project scheduling. OR Spektrum 23:297–324

[11] Kolisch R, Meyer K, Mohr R, Schwindt C, Urmann M (2003) Ablaufplanung für die Leitstrukturoptimierung in der Pharmaforschung. Zeitschrift für Betriebswirtschaft 73:825–858

[12] Mellentien C, Schwindt C, Trautmann N (2004) Scheduling the factory pick-up of new cars. OR Spectrum 26:579–601

[13] Neumann K (1965a) Zur Theorie und Praxis der dynamischen Optimierung, Teil I. Unternehmensforschung 9:169–186

[14] Neumann K (1965b) Zur Theorie und Praxis der dynamischen Optimierung, Teil II. Unternehmensforschung 9:201–216

[15] Neumann K (1968) Die Existenz von Lösungen gewisser Optimalprobleme für lineare Funktionale. Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete 10:63–69

[16] Neumann K (1969a) Dynamische Optimierung. BI-Hochschultaschenbücher, Band 741/741a. Bibliographisches Institut, Mannheim

[17] Neumann K (1969b) Optimierung von Funktionalen unter nichtholonomen Nebenbedingungen, Teil I. Unternehmensforschung 13:37–57

[18] Neumann K (1969c) Optimierung von Funktionalen unter nichtholonomen Nebenbedingungen, Teil II. Unternehmensforschung 13:110–122

[19] Neumann K (1970a) Ein Verfahren zur Lösung gewisser nichtlinearer Kontrollprobleme. Computing 6:249–263

[20] Neumann K (1970b) Reduzierung des Rechenaufwandes bei der Behandlung sequentieller Entscheidungsprozesse. Operations-Research-Verfahren VIII:184–201

[21] Neumann K (1971a) Ein Existenzsatz für nichtlineare Steuerungsprobleme. Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete 19:172–179

[22] Neumann K (1971b) Decision activity networks. Methods of Operation Research 13:182–191

[23] Neumann K (1975a) Operations-Research-Verfahren, Band II: Dynamische Optimierung, Lagerhaltung, Simulation, Warteschlangen. Hanser, München

[24] Neumann K (1975b) Zur Lösung nichtlinearer Kontrollprobleme mit Hilfe eines Gradientenverfahrens. Computing 14:119–130

[25] Neumann K (1976) Evaluation of general decision activity networks. Operations-Research-Verfahren 25:268–275

[26] Neumann K (1977a) Optimal control of decision activity networks. In: Collatz L, Meinardus G, Wetterling W (eds.) Numerical Methods for Optimization Problems, Vol. 3. International Series of Numerical Mathematics, Vol. 36. Birkhäuser, Basel, pp. 191–212

[27] Neumann K (1977b) Structure and evaluation of decision activity networks. Zeitschrift für angewandte Mathematik und Mechanik 57:T 241–342

[28] Neumann K (1979a) Earliest and latest times in special GERT networks. Operations-Research-Verfahren 30:206–214

[29] Neumann K (1979b) Recent advances in temporal analysis of GERT networks. Zeitschrift für Operations Research 23:153–177

[30] Neumann K (1981) Optimal time-cost trade-offs by means of stochastic networks. Methods of Operations Research 42:5–16

[31] Neumann K (1984a) Precedence relation and stochastic evolution structure of general projects. Methods of Operations Research 48:399–422

[32] Neumann K (1984b) Recent developments in stochastic activity networks. Information Systems and Operational Research 22:219–248

[33] Neumann K (1985) EOR project networks. Computing 34:1–15

[34] Neumann K (1989) Scheduling of stochastic projects by means of GERT networks. In: Slowinski R, Węglarz J (eds.) Advances in Project Scheduling. Elsevier, Amsterdam, pp. 467–496

[35] Neumann K (1990) Stochastic Project Networks: Temporal Analysis, Scheduling, and Cost Minimization. Lecture Notes in Economics and Mathematical Systems, Vol. 344. Springer, Berlin

[36] Neumann K (1999) Scheduling of projects with stochastic evolution structure. In: Węglarz, J. (ed.) Project Scheduling: Recent Models, Algorithms and Applications. Kluwer, Boston, pp. 309–332

[37] Neumann K, Morlock M (2002) Operations Research. Hanser, München

[38] Neumann K, Neumann Th (1963) Die Anwendung des elektronischen Analogrechners auf das dynamische Optimieren. Unternehmensforschung 7:117–130

[39] Neumann K, Schneider WG (1999) Heuristic algorithms for job-shop scheduling problems with stochastic precedence constraints. Annals of Operations Research 92:45–63

[40] Neumann K, Schwindt C (1997) Activity-on-node networks with minimal and maximal time lags and their application to make-to-order production. OR Spektrum 19:205–217

[41] Neumann K, Schwindt C (1998) A capacitated hierarchical approach to make-to-order production. European Journal of Automation 32:397–413

[42] Neumann K, Schwindt C (2002) Project scheduling with inventory constraints. Mathematical Methods of Operations Research 56:513–533

[43] Neumann K, Steinhardt U (1979a) GERT networks and the time-oriented evaluation of projects. Lecture Notes in Economics and Mathematical Systems, Vol. 172. Springer, Berlin

[44] Neumann K, Steinhardt U (1979b) Time planning by means of GERT networks with basic elements. Operations-Research-Verfahren 30:111–129

[45] Neumann K, Zhan J (1995) Heuristics for the minimum project-duration problem with minimal and maximal time-lags under fixed resource constraints. Journal of Intelligent Manufacturing 6:145–154

[46] Neumann K, Zimmermann J (1998) Heuristic procedures for parallel-machine scheduling problems with stochastic precedence constraints. Annals of Operations Research 83:115–136

[47] Neumann K, Zimmermann J (1999) Resource levelling for projects with schedule-dependent time windows. European Journal of Operational Research 117:591–605

[48] Neumann K, Zimmermann J (2000) Procedures for resource levelling and net present value problems in project scheduling with general temporal and resource constraints. European Journal of Operational Research 127:425–443

[49] Neumann K, Zimmermann J (2002) Exact and truncated branch-and-bound procedures for resource-constrained project scheduling with discounted cash flows and general temporal constraints. Central European Journal of Operations Research 10:357–380

[50] Neumann K, Nübel H, Schwindt C (2000) Active and stable project scheduling. Mathematical Methods of Operations Research 52:441–465

[51] Neumann K, Schwindt C, Trautmann N (2002a) Advanced production scheduling for batch plants in process industries. OR Spectrum 24:251–279

[52] Neumann K, Schwindt C, Trautmann N (2005) Scheduling of continuous and discontinuous material flows with intermediate storage restrictions. European Journal of Operational Research 165:495–509

[53] Neumann K, Schwindt C, Zimmermann J (2002b) Recent results on resource-constrained project scheduling with time windows: Models, solution methods, and applications. Central European Journal of Operations Research 10:113–148

[54] Neumann K, Schwindt C, Zimmermann J (2003a) Project Scheduling with Time Windows and Scarce Resources. Springer, Berlin, 2003

[55] Neumann K, Schwindt C, Zimmermann J (2003b) Order-based neighborhoods for project scheduling with nonregular objective functions. European Journal of Operational Research 149:325–343

[56] Schön-Peterson C (2003) Preismanagement für Telekommunikationsdienstleistungen. Deutscher Universitäts-Verlag, Wiesbaden

[57] Schwindt C, Trautmann N (2003) Scheduling the production of rolling ingots: Industrial context, model, and solution method. International Transactions in Operational Research 10:547-563

Part I
Optimization

Matrices in Shop Scheduling Problems

Heidemarie Bräsel

Faculty of Mathematics
Otto-von-Guericke University Magdeburg

Es ist für mich eine ehrenvolle Aufgabe, einen Beitrag für dieses Buch einzubringen. Gleichzeitig ist es ein herzliches Dankeschön für Herrn Prof. Klaus Neumann für seine wissenschaftlichen Arbeiten, deren Ergebnisse ich sehr gern nutze, und für seine Unterstützung und sein stetes Interesse an der Entwicklung unserer Forschungsgruppe. Ich verbinde dies mit allen guten Wünschen für einen gesunden Ruhestand der Familie Neumann, der - dessen bin ich mir sicher - öfter auch in einen Unruhestand ausarten wird.

1 Introduction

In this paper shop scheduling problems are modeled by matrices. Initially we assume that each job is processed at most once on each machine. It is shown how the model can be extended to shop problems with more than one operation on each machine and to the case that preemption is allowed.

Modelling shop problems by matrices is a very natural approach of modelling such scheduling problems. At first it was presented by BRÄSEL [1]. The model is easy comprehensible and can be applied to simplify the description of algorithms in this field, for instance the block-approach idea for job shop problems and algorithms in the case of unit processing times.

Moreover, this model gives rise to new theoretical results. We give a brief review on such papers. The complexity question of some open shop problems with unit processing times was solved, see for instance BRÄSEL ET AL. [7], [8] and [9], TAUTENHAHN [16] and [17]. The insertion technique (cf. [1]) was developed for enumeration algorithms and beam search strategies, see for instance BRÄSEL ET AL. [10], WERNER AND WINKLER [18] and SOTSKOW ET AL. [15]. Theoretical results were obtained for counting problems, see BRÄSEL AND KLEINAU [5], HARBORTH [14] and BRÄSEL ET AL. [2] and [3]. Moreover, the model was applied for structural investigations of sequences and schedules: Shop scheduling spaces were characterizised algebraically by DHAMALA [12]. The irreducibility theory was developed, introduced by BRÄSEL AND KLEINAU [6]. Here especially the papers of BRÄSEL ET AL. [2], [3] and WILLENIUS [19] has to be mentioned. Furthermore, the software package LiSA works with this model succesfully.

However, there is no article in English to explain the basic model in detail. This paper closes this gap. It is organized like an introductory lecture on shop problems. We start with basic notations, give an overview on the used graphs and their description by matrices and present simple algorithms concerning the defined matrices. The insertion technique for construction of sequences is introduced and some properties of sequences are charakterized. We next show how the model can be modified for other classes of shop problems. Finally, the software package LiSA - A Library of Scheduling Algorithms is presented which contains the introduced matrices and their visualization as graphs and Gantt charts.

2 Basic Notations

In a *shop scheduling problem* a set of n *jobs* A_i, $i \in I = \{1, \ldots, n\}$, has to be processed on a set of m *machines* M_j, $j \in J = \{1, \ldots, m\}$, in a certain machine environment α under certain additional constraints β such that an objective function γ is optimized. Such a problem is called deterministic if all parameters are fixed and given in advance. Various optimization problems concerning allocation of restricted resources can be modeled as scheduling problems. We use the standard $\alpha \mid \beta \mid \gamma$ classification of deterministic scheduling problems developed by GRAHAM ET AL. [13].

At first we consider so-called *classical* shop problems, i.e., each job is processed on each machine at most once.

Processing of job A_i on machine M_j is called an *operation* (ij). PT denotes the matrix of processing times: $PT = [p_{ij}]$. The set of all operations SIJ is given by $SIJ = \{(ij) \mid p_{ij} > 0\}$. We assume that each job is processed on at most one machine at a time and each machine processes at most one job at a time. For certain shop problems, a *release time* $r_i \geq 0$, a *due date* $d_i \geq 0$ or a weight $w_i > 0$ for job A_i, $i \in I$, are requested. Let u_i and v_j be the number of operations for job A_i and on machine M_j, respectively. Then we define:

The *machine order of the job* A_i is the order of machines on which this job has to be processed: $M_{j_1} \rightarrow M_{j_2} \rightarrow \ldots \rightarrow M_{j_{u_i}}$.

The *job order on machine* M_j is the order of the jobs which this machine processes: $A_{i_1} \rightarrow A_{i_2} \rightarrow \ldots \rightarrow A_{i_{v_j}}$.

In a *job shop problem* ($\alpha = J$) the machine order of each job is given in advance. In a *flow shop problem* ($\alpha = F$) the machine orders of each job are the same , w.l.o.g. in the case of $SIJ = I \times J$: $M_1 \rightarrow M_2 \rightarrow \ldots \rightarrow M_n$. In an *open shop problem* ($\alpha = O$) both machine orders and job orders can be chosen arbitrarily. Other precedence constraints on the operations can be easily integrated into the model.

In a shop problem a combination of machine orders and job orders is to determine such that a time table of processing (schedule) can be constructed, which satisfies the additional constraints and minimizes the given objective function.

Let C_i be the completion time of job A_i. An objective function $\gamma = F(C_1, \ldots, C_n)$ is called *regular* if it has the following property:
If for two schedules S and S^* the inequality $C_i^* \geq C_i$ holds for all $i \in I$ then $F(C_1^*, \ldots, C_n^*) \geq F(C_1, \ldots, C_n)$ is satisfied.

The makespan C_{max}, the weighted sum of completion times $\sum w_i C_i$, the maximum lateness L_{max}, the weighted tardiness $\sum w_i T_i$ and the weighted number of late jobs $\sum w_i U_i$ are regular, where:
$C_{max} = \max\limits_{i \in I}\{C_i\}$, $L_{max} = \max\limits_{i \in I}\{d_i - C_i\}$, $\sum w_i T_i = \sum\limits_{i \in I} w_i \max\{0, d_i - C_i\}$
and $U_i = \begin{cases} 1, \text{ if } C_i > d_i \\ 0, \text{ otherwise} \end{cases}$ for all $i \in I$. Often $w_i = 1$ for all $i \in I$ holds.

3 Graphs and Matrices for Shop Problems

This chapter starts with a model of shop problems where preemption of the operations is not allowed.

3.1 Partial Orders and Schedules

We define the following digraphs where in each case the set of vertices is the set SIJ of operations:

- The digraph of machine orders $G(MO) = (SIJ, A_{mo})$ contains all arcs which describe the direct precedence constraints in all machine orders.
  $((ij),(kl)) \in A_{MO} \Longleftrightarrow \begin{cases} i = k \ \wedge \text{ after the processing of job } A_i \text{ on} \\ M_j \text{ job } A_i \text{ is processed on machine } M_l \end{cases}$
- The digraph of job orders $G(JO) = (SIJ, A_{JO})$ contains all arcs which describe the direct precedence constraints in all job orders.
  $((ij),(kl)) \in A_{JO} \Longleftrightarrow \begin{cases} j = l \ \wedge \text{ after the processing of job } A_i \text{ on} \\ \text{machine } M_j \text{ machine } M_j \text{ processes } A_k. \end{cases}$
- The digraph $G(MO, JO) = (SIJ, A)$ contains all arcs of $A = A_{MO} \cup A_{JO}$.

A combination (MO, JO) of machine orders and job orders is called *feasible*, if the corresponding digraph $G(MO, JO)$ does not contain a cycle. In this

case $G(MO, JO)$ is called a *sequence graph.* The described acyclic graphs are partial orders on the set of all operations.

Example 1 *Assume that three jobs have to be processed on four machines. The matrix PT of processing times is given by*

$$PT = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 2 & 3 & 4 & 3 \\ 1 & 5 & 1 & 2 \end{bmatrix}, \text{ therefore } SIJ = I \times J \backslash \{(13)\} \text{ holds.}$$

We consider the following machine orders and job orders:

$$\begin{array}{ll} A_1: & M_1 \to M_2 \to M_4 \\ A_2: & M_2 \to M_4 \to M_1 \to M_3 \\ A_3: & M_4 \to M_1 \to M_2 \to M_3 \end{array} \qquad \begin{array}{ll} M_1: & A_1 \to A_2 \to A_3 \\ M_2: & A_2 \to A_3 \to A_1 \\ M_3: & A_3 \to A_2 \\ M_4: & A_3 \to A_1 \to A_2 \end{array}$$

The corresponding digraph $G(MO, JO)$, see Figure 1, contains vertical *arcs, which represent job orders on the machines and* horizontal *arcs representing machine orders of the jobs.*

The combination of machine orders and job orders is not feasible because $G(MO, JO)$ contains the cycle $(12) \to (14) \to (24) \to (21) \to (31) \to (32) \to (12)$. *Since we have a cycle, there can not exist any schedule of processing.*

If we choose the natural order of the machines in each machine order and of the jobs in each job order, the digraph $G(MO, JO)$ cannot contain any cycle, because all arcs are directed to the left or downwards. In this case a corresponding schedule can easily be constructed.

Now we assign the weight p_{ij} to each vertex (ij) of the sequence graph $G(MO, JO)$. Then a schedule can be constructed. Usually schedules are described by the start times or the completion times of all operations. There exist the following classes of schedules:
A schedule is called a *non-delay* schedule, if no machine is kept idle when there exists a job available for processing.
A schedule is called *active*, if no operation can be completed earlier by changing the job orders without delaying any other operation.
A schedule is called *semiactive*, if no operation can be completed earlier without changing the job order on any of the machines.

Note, that each non-delay schedule is also active and each active schedule is also semiactive, but not vice versa.

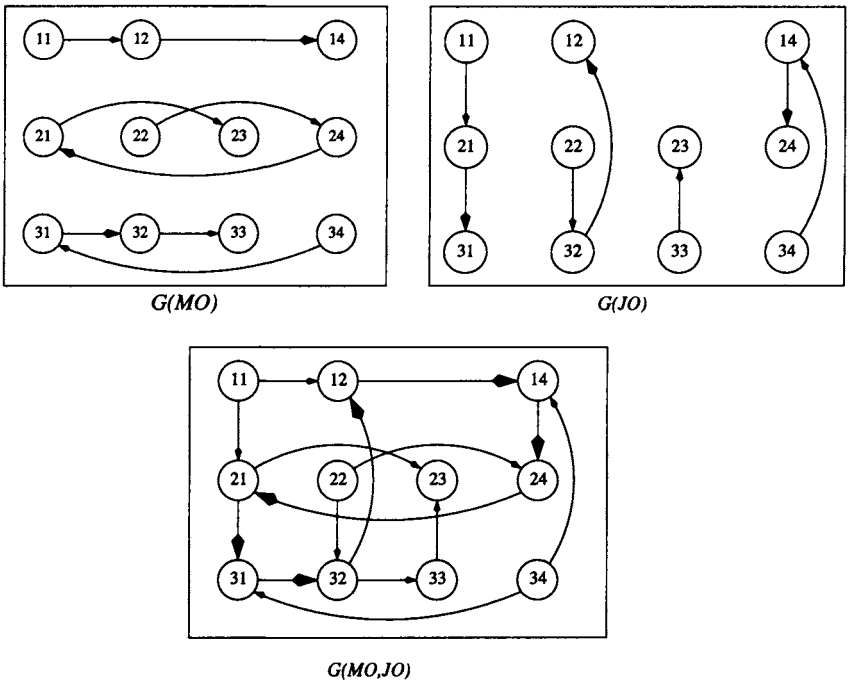


Figure 1: $G(MO)$, $G(JO)$ und $G(MO, JO)$ for Example 1

In the case of regular objective functions there always exists an optimal semiactive schedule and the computing of a longest path in $G(MO, JO)$ yields the makespan. We use the notation *longest path* with respect to the sum of the weights of the vertices contained in the path. Schedules are visualized by *Gantt charts*, which can be *machine oriented* or *job oriented.* In Figure 2 a job-oriented Gantt chart of a schedule with minimal makespan is given (see Example 1). There cannot be any better schedule because the longest job A_2 has no idle time within its processing.

In general, the set of schedules is infinite, but the set of sequences is finite. The binary relation R in the set of schedules:
"schedule 1 R schedule 2 if and only if both schedules have the same machine orders and job orders" is an equivalence relation. We can choose all semiactive schedules with unit processing times as representativs of the equivalence classes, whose number is finite.

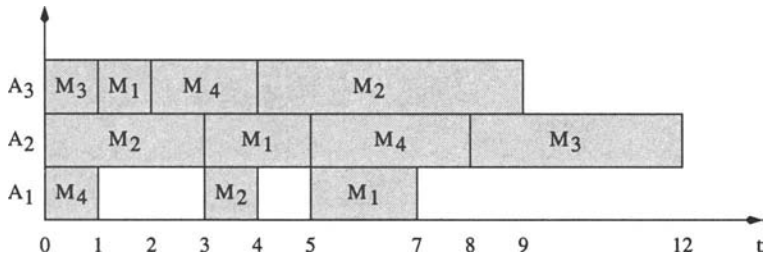


Figure 2: Job-oriented Gantt chart

3.2 Matrices in Shop Problems

In the literature the most commonly used model for shop problems is the well-known disjunctive graph model, see for instance BRUCKER [11]. We obtain the model used here by the following modifications:
- Cut the inserted source and sink and the corresponding incident arcs.
- Determine an acyclic orientation of the disjunctive graph.

We obtain the sequence graph (cf. Section 3.1) by deleting all transitive arcs which are not direct precedence constraints in the machine orders and in the job orders.

Now we define a set of matrices, where in each matrix an information of the operation (ij) on position (ij) is contained, this is the real advantage of the model. The digraphs $G(MO), G(JO)$ and $G(MO, JO)$ and in particular, the structure of the contained paths are visible by the matrices without drawing the digraphs. The number of vertices on a longest path with respect to unit weights of all vertices from a source to the vertex v is called *rank* of v: $rk(v)$. Now we define the following matrices:

- the *machine order matrix* $MO = [mo_{ij}]$: mo_{ij} is the rank of the operation $(ij) \in SIJ$ in the digraph $G(MO)$.
- the *job order matrix* $JO = [jo_{ij}]$: jo_{ij} is the rank of the operation $(ij) \in SIJ$ in the digraph $G(MO)$.
- the *sequence (matrix)* $PO = [po_{ij}]$: po_{ij} is the rank of the operation $(ij) \in SIJ$ in the sequence graph $G(MO, JO)$.

These matrices describe structural properties of a solution of a shop problem.

We extend this set by matrices with properties of the weighted sequence graph, i.e. the corresponding schedule (see Figure 3):

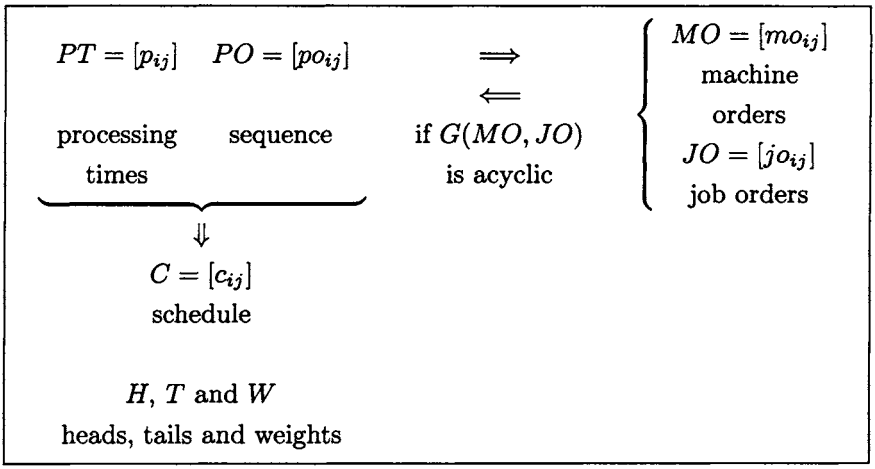


Figure 3: Matrices for shop problems

- the *completion time matrix* $C = [c_{ij}]$: c_{ij} is the completion time of the operation $(ij) \in SIJ$;

- the *head matrix* $H = [h_{ij}]$: the head h_{ij} is the lenght of the shortest time period which all predecessors of operation (ij) need for their processing, i.e. the weight of a longest path from a source to a direct predecessor of (ij) in the sequence graph;

- the *tail matrix* $T = [t_{ij}]$: the tail t_{ij} is the lenght of the shortest time period which all successors of (ij) need for their processing, i.e. the weight of a longest path from a direct successor of (ij) to a sink in the sequence graph;

- the matrix $W = H + PT + T$: the weight w_{ij} is the weight on a longest path across the operation (ij).

Clearly, the weight of a longest path can be obtained as maximal value in the completion time matrix C. By definition of W all operations with maximal value in W belong to at least one longest path.

Note, that in the case of a job shop problem with given matrix MO all sequences PO that contain the machine orders of MO are feasible.

3.3 Sequences, Schedules and Latin Rectangles

A *Latin rectangle* $LR[n,m,r]$ is a matrix with n rows, m columns and entries from the set $\{1,\ldots,r\}$, where each element from this set occurs at most once in each row and each column, respectively. If $n = m = r$ the matrix is denoted as latin square $LQ(n)$ of order n.

By definition of the rank of an operation, any sequence $PO = [p_{ij}]$ with $1 \leq po_{ij} \leq r$ has the following properties:
(a) Each entry k, $k \in \{1,\ldots,r\}$, is contained at most once in each row and in each column, respectively.
(b) For each $po_{ij} = k > 1$ the entry $k-1$ exists in row i or in column j.

Therefore, a sequence PO is a Latin rectangles, which satisfies in addition the so-called sequence condition (b). Now a feasible combination of MO and JO for given n and m can be easily constructed, if both matrices can be chosen arbitrarily or with repect to a given MO.

Example 2 *Consider a job shop problem with $n = 3$, $m = 4$ and the machine order matrix MO. If MO is a Latin rectangle, then a sequence is given by $PO = MO$. In the other case we determine a column where an (minimal) entry occurs more than once and construct a linear order on this operations, i.e., the corresponding arcs are inserted into $G(MO)$ and the rank matrix will be updated. The operations with fixed rank will be marked. This procedure is repeated until a Latin rectangle is obtained.*

$$MO = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 2 & 4 \\ 4 & 1 & 3 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 1* & 2* & \mathbf{3} & 4 \\ 2* & 4 & \mathbf{3} & 5 \\ 4 & 1* & \mathbf{3} & 2* \end{bmatrix} \quad PO = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 4 & 6 \\ 6 & 1 & 5 & 2 \end{bmatrix}$$

Note, that in the case $p_{ij} = 1$ for all operations the equality $PO = C$ holds for all shop problems without additional restrictions . Here the computing of an optimal schedule is equivalent to the construction of a certain Latin rectangle. Let us consider the problem $O \mid p_{ij} = 1 \mid C_{max}$. We have to compute a Latin rectangle $PO = C = LR[n,m,r]$ where r is minimal. Therefore, all schedules $C = LR[n,m,\max\{n,m\}]$ are optimal.

Such problems are related to edge coloring problems on graphs, which should be shortly described:

Each operation set SIJ of a shop problem can be assigned to a bipartite graph $G^b(SIJ) = (I \cup J, E)$ with $e = \{ij\} \in E$ iff $(ij) \in SIJ$. Therefore some open shop problems are equivalent to certain edge coloring problems on $G^b(SIJ)$:
(a) $O \mid p_{ij} = 1 \mid C_{max}$: Determine the chromatic index of $G^b(SIJ)$.

(b) $O \mid pmtn \mid C_{max}$: Solve a weighted edge coloring problem on $G^b(SIJ)$, where the edges are weighted by the processing times.
(c) $O \mid\mid C_{max}$: Solve an interval edge coloring problem on $G^b(SIJ)$, where the edges are weighted by the processing times.

3.4 Basic Algorithms in the Model

This section contains some basic algorithms for the modeling of shop problems by matrices. Note that the operations can be ordered in time $O(|SIJ|)$ by nondecreasing ranks: In a first loop we count the operations with rank k. Let n_k be the number of operations with rank k, $k = 1, \ldots, r$, where r is the maximal rank contained in PO. Next let $pointer(k) = \sum_{l=1}^{k-1} n_l + 1$ be the pointer to the first operation with rank k. Finally we sort in a second loop step by step the operations according to these pointers and update the applied pointer.

Algorithm 1 compute the sequence PO from given matrices MO and JO, if the combination (MO, JO) is feasible. The set MQ contains all operations, which are sources in both $G(MO)$ and $G(JO)$.

Algorithm 1: Computation of PO from MO and JO, if $G(MO, JO)$ is acyclic

Input: SIJ, MO and JO on the operation set SIJ;
Output: PO on the operation set SIJ, if $G(MO, JO)$ is acyclic;
BEGIN $k := 0$;
  REPEAT
    $k := k + 1$;
    Calculate $MQ = \{(ij) \in SIJ \mid mo_{ij} = jo_{ij} = 1\}$;
    IF $MQ = \emptyset$ **THEN** (MO, JO) is infeasible and **STOP**;
    FORALL $(ij) \in MQ$ **DO**
      BEGIN
        $po_{ij} := k$; Label row i in MO; Label column j in JO;
      END;
    $SIJ := SIJ \setminus MQ$;
    FORALL $(ij) \in SIJ$ in a labeled row in MO **DO**
      $mo_{ij} := mo_{ij} - 1$;
    FORALL $(ij) \in SIJ$ in a labeled column in JO **DO**
      $jo_{ij} := jo_{ij} - 1$;
    delete all labels;
  UNTIL $SIJ = \emptyset$;
END.

Algorithm 2 computes MO and JO from PO. Here a_i and b_j are the smallest integers, which are available for the rank of an operation of job A_i and of an operation on machine M_j, respectively. The maximal entry in sequence PO is denoted by r in the following algorithms.

Algorithm 2: Computing of MO and JO from PO

Input: r, I, J, SIJ, PO on the operation set SIJ;
Output: MO und JO on the operation set SIJ;
BEGIN
  FORALL $i \in I$ **DO** $a_i := 1$; **FORALL** $j \in J$ **DO** $b_j := 1$;
  FOR $k := 1$ **TO** r **DO**
    FORALL $(ij) \in SIJ$ with $po_{ij} = k$ **DO**
      BEGIN
        $mo_{ij} := a_i$ and $a_i := a_i + 1$;
        $jo_{ij} := b_j$ and $b_j := b_j + 1$;
      END;
END.

Algorithm 3 generates the corresponding semiactive schedule, i.e. the matrix C of completion times of all operations, from a corresponding pair PT and PO. Here r_i and $\overline{r}_j$ denote the currently smallest possible start times of operation (ij), i.e. for job A_i, and on machine M_j, respectively.

Algorithm 3: Computing of C from PT and PO

Input: r, I, J, SIJ, PT and PO on the operation set SIJ;
Output: C on the operation set SIJ.
BEGIN
  FORALL $i \in I$ **DO** $r_i := 0$; **FORALL** $j \in J$ **DO** $\overline{r}_j := 0$;
  FOR $k := 1$ **TO** r **DO**
    FORALL $(ij) \in SIJ$ with $po_{ij} = k$ **DO**
      BEGIN
        $c_{ij} := \max\{r_i, \overline{r}_j\} + p_{ij}$;
        $r_i := c_{ij}$; $\overline{r}_j := c_{ij}$;
      END;
END.

Algorithm 4 determines the matrices $H = [h_{ij}]$ and $T = [t_{ij}]$ from PT and PO.

Algorithm 4: Computing of H and T

Input: r, I, J, SIJ, PT and PO on the operation set SIJ;
Output: H and T on the operation set SIJ.
BEGIN
FORALL $i \in I$ **DO BEGIN** $r_i := 0$; $s_i := 0$; **END**;
FORALL $j \in J$ **DO BEGIN** $\overline{r}_j := 0$; $\overline{s}_j := 0$; **END**;
FOR $k := 1$ **TO** r **DO**
BEGIN
FORALL $(ij) \in SIJ$ with $po_{ij} = k$ **DO**
BEGIN
$h_{ij} := \max\{r_i, \overline{r}_j\}$; $r_i := h_{ij} + p_{ij}$; $\overline{r}_j := h_{ij} + p_{ij}$;
END;
FORAll $(ij) \in SIJ$ with $po_{ij} = r - k + 1$ **DO**
BEGIN
$t_{ij} := \max\{s_i, \overline{s}_j\}$; $s_i := t_{ij} + p_{ij}$; $\overline{s}_j := t_{ij} + p_{ij}$;
END;
END;
END.

We obtain the so-called reverse schedule S^{-1} to schedule S by reversing all arcs in the corresponding sequence graph. Clearly, $C_{max}(S) = C_{max}(S^{-1})$ is satisfied and the head of an operation (ij) in S is the tail of this operation in S^{-1} and vice versa. In Algorithm 4 $r_i, \overline{r}_j$ are again the earliest start times for job A_i and on machine M_j, respectively. $s_i, \overline{s}_j$ denote the earliest start times of job J_i and on machine M_j in the backwards calculation.

We close this section with an example illustrating the model.

Example 3 *Consider the matrix PT of processing times of Example 1, where due dates $d_1 = 6, d_2 = 12, d_3 = 8$ are given. The following combination of machine orders and job orders is feasible because the corresponding graph $G(MO, JO)$ is acyclic.*

Algorithm 1 computes the sequence PO and Algorithm 3 computes the schedule C. The matrices H and T are computed by Algorithm 4, therefore $W = H + PT + T$ can be computed. The longest path is unique: $(22) \rightarrow (21) \rightarrow (24) \rightarrow (23)$.
Schedule C yields $C_{max} = 12$ and $C_1 = 7$, $C_2 = 12$, $C_3 = 9$, therefore $\sum C_i = 28$, $L_{max} = 1$, $\sum T_i = 2$ and $\sum U_i = 1$ follow. Note, that this schedule is optimal in the open shop case with respect to the objective functions C_{max} and L_{max}. However, there exist better schedules with respect to $\sum C_i$, $\sum T_i$ and $\sum U_i$.

$A_1: \quad M_4 \to M_2 \to M_1$
$A_2: \quad M_2 \to M_1 \to M_4 \to M_3$
$A_3: \quad M_3 \to M_1 \to M_4 \to M_2$

$M_1: \quad A_3 \to A_2 \to A_1$
$M_2: \quad A_2 \to A_1 \to A_3$
$M_3: \quad A_3 \to A_2$
$M_4: \quad A_1 \to A_3 \to A_2$

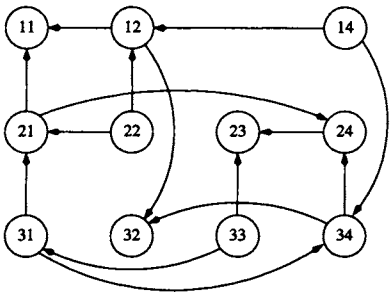


Figure 4: Machine orders, job orders and the sequence graph $G(MO, JO)$

We get the following information on operation (31)*: the processing time is 1, it is the second operation for job A_1 and it is the first operation on machine M_1, the completion time is 2, the head and the tail of* (31) *are 1 and 9, respectively. The longest path on this operation has weight 11.*

$$PT = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 2 & 3 & 4 & 3 \\ 1 & 5 & 1 & 2 \end{bmatrix} \qquad PO = \begin{bmatrix} 4 & 2 & - & 1 \\ 3 & 1 & 5 & 4 \\ 2 & 4 & 1 & 3 \end{bmatrix}$$

$$\Leftrightarrow \quad MO = \begin{bmatrix} 3 & 2 & - & 1 \\ 2 & 1 & 4 & 3 \\ 2 & 4 & 1 & 3 \end{bmatrix} \qquad JO = \begin{bmatrix} 3 & 2 & - & 1 \\ 2 & 1 & 2 & 3 \\ 1 & 3 & 1 & 2 \end{bmatrix}$$

$$C = \begin{bmatrix} 7 & 4 & - & 1 \\ 5 & 3 & 12 & 8 \\ 2 & 9 & 1 & 4 \end{bmatrix}$$

$$W = \begin{bmatrix} 5 & 3 & - & 0 \\ 3 & 0 & 8 & 5 \\ 1 & 4 & 0 & 2 \end{bmatrix} + \begin{bmatrix} 2 & 1 & - & 1 \\ 2 & 3 & 4 & 3 \\ 1 & 5 & 1 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 5 & - & 9 \\ 7 & 9 & 0 & 4 \\ 9 & 0 & 10 & 7 \end{bmatrix}$$

$$= \begin{bmatrix} 7 & 9 & - & 10 \\ 12 & 12 & 12 & 12 \\ 11 & 9 & 11 & 11 \end{bmatrix}$$

4 On Sequences in Shop Problems

In this section we discuss properties of sequences. We start with construction of sequences by the insertion idea, which is used in exact and heuristic algorithms for shop problems. It is shown in an example that we can exclude sequences because there is another sequence which has the same or a better objective function value for any choice of processing times .
The section closes with a distribution of open shop sequences to job shop classes.

4.1 Generating Sequences and Reducibility

Consider a sequence on the operation set $SIJ \subset I \times J$ and an operation $(ij) \in I \times J \setminus SIJ$. Now this operation is inserted into the given sequence graph with the following properties:
- a sequence on $SIJ \cup \{(ij)\}$ is generated and
- all precedence constraints of the old sequence graph are again contained in the new one.

In general, there are $u_i + 1$ choices to insert (ij) in the machine order of job A_i and $v_j + 1$ ways to insert (ij) in the job order on machine M_j, where u_i and v_j are the number of operations of job A_i and on machine M_j, respectively. In addition, there exists $(u_i + 1)(v_j + 1)$ possibilities of insertion, in each case Algorithm 1 can decide wether the corresponding graph $G(MO, JO)$ is acyclic. But because of the sequence condition we know, that for open shop problems in the cases listed below a new sequence is generated:

1. Set $rk(ij) = 1$, i.e., insert (ij) as source in the sequence graph, i.e. as direct predecessor of the first operation of job A_i and of the first operation on machine M_j, and update the ranks of all successors of (ij).

2. Set $rk(ij) = k + 1$, i.e., insert (ij) as direct successor of an operation of job A_i or of an operation on machine M_j with rank k and update the ranks of all successors of (ij).

3. If there are two operations (il) for job A_i and (kj) on machine M_j with $rk(il) = rk(kj)$ then insert (ij) as direct successor of one of the two and as direct predecessor of the other one and update the corresponding ranks. In both cases a cycle cannot be generated because there exist no path between both operations in the sequence graph.

For a job shop problem we have in addition to take into account the given machine order of job A_i, i.e., some of the described cases are omitted. Other given precedence constraints requires a further modification. We illustrate the insertion idea by an example:

Example 4 *Consider the following sequence PO on SIJ. Here we write* $po_{ij} = .$ *if* $(ij) \notin SIJ$. *Now we construct sequences by insertion of the operation* (12) *such that the properties above described are satisfied. There are at most 6 new sequences on* $SIJ \cup \{(12)\}$. *Case 1 yields* PO_1, *case 2 yields* PO_2 *and* PO_3 *and by case 3 we obtain* PO_4 *and* PO_5. *The updated ranks are marked. Note, that the last option yields a new sequence* (PO_6) *as well, because there is no path from* (32) *to* (13) *in the sequence graph of PO.*

$$PO = \begin{bmatrix} 1 & . & 2 \\ 3 & . & . \\ 2 & 1 & . \end{bmatrix},$$

$$PO_1 = \begin{bmatrix} \mathbf{2} & \mathbf{1} & \mathbf{3} \\ \mathbf{4} & . & . \\ \mathbf{3} & \mathbf{2} & . \end{bmatrix}, PO_2 = \begin{bmatrix} 1 & \mathbf{2} & \mathbf{3} \\ 3 & . & . \\ 2 & 1 & . \end{bmatrix}, PO_3 = \begin{bmatrix} 1 & \mathbf{3} & 2 \\ 3 & . & . \\ 2 & 1 & . \end{bmatrix},$$

$$PO_4 = \begin{bmatrix} 1 & \mathbf{2} & \mathbf{3} \\ \mathbf{5} & . & . \\ \mathbf{4} & \mathbf{3} & . \end{bmatrix}, PO_5 = \begin{bmatrix} 3 & \mathbf{2} & \mathbf{4} \\ \mathbf{5} & . & . \\ \mathbf{4} & 1 & . \end{bmatrix}, PO_6 = \begin{bmatrix} 1 & \mathbf{3} & 2 \\ \mathbf{6} & . & . \\ \mathbf{5} & \mathbf{4} & . \end{bmatrix}.$$

Note, that among the new sequences PO_3 is the best one in the case of makespan minimization. For each choice of processing times $C_{max}(PO_3) \leq C_{max}(PO_k)$ holds for all $k = 1, \ldots, 6$. The reason is the structure of all paths in the correponding sequence graphs. We call a path trivial, if it contains only operations of a single job or of a single machine, respectively. In the sequence graph of PO_3 there exists only one nontrivial path: $(32) \rightarrow (31) \rightarrow (21)$. But $(32) \rightarrow (31) \rightarrow (21)$ is also a path or a part of a path in any other sequence graph because it is contained in the sequence graph of PO. A sequence PO is *reducible* to a sequence PO^*, if for all matrices PT the inequality $C_{max}(PO^*) \leq C_{max}(PO)$ holds. Therefore all sequences are

of interest which are not reducible to another one, we call them *irreducible* sequences. Among all irreducible sequences there must be an optimal one. WILLENIUS presents in [19] results in the irreducibility theory with respect to other regular objective functions.

4.2 Distribution of Sequences in Shop Problems

Because of its relevance for practical applications the flow shop problem was investigated more in detail than other job shop cases. For flow shop problems, each combination of MO and JO is feasible, i.e., there exist $(n!)^m$ sequences. In BRÄSEL ET AL. [3] it is shown, that the matrix MO strong influences the properties of a job shop problem. We say that MO_1 is isomorph to MO_2 if MO_2 is obtained from MO_1 by permuting the job numbers and machine numbers. The number of sequences is the same in both cases. In the case of makespan minimization, the problem with given MO has the same properties as MO^{-1}, i.e. the reverse machine order. From the mathematical point of view, each isomorphism class has its own properties.

Example 5 *A shop problem with 3 jobs and 3 machines is given. Then there are* $(m!)^n(n!)^m = 46.656$ *combinations (MO,JO), 27.492 are infeasible and 19.164 are feasible.*

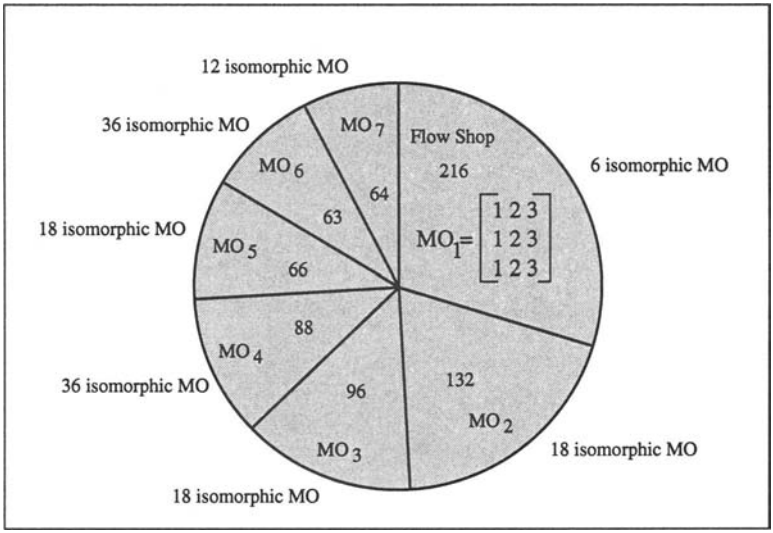


Figure 5: Number of sequences in different job shop problems

There exists 10 isomorphism classes with respect to the contained MO. If we consider makespan minimization, three of them can be deleted because MO is isomorphic to MO^{-1}. In [3] a formular for the number of isomorphism classes of the set of all machine order matrices is developed. Figure 5 shows the distribution of all sequences in the open shop to the different job-shops.

$$MO_2 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 3 & 2 \end{bmatrix}, MO_3 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix}, MO_4 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \\ 2 & 1 & 3 \end{bmatrix}$$

$$MO_5 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix}, MO_6 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \\ 3 & 1 & 2 \end{bmatrix}, MO_7 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$$

For this small parameter the number of sequences range between 63 and 216. Note, that in the case $(n, m) = (3, 4)$, the smallest number is 473 and the greatest number is 13.824. It is interesting, that the class MO_6 has the minimum number of sequences, although it is not a Latin square.

5 Generalization of the Model

The model can be generalized for other shop scheduling problems. First we consider shop problems, where preemption is allowed, cf. BRÄSEL AND HENNES [4].
Let $z_{ij} - 1$, $z_{ij} \geq 1$, be the number of preemptions of operation (ij). Then each operation of *SIJ* is splitted in an ordered set of split operations $\{(ij)_k \mid k = 1, \ldots, z_{ij}\}$, where the split operation $(ij)_k$ has to be processed before the split operation $(ij)_{k+1}$ can start, $k = 1, \ldots, z_{ij} - 1$. We denote the operation set with all split operation *pSIJ*. Now we define analogously to shop problems without *pmtn* the digraphs $G^p(MO) = (pSIJ, A_{MO})$, $G^p(JO) = (pSIJ, A_{JO})$ and $G^p(MO, JO) = (pSIJ, A)$ where $A = A_{MO} \cup A_{JO}$ holds.

- The digraph $G^p(MO) = (SIJ, A_{MO})$ contains all arcs which are given by the direct precedence constraints between two split operations for the same job A_i, $i = 1, \ldots, n$, i.e. in all machine orders.
- The digraph $G^p(JO) = (SIJ, A_{JO})$ contains all arcs which are given by the direct precedence constraints between two split operations on the same machine M_j, $j = 1, \ldots, m$, i.e. in all job orders.

- The digraph $G^p(MO, JO) = (pSIJ, A)$ contains all arcs of $A = A_{MO} \cup A_{JO}$.

We call $G^p(MO, JO)$ a preemptive sequence graph, if it is acyclic. Now we define the following "preemptive" matrices whose elements are ordered sets:

- the preemptive machine order matrix $pMO = [\{mo_{ij}^1, \ldots, mo_{ij}^{z_{ij}}\}]$, where mo_{ij}^k is the rank of the split operation $(ij)_k$ in $G^p(MO)$;

- the preemptive job order matrix $pJO = [\{jo_{ij}^1, \ldots, jo_{ij}^{z_{ij}}\}]$, where jo_{ij}^k is the rank of the split operation $(ij)_k$ in $G^p(JO)$;

- the preemptive sequence (matrix) $pPO = [\{po_{ij}^1, \ldots, po_{ij}^{z_{ij}}\}]$, where po_{ij}^k is the rank of the vertex $(ij)_k$ in the sequence graph $G^p(MO, JO)$.

To obtain a *preemptive schedule* pC we have to split each processing time p_{ij} in z_{ij} parts: $p_{ij} = \sum_{k=1}^{z_{ij}} p_{ij}^k$. Then we get:

- the preemptive processing time matrix $pPT = [\{p_{ij}^1, \ldots, p_{ij}^{z_{ij}}\}]$, where p_{ij}^k is the processing time of the split operation $(ij)_k$.

- the preemptive schedule $pC = [\{c_{ij}^1, \ldots, c_{ij}^{z_{ij}}\}]$, where c_{ij}^k is the completion time of the split operation $(ij)_k$.

The matrices H, T and W can also be modified to the preemptive case. Algorithm 1- 4 can be adopted easily, cf. BRÄSEL AND HENNES [4].
The model is also applicable for flow shop problems, job shop problems and mixed shop problems with preemption, since further precedence constraints can be easily integrated.

If we consider shop problems where a job has to be processed more than once on a machine, this model can also be applied. In this case the split matrix Z and the corresponding matrix pPT are given.

For shop problems where preemption is allowed we have no fixed splitting of the operations. Thus additionally we have to find a splitting of the operations. The section closes with an example.

Example 6 *In Figure 6 a preemptive sequence graph with 3 jobs and 3 machines is shown. We have* $z_{ij} = 2$ *for* $(ij) \in \{(21), (31), (32)\}$. *All other operations are not splitted.*

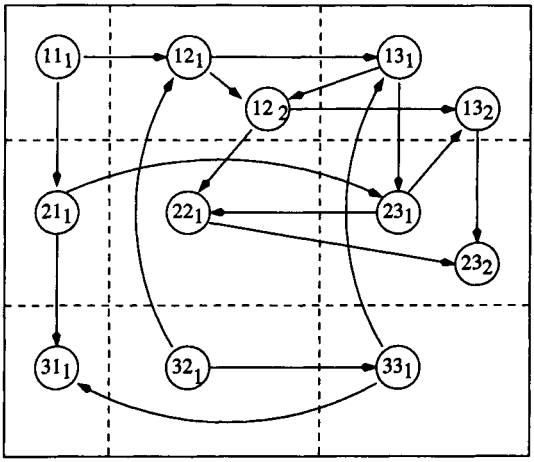


Figure 6: Preemptive sequence graph

The corresponding preemptive matrices are:

$$pMO = \begin{bmatrix} \{1\} & \{2,4\} & \{3,5\} \\ \{1\} & \{3\} & \{2,4\} \\ \{3\} & \{1\} & \{2\} \end{bmatrix}, pJO = \begin{bmatrix} \{1\} & \{2,3\} & \{2,4\} \\ \{2\} & \{4\} & \{3,5\} \\ \{3\} & \{1\} & \{1\} \end{bmatrix},$$

$$pPO = \begin{bmatrix} \{1\} & \{2,4\} & \{3,5\} \\ \{2\} & \{5\} & \{4,6\} \\ \{3\} & \{1\} & \{2\} \end{bmatrix}.$$

Using the preemptive processing time matrix pPT the preemptive schedule pC can be computed:

$$pPT = \begin{bmatrix} \{3\} & \{2,2\} & \{4,3\} \\ \{4\} & \{3\} & \{3,1\} \\ \{6\} & \{2\} & \{3\} \end{bmatrix}, pC = \begin{bmatrix} \{3\} & \{5,11\} & \{9,15\} \\ \{7\} & \{15\} & \{12,16\} \\ \{13\} & \{2\} & \{5\} \end{bmatrix}.$$

Again all objective functions $f(C_1, \ldots, C_n)$ can be easily computed from this schedule with $C_1 = 15$, $C_2 = 16$ and $C_3 = 13$.

The sequence pPO has the following properties, where r is the maximal rank in pPO:

(a) Each element of $\{1, \ldots, r\}$ occurs at most once in each row of PO and in each column of PO, respectively
(b) To any element $po_{ij}^k = l > 1$ there is an element $l-1$ in row i or column j of pPO (*sequence condition*).

6 LiSA: A Library of Scheduling Algorithms

The described model is applied in the software package LiSA - A Library of Scheduling Algorithms, which is developed for solving deterministic scheduling problems, especially for shop problems. All notations used in this paper are used in LiSA, too. We refer the interested reader to the LiSA-homepage: http://lisa.math.uni-magdeburg.de. The following example shows the using of LiSA. We consider an open shop problem with $m = 4$

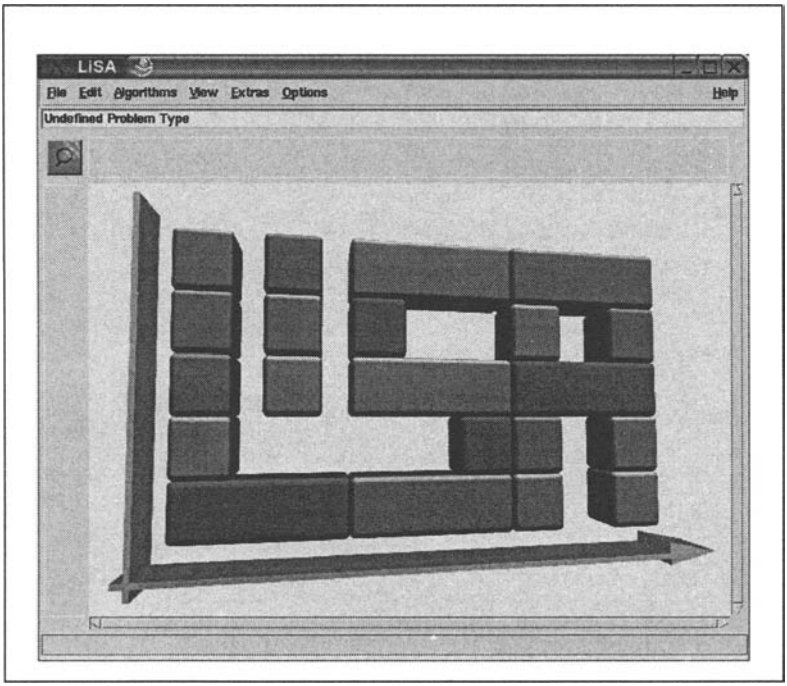


Figure 7: LiSA - A Library of Scheduling Algorithms

machines, $n = 4$ jobs and makespan minimization without any additional

constraints. The processing times are given by the following matrix PT:

$$PT = \begin{bmatrix} 12 & 6 & 15 & 7 \\ 13 & 6 & 7 & 13 \\ 3 & 14 & 8 & 7 \\ 10 & 13 & 9 & 7 \end{bmatrix}$$

What steps are necessary? Figure 8 shows a snapshot of the corresponding LiSA windows. After starting LiSA we choose the `New` button in the `File` menue. The problem type window is opened and we enter our problem in the $\alpha \mid \beta \mid \gamma$ denotation, namely $O \mid\mid C_{max}$. Moreover, enter $n = 4$ and $m = 4$. Now LiSA provides us with all modules for the considered problem. We start with the input of the processing times (buttons `Edit`, `Parameter`). We can do it by hand or by using the random generator. Note that all input data can also be read from a file.

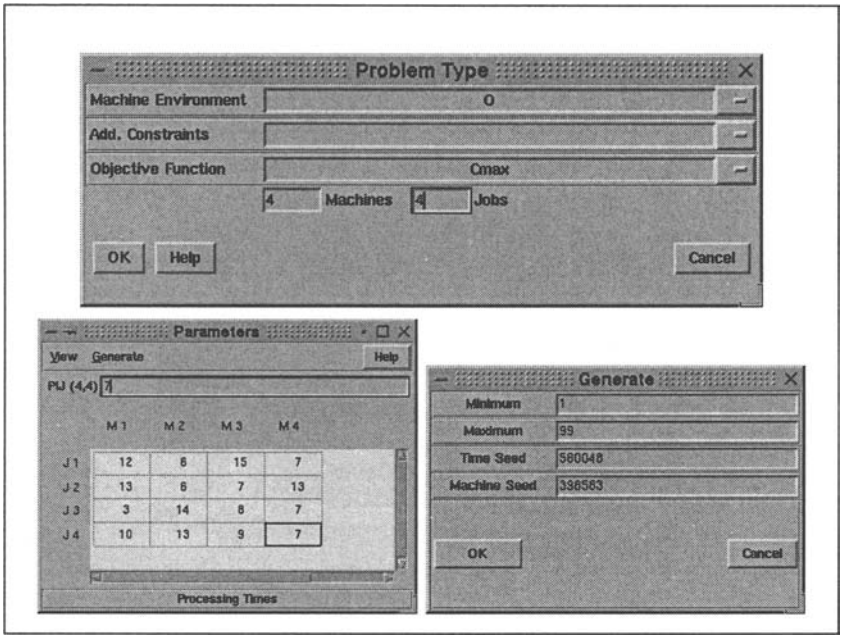


Figure 8: Input of the example

In the `Algorithms` menue exact and heuristic algorithms are available. Figure 9 shows some possibilities for the considered example. For instance,

the LPT-Rule (Longest Processing Time First) can be applied. After the construction of a first schedule we are able to use neighbourhood search algorithms, here simulated annealing with the 3_CR neighbourhood is chosen. In LiSA simulated annealing, iterative improvement, threshold acceptance and tabu search are available. Finally, matching heuristics can be used for the problem, here with minimal bottleneck objective function.

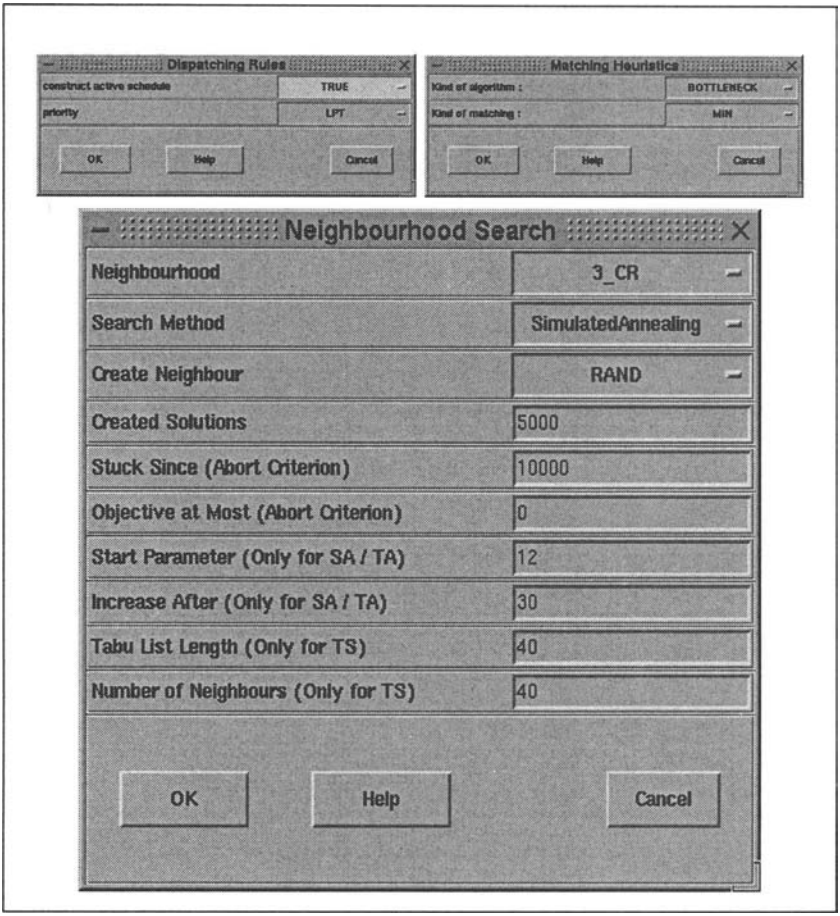


Figure 9: Heuristic algorithms for the considered problem

Usually a call of an algorithm produces the Gantt chart of the generated schedule , cf. Figure 11, but there are also other output options, cf. Figure 10. Here we can choose the sequence matrix or the sequence graph in the

`View` menue and the schedule described as the matrix of completion times or as a Gantt chart. Note, that under `Options` some options of the Gantt chart can be chosen: for instance, it can be machine oriented or job oriented or the critical path may be highlighted. If the number of jobs or machines is large, the Gantt chart is complex and the use of the zoom makes sense.

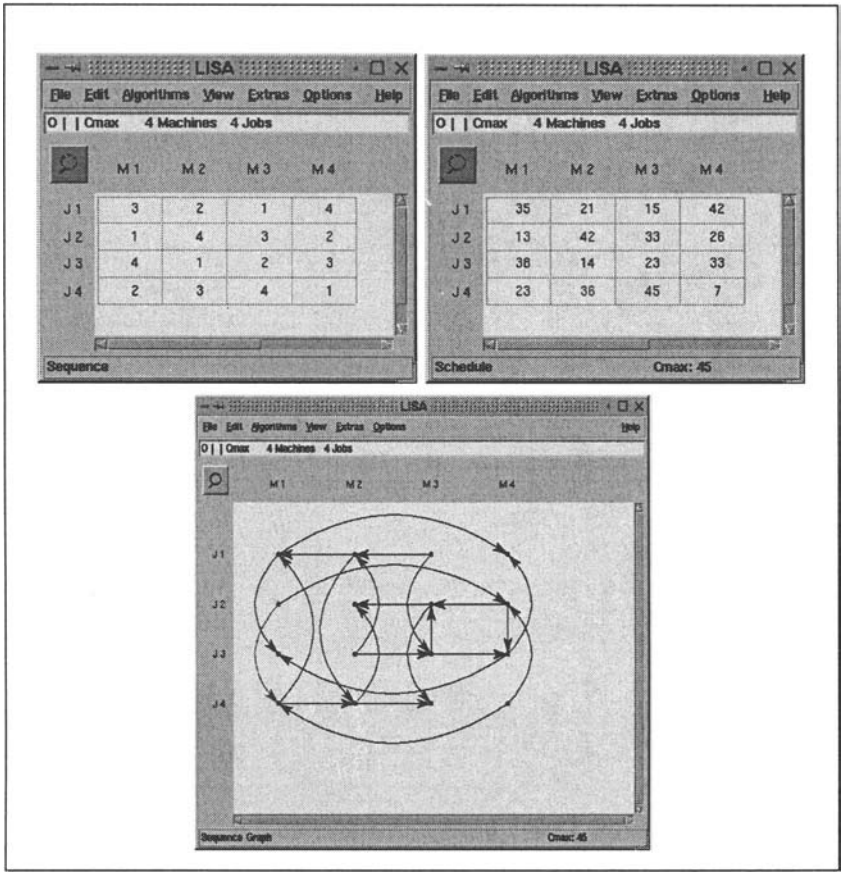


Figure 10: Output of LiSA: PO, C and G(MO, JO)

LiSA has some special modules, the most important one is the complexity module. Whenever LiSA has the $\alpha \mid \beta \mid \gamma$ notation of a problem, it determines the complexity status and gives, in addition, a complete reference. This module uses the data base of the scheduling group in Osnabrück. The software LiSA has a modular structure. The main part of LiSA con-

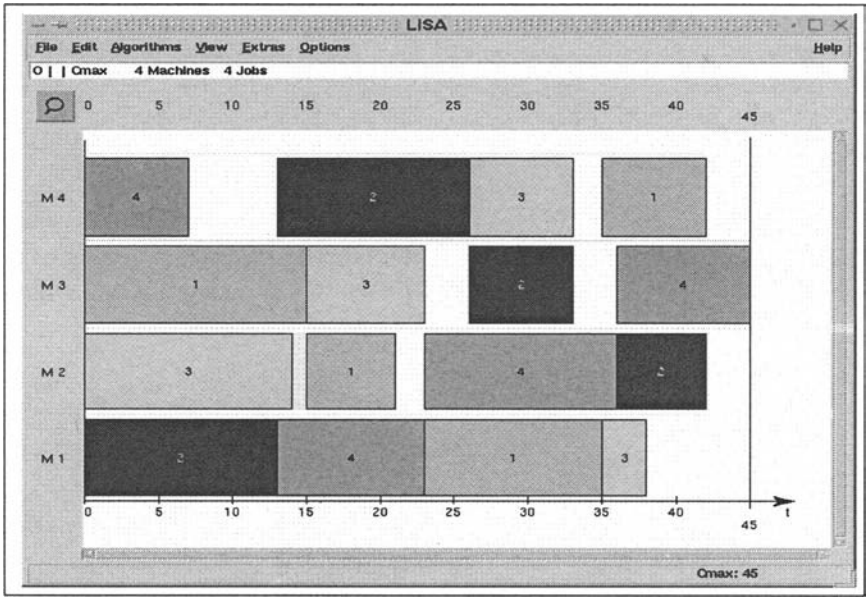


Figure 11: Output of LiSA: Gantt chart

tains all basic algorithms related to the model, the input and output procedures and the modules of the graphical user interface (GUI). Additionally, the program data is managed here and the work with external algorithms is coordinated.

In LiSA all algorithms used for solving scheduling problems are encapsulated in external modules. These modules are autonomous binaries with a common command line interface. They communicate with the main program via files. So they can be used both from within the LiSA GUI and independently without any GUI in a batch mode. The inclusion of new algorithms in LiSA is done by an algorithm description file and a corresponding help file in HTML format.

References

[1] Bräsel H (1990) Latin Rectangle in Scheduling Theory. Habilitation (in German), University Magdeburg, Germany

[2] Bräsel H, Harborth M, Tautenhahn T, Willenius P (1999) On the set of solutions of the open shop problem. Annals of Operations Research 92:241–263

[3] Bräsel H, Harborth M, Tautenhahn T, Willenius P (1999) On the hardness of the classical job shop problem. Annals of Operations Research 92:265–279

[4] Bräsel H, Hennes H (2004) On the open-shop problem with preemption and minimizing the average completion time. European Journal of Operational Research 157:607–619

[5] Bräsel H, Kleinau M (1992) On the number of feasible schedules of the open-shop-problem – An application of special Latin rectangles. Optimization 23:251–260

[6] Bräsel H, Kleinau M (1996) New steps in the amazing world of sequences and schedules. Mathematical Methods of Operations Research 43:195–214

[7] Bräsel H, Kluge D, Werner F (1994) A polynomial algorithm for the $n|m|O, t_{ij} = 1, tree|C_{max}$ open-shop problem. European Journal of Operational Research 72:125–134

[8] Bräsel H, Kluge D, Werner F (1995) A polynomial algorithm for an open shop problem with unit processing times and tree constraints. Discrete Applied Mathematics 59:11–21

[9] Bräsel H, Kluge D, Werner F (1996) Polynomial time algorithms for special open shop problems with precedence constraints and unit processing times. RAIRO Operations Research 30:65–79

[10] Bräsel H, Tautenhahn T, Werner F (1993) Constructive heuristic algorithms for the open shop problem. Computing 51:95–110

[11] Brucker P (2001) Scheduling Algorithms, Third Edition. Springer Verlag, Berlin, Heidelberg, New York

[12] Dhamala TN (2002) Shop Scheduling Solution-Spaces with Algebraic Characterizations. Dissertation, Shaker Verlag, Mathematics

[13] Graham RE, Lawler EL, Lenstra JK, Rinnooy Kan AHG (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann. Discrete Math. 4:287–326

[14] Harborth M (1999) Structural Analysis of Shop Scheduling Problems: Counting Problems, Potential Optimality and New Enumeration Algorithms. Dissertation (in German), GCA-Verlag, Forschen und Wissen: Mathematik

[15] Sotskow Y, Tautenhahn T, Werner F (1999) On the application of insertion techniques for job shop problems with setup times. RAIRO 33(2):209–245

[16] Tautenhahn T (1993) Open-Shop Problems with Unit Processing Times. Dissertation (in German), University Magdeburg

[17] Tautenhahn T (1996) On unit-time open shops with additional restrictions. ZOR 43(2):215–231

[18] Werner F, Winkler A (1995) Insertion techniques for the heuristic solution of the job shop problem. Discrete Applied Mathematics 58(2):191–211

[19] Willenius P (2000) Irreducibility Theory for Shop Scheduling Problems. Dissertation (in German), Shaker Verlag, Mathematics

Saddle Points from the Geometric Point of View

Wolfgang Eichhorn

Institut für Wirtschaftstheorie und Operations Research
Universität Karlsruhe (TH)

Nearly everybody can tell us what a *saddle* really is. Most of those who teach or study (a) mathematics, (b) mathematical economics, (c) econometrics, or (d) operations research are able to define the term *saddle point*—verbally or by means of equations and/or inequalities. But only very few of them are able or willing to draw, on a piece of paper or in a paper to be published, a surface in $\mathbb{R}^3$ that contains a saddle point.

Let us have a look into the literature on (a)–(d) dealing with saddles and saddle points. In many cases we do not find any figures showing saddle surfaces. Where there are such figures they sometimes look pretty from the geometric point of view, but very often have nothing or nearly nothing to do with the mathematical context or problem actually being treated.

This critical statement can be enhanced when we look into the literature on constrained optimization problems. There the general problem is usually formulated as follows:

$$\begin{aligned}&\text{Minimize } F(\boldsymbol{x}) \text{ under the constraints } \boldsymbol{G}(\boldsymbol{x}) \leq \boldsymbol{0},\\ &\text{where } F : \mathbb{R}^n \to \mathbb{R},\ \boldsymbol{G} : \mathbb{R}^n \to \mathbb{R}^m \text{ (the domains of}\\ &\text{definition of functions } F \text{ and } \boldsymbol{G} \text{ may be only parts of } \mathbb{R}^n),\\ &\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n,\ \boldsymbol{G}(\boldsymbol{x}) = (G_1(\boldsymbol{x}) \ldots, G_m(\boldsymbol{x})) \in \mathbb{R}^m.\end{aligned} \tag{1}$$

It is well known that we can formulate conditions, either necessary or necessary and sufficient, for a vector $\boldsymbol{x}$ to solve this problem. These conditions can be stated with aid of Lagrange functions. As we know, these are functions $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ defined by

$$\begin{aligned}L(\boldsymbol{x}, \boldsymbol{u}) &= F(\boldsymbol{x}) + \boldsymbol{u} \cdot \boldsymbol{G}(\boldsymbol{x})\\ &= F(\boldsymbol{x}) + u_1 G_1(\boldsymbol{x}) + \cdots + u_m G_m(\boldsymbol{x}),\\ &\text{where } \boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n,\ \boldsymbol{u} = (u_1, \ldots, u_m) \in \mathbb{R}^m\\ &\text{and } u_1, \ldots, u_m \text{ are the } \textit{Lagrange multipliers}.\end{aligned} \tag{2}$$

Global saddle points of the Lagrange function can help us solve global minimum problems (1). Global saddle points are points $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}})$ in the

$(n+m)$-dimensional space $\mathbb{R}^n \times \mathbb{R}^m$ such that

$$L(\hat{\boldsymbol{x}}, \boldsymbol{u}) \leq L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) \leq L(\boldsymbol{x}, \hat{\boldsymbol{u}}) \text{ for all } \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{u} \in \mathbb{R}^m. \tag{3}$$

For $n = m = 1$, *saddle surfaces* $\{(x, u, \ell) \mid x \in \mathbb{R}, u \in \mathbb{R}, \ell = L(x, u) \in \mathbb{R}\}$ in many cases can easily be drawn. They contain the points $(\hat{x}, \hat{u}, \hat{\ell}) = (\hat{x}, \hat{u}, L(\hat{x}, \hat{u}))$ as very important points of the saddles.

Notice that, because of the special form (2) of the Lagrange functions, the functions

$$\boldsymbol{u} \mapsto L(\boldsymbol{x}, \boldsymbol{u}) = F(\boldsymbol{x}) + \boldsymbol{u} \cdot \boldsymbol{G}(\boldsymbol{x}) \tag{4}$$

are *affine* for all $\boldsymbol{x}$. Specially for $\boldsymbol{x} = \hat{\boldsymbol{x}}$, by (3), $\boldsymbol{u} \mapsto L(\hat{\boldsymbol{x}}, \boldsymbol{u})$ has to have a global minimum at $\hat{\boldsymbol{u}}$, so, being affine, it has to be constant:

$$L(\hat{\boldsymbol{x}}, \boldsymbol{u}) = c \quad \text{(constant)}.$$

People interested in problems on constrained optimization would highly appreciate papers, monographs or books on these problems and their solutions that do not only introduce the term saddle point but also illustrate both saddles and saddle points from the geometric point of view. To state it in other words: Chapters on constrained optimization in papers, monographs, books or text books should in any case satisfy the following conditions C.1–C.4.

C.1 They contain one or more figure(s) showing saddle point(s) in a saddle surface.

C.2 Each of the figures shows, within an (x, u, ℓ)-coordinate system, a saddle surface containing saddle point(s) such that the saddle point condition (3) in case $n = m = 1$ is precisely illustrated by the figure.

C.3 Each figure shows a saddle surface that is, because of (4), built up by straight lines.

C.4 At least one figure shows exactly one saddle point in a saddle surface that is built up by *non-parallel* straight lines.

Up to now the author did not succeed in finding, in the relevant literature, any place satisfying conditions C.1–C.4. Dear Klaus, my congratulations and compliments include: Concerning conditions C.1–C.4 *your* relevant publications are closer to satisfaction than all the others I know.

As an attempt at satisfying conditions C.1–C.4 see the figure.

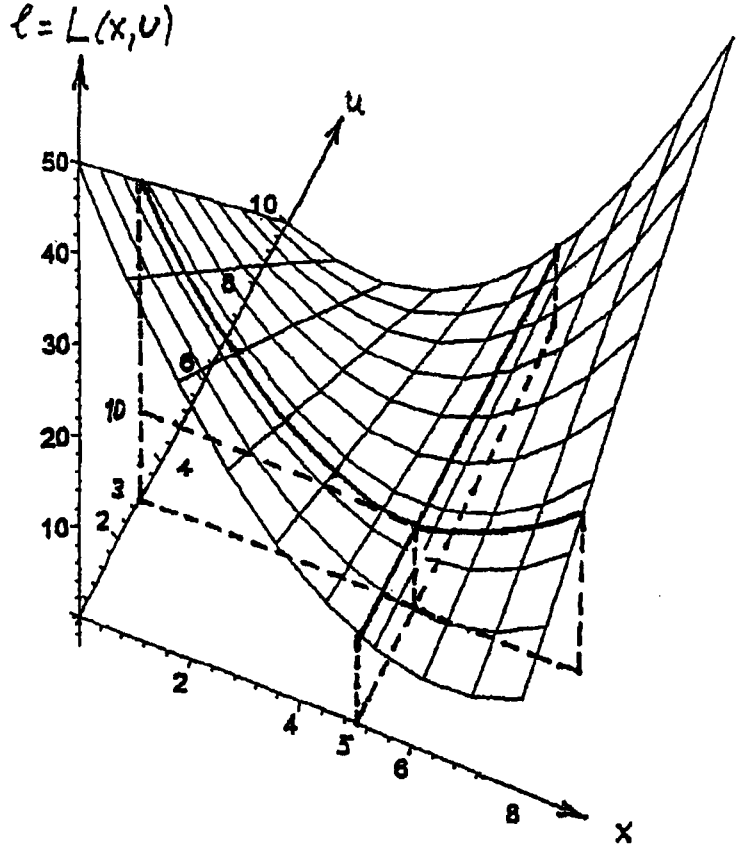


Figure 1: Global saddle point $(\hat{x}, \hat{u}) = (5, 3)$ *on* $\mathbb{R} \times \mathbb{R}$ *of the Lagrange function* $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, $L(x, u) = x^2 - 13x + 50 + (x - 5)u$, *for the problem: Minimize* $x^2 - 13x + 50$ *under the constraint* $x \leq 5$

Nachbarschaftssuche bei mehrkriteriellen Flow Shop Problemen

Walter Habenicht und Martin Josef Geiger

Insitut für Betriebswirtschaftslehre
Universität Hohenheim

1 Einleitung

Das vorherrschende Paradigma der Ablaufplanung ist nach wie vor geprägt durch den klassischen Optimierungsansatz. Es werden diskrete (kombinatorische) Optimierungsprobleme betrachtet, die eine Zielfunktion unter Beachtung bestimmter Präzedenzbeziehungen und ggf. weiterer problemspezifischer Restriktionen optimieren. Dabei sind in den letzten Jahren erhebliche Fortschritte in der Entwicklung leistungsfähiger Lösungsalgorithmen erzielt worden (vgl. z. B. Brucker, 2001).

Das Spektrum der Zielfunktionen, das hierbei betrachtet wird, reicht von Kostenzielen (Umrüstkosten, Kapitalbindungskosten...) über Zeitziele (durchschnittliche oder maximale Durchlaufzeiten...) bis zu Terminzielen (durchschnittliche oder maximale Terminüber-(unter-)schreitung, Zahl der Terminüber-(unter-)schreitungen...). Optimierungsansätze greifen eine der Zielsetzungen heraus und bestimmen die zugehörige optimale Lösung.

Nun ist bei praktischen Ablaufplanungsproblemen keineswegs immer klar, welche dieser Zielsetzungen für das Problem bestimmend oder von solch herausragender Bedeutung ist, dass sich die Lösung daran orientieren sollte. Vielmehr werden häufig mehrere Ziele gleichzeitig in Betracht zu ziehen sein, ohne dass sich deren jeweilige Bedeutung exakt quantifizieren ließe.

Für diesen Fall bietet die Mehrzieloptimierung Lösungskonzepte an (einen Überblick findet man z. B. in Habenicht et al., 2002). Diese unterscheiden sich von Optimierungsansätzen insbesondere dadurch, dass sie nicht darauf abzielen, *eine* Lösung zu finden, sondern in einem interaktiven Prozess dem Entscheidungsträger *mehrere* Lösungen präsentieren, aus denen er eine Auswahl treffen kann.

Ziel dieses Beitrags ist es, zu zeigen, wie im Bereich der Ablaufplanung ein derartiges Lösungskonzept gestaltet sein kann. Dabei konzentrieren wir uns auf den Bereich der (Permutations-) Flow Shop Probleme. Für Probleme dieses Typs haben sich Suchverfahren als Erfolg versprechend erwiesen.

Wir werden im Folgenden zeigen, wie hier das Konzept der Nachbarschaftssuche in zweifacher Weise angewendet werden kann. Zum einen im Alternativenraum zur Identifizierung effizienter Ablaufpläne und zum anderen im Ergebnisraum zur Auswahl eines „besten" Ablaufplans. Natürlich werden in beiden Fällen unterschiedliche Nachbarschaftsbegriffe zur Anwendung kommen.

1.1 Multikriterielle Flow Shop Probleme

Flow Shop Probleme innerhalb der Ablaufplanung sind durch ein Vorliegen einer Menge an Maschinen $\mathcal{M} = \{M_1, \ldots, M_m\}$ gekennzeichnet, welche zur Bearbeitung einer Menge an Fertigungsaufträgen $\mathcal{J} = \{J_1, \ldots, J_n\}$ bereit stehen. Jeder Auftrag besteht hierbei aus einer Menge an Operationen (Arbeitsgänge) $\{O_{j1}, \ldots, O_{jo_j}\}$, wobei jede Operation O_{jk} auf einer Maschine $M_i \in \mathcal{M}$ zu bearbeiten ist und eine nichtnegative Bearbeitungszeit p_{jk} aufweist. Die Operationen der Aufträge sind durch Präzedenzbeziehungen $O_{jk} \prec O_{jk+1}$ geordnet, so dass der Beginn der Bearbeitung einer Operation O_{jk+1} erst nach Abschluss der Bearbeitung von O_{jk} möglich ist. Ferner sind die Maschinenfolgen der Arbeitsgänge für alle Aufträge identisch (Reihenfertigung), wodurch sich eine Abgrenzung der Problemstellung zu anderen Bereichen der Ablaufplanung wie z. B. dem Job Shop Scheduling mit unterschiedlichen Maschinenfolgen (Werkstattfertigung) ergibt (vgl. Błażewicz et al., 2001). Für den hier betrachteten Fall des *Permutations Flow Shop Scheduling* ist zudem davon auszugehen, dass auch die Auftragsfolgen auf den m Maschinen identisch sind, d. h. die Bearbeitung bezüglich der Warteschlangen vor den Maschinen nach dem First In First Out Prinzip vonstatten geht. Die Lösung einer derartigen Problemstellung ist somit eine für alle Maschinen identische Permutation $\pi = \{\pi_1, \ldots, \pi_n\}$ aller Aufträge $\mathcal{J}$.

Eine zulässige Alternative $x \in X$ eines Flow Shop Scheduling Problems definiert unter Berücksichtigung aller Nebenbedingungen für alle Operationen O_{jk} Startzeitpunkte t_{jk}. Unter Einbezug der Bearbeitungszeiten p_{jk} ergibt sich unter Annahme einer nicht unterbrechbaren Bearbeitung somit ein Endzeitpunkt der Operation O_{jk} von $C_{jk} = t_{jk} + p_{jk}$ sowie ein Fertigstellungszeitpunkt C_j des kompletten Fertigungsauftrags von $C_j = C_{jo_j}$.

Ausgehend von der Alternativenrepräsentation in Form einer Auftragspermutation π ist die Ermittlung der Starttermine aller Arbeitsgänge gemäß den Gleichungen (1), (2), (3) und (4) möglich.

$$
\begin{aligned}
C_{\pi_1 1} &= p_{\pi_1 1} && (1)\\
C_{\pi_j 1} &= C_{\pi_{j-1} 1} + p_{\pi_j 1} \quad \forall j \in \{2,\ldots,n\} && (2)\\
C_{\pi_1 k} &= C_{\pi_1 k-1} + p_{\pi_1 k} \quad \forall k \in \{2,\ldots,m\} && (3)\\
C_{\pi_j k} &= \max\{C_{\pi_{j-1} k}, C_{\pi_j k-1}\} + p_{\pi_j k} && (4)\\
&\quad \forall j \in \{2,\ldots,n\}, k \in \{2,\ldots,m\}
\end{aligned}
$$

Die Planung der Bearbeitungszeiten der Operationen erfolgt in der überwiegenden Anzahl der verfügbaren wissenschaftlichen Publikationen unter monokriterieller Minimierung des maximalen Fertigstellungszeitpunkts $C_{max} = \max\{C_1,\ldots,C_n\}$. Zwar wurde der multikriterielle Charakter praktischer Probleme der Ablaufplanung bereits in frühen Arbeiten bemerkt (vgl. Conway et al., 1967, Rinnooy Kan, 1976), jedoch tragen erst in jüngster Zeit Forschungsaktivitäten der simultanen Existenz multipler Zielsetzungen Rechnung (T'kindt und Billaut, 2002). Weitere Zielsetzungen von wesentlicher Bedeutung sind in diesem Kontext die Minimierung der Summe der Fertigstellungszeitpunkte $C_{sum} = \sum C_j$ sowie an Lieferterminen d_j der Aufträge J_j orientierte Ziele. Insbesondere Verspätungen $T_j = \max\{C_j - d_j, 0\}$ werden hierbei betrachtet. Die Minimierung der maximalen Verspätung $T_{max} = \max T_j$ und der Summe der Terminüberschreitungen $T_{sum} = \sum T_j$ sind hier wichtige Zielkriterien. Für Problemstellungen, bei denen eine Verletzung vereinbarter Liefertermine zu fixen Konventionalstrafen führt, kommt zudem die Betrachtung der Anzahl der verspäteten Aufträge $U = \sum U_j$ mit $U_j = 1$ für $T_j > 0$ und $U_j = 0$ für $T_j = 0$ in Frage.

Bislang konnten nur wenige Lösungsansätze entwickelt werden, welche für spezielle mehrkriterielle Flow Shop Probleme ein Auffinden aller optimaler Lösungen garantieren. So beschreiben Daniels und Chambers (1990) einen Branch-and-Bound Algorithmus für Probleme mit zwei Maschinen und den Zielfunktionen C_{max} und T_{max}, und Liao et al. (1997) ein Verfahren für entsprechende Probleme mit den Optimalitätskriterien C_{max} und U. Demgegenüber kommen heuristische Methoden, deren Anwendung nicht auf Zweimaschinenprobleme beschränkt ist und die eine Approximation der optimalen Lösungen innerhalb kurzer Zeit ermöglichen, vergleichsweise häufig zum Einsatz. In diesem Zusammenhang sind vor allem lokale Suchverfahren anzuführen, auf welche noch in Abschnitt 2 detaillierter eingegangen wird.

1.2 Konzepte der diskreten Vektoroptimierung

Ein allgemeines diskretes Vektoroptimierungsproblem kann wie folgt formuliert werden:

$$\text{„Minimiere“ } y = f(x) \tag{5}$$

$$x \in X \tag{6}$$

Hierin ist X die endliche Alternativenmenge und $f : X \to Y \subset \mathbb{R}^m$ eine vektorwertige Zielfunktion. (5) und (6) bezeichnen das Vektoroptimierungsproblem im Alternativenraum. Die folgende äquivalente Formulierung beschreibt das Problem im Ergebnisraum:

$$\text{„Minimiere“ } y \in Y \tag{7}$$

Die Minimierungsvorschrift ist hier in seiner vektoriellen Form zu interpretieren als die Bestimmung von nicht-dominierten bzw. effizienten Elementen. Wir bezeichnen mit $N(Y) := \{y \in Y \mid y' \leqslant y \wedge y' \neq y \Rightarrow y' \notin Y\}$ die Menge nicht-dominierter Ergebnisse und mit $P := \{x \in X \mid f(x) \in N(Y)\}$ die Menge effizienter Alternativen.

Wir wollen im Folgenden unter der Lösung eines Vektoroptimierungsproblems die Bestimmung jener effizienten Alternative verstehen, die den Präferenzen des Entscheidungsträgers „bestmöglich entspricht“. Hierzu verfolgen wir ein interaktives Lösungskonzept, in dem der Entscheidungsträger partielle Informationen über seine Präferenzen bereitstellt, die zur Ergebnisauswahl herangezogen werden.

Ein Lösungskonzept muss zwei Aufgaben erfüllen. Es muss effiziente Alternativen identifizieren und es muss ein Auswahlverfahren unter den effizienten Alternativen bereitstellen. Je nachdem, ob es diese Aufgaben simultan oder sukzessiv ausführt, sprechen wir von einem einstufigen oder einem zweistufigen Verfahren.

Einstufige Verfahren sind so aufgebaut, dass sie zunächst bestimmte Präferenzinformationen (Zielgewichte, Anspruchsniveaus etc.) vom Entscheidungsträger erfragen. Auf der Basis dieser Informationen wird ein Optimierungsproblem gebildet, dessen Lösung eine effiziente Alternative generiert. Akzeptiert der Entscheidungsträger diese Alternative, gilt das Problem als gelöst. Akzeptiert er sie nicht, dann wird er aufgefordert, seine Präferenzinformationen zu revidieren und es wird ein neues Optimierungsproblem gelöst.

Bei dieser Vorgehensweise ist der Entscheidungsträger durchgängig in das Verfahren involviert. Handelt es sich bei dem zu lösendes Problem um ein $\mathcal{NP}$-schweres Optimierungsproblem, – und dies ist bei Ablaufplanungsproblemen sehr häufig der Fall – dann sind bei realen Problemstellungen häufig nichtakzeptable Antwortzeiten zu erwarten. In diesem Fall bietet sich ein zweistufiges Lösungskonzept an.

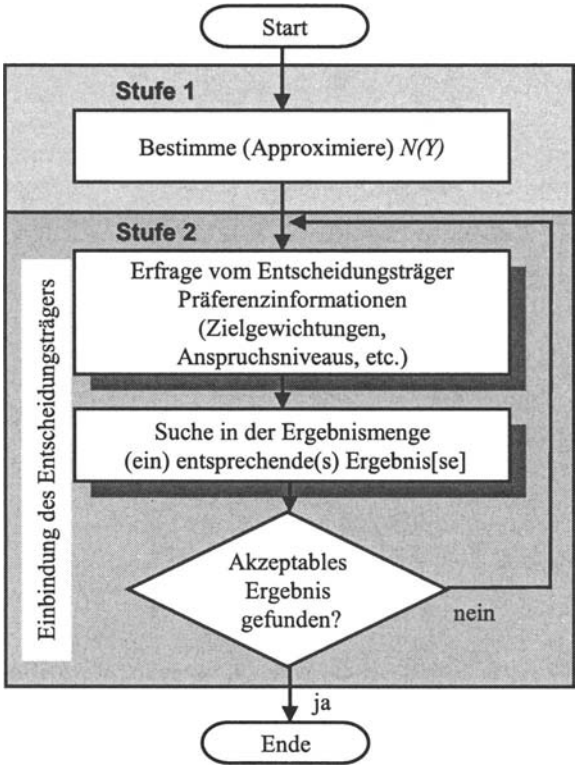


Abbildung 1: Zweistufiges Konzept

In Abbildung 1 ist ein zweistufiges Lösungskonzept skizziert. Hier werden Identifizierungs- und Auswahlproblem nacheinander gelöst. Die erste Stufe ist allein der Identifizierung aller effizienten Alternativen beziehungsweise deren nichtdominierten Ergebnissen gewidmet. Hierfür ist die Mitwirkung des Entscheidungsträgers nicht erforderlich. Bei schwer lösbaren Problemen wird man sich auf eine Approximation der effizienten Menge beschränken. Nach Abschluss der ersten Stufe ist die Menge der effizienten Alternativen (bzw. deren Approximation) explizit bekannt. Damit ist auch die Menge nicht-dominierter Ergebnisse bekannt. Der Auswahlprozess wird dann als Suchprozess in dieser Menge realisiert.

Wir werden im Folgenden zeigen, wie im Falle der Ablaufplanung von mehrkriteriellen Flow Shop Problemen in beiden Stufen das Konzept der Nachbarschaftssuche sinnvoll eingesetzt werden kann.

2 Nachbarschaftssuche im Alternativenraum

2.1 Grundlagen lokaler Suche

Verfahren der Nachbarschaftssuche im Alternativenraum können wie in Abbildung 2 skizziert beschrieben werden. Ausgehend von der Generierung einer Ausgansglösung x werden durch Anwendung von so genannten Nachbarschafsoperatoren neue Alternativen erzeugt. Diese werden im Folgenden auf Akzeptanz geprüft und entweder verworfen oder in den Suchprozess integriert. Formal beschreibt ein Nachbarschaftoperator $\mathbf{N}$ hierbei eine Verfahrensvorschrift, die zu einer gegebenen Lösung x durch Modifikation eine Menge anderer Lösungen $\mathbf{N}(x) \subseteq X$, eine so genannte *Nachbarschaft* erzeugt. Alle Lösungen $x' \in \mathbf{N}(x)$ heissen *in der Nachbarschaft* oder *Nachbarn* von x.

1: Eröffnungsverfahren: Generiere Ausgansglösung x
2: **Wiederhole**
3: Verbesserungsverfahren: Erzeuge neue Lösung
4: Bewerte neue Lösung, prüfe auf Akzeptanz
5: **Bis** Abbruchkriterium erfüllt

Abbildung 2: Pseudo-code eines lokalen Suchalgorithmus

Die Festlegung einer Nachbarschaftsdefinition $\mathbf{N}$ induziert somit in der Alternativenmenge X einen gerichteten Graphen $G_{\mathbf{N}}(V, A)$, bei dem jede Lösung $x \in X$ durch einen Knoten $v_x \in V$ repräsentiert wird. Ein Pfeil $a = (v_x, v_{x'}) \in A$ existiert, falls $x' \in \mathbf{N}(x)$. Die Vorgehensweise des lokalen Suchalgorithmus kann somit als ein iteratives Wandern in $G_{\mathbf{N}}$ von Knoten zu Knoten entlang der Richtung der definierten Pfeile interpretiert werden.

Für die Analyse lokaler Suchverfahren ist insbesondere von Interesse, ob es unter Einsatz der Nachbarschaftsdefinition $\mathbf{N}$ möglich ist, von jeder Alternative $x \in X$ die Optimallösung x^* beziehungsweise bei multikriterieller Optimierung alle Elemente x^* der Menge Pareto-optimaler Alternativen P zu erreichen. Da diese jedoch in aller Regel nicht bekannt sind, wird zum Nachweis einer möglichen Konvergenz des Algorithmus eine allgemeinere Betrachtung von $\mathbf{N}$ vorgenommen. Hierzu werden die Begriffe des *Wegs* und der *Verbundenheit von X bezüglich* $\mathbf{N}$ eingeführt.

Definition 1 (Weg von $G_{\mathbf{N}}$) *Eine Folge von Pfeilen $a_1, \ldots, a_k$ heißt* Weg $W_{\mathbf{N}}$ von $G_{\mathbf{N}}$, *falls eine Folge $v_{x_1}, \ldots, v_{x_k}$ von Knoten mit $a_h = (v_{x_{h-1}}, v_{x_h}) \forall\, h = 2, \ldots, k$ existiert. Existieren in $G_{\mathbf{N}}$ keine parallelen Pfeile, so kann ein Weg $W_{\mathbf{N}}$ durch die in ihm enthaltenen Knoten symbolisiert werden, z. B. $w_{\mathbf{N}} = (v_{x_1}, \ldots, v_{x_k})$.*

Definition 2 (Verbundenheit von X bezüglich N) *Die Alternativenmenge X heißt* verbunden bezüglich der Nachbarschaft N, *falls ein Weg von v_x nach $v_{x'}$ in $G_{\mathbf{N}} \forall v_x, v_{x'} \in A$ existiert.*

Definition 2 stellt somit eine wesentliche, notwendige Bedingung für die Konvergenz von auf Nachbarschaftssuche im Alternativenraum basierenden Verfahren dar.

Bezüglich der konkreten Ausgestaltung des Suchverfahrens sind unter dem Begriff der *Metaheuristiken* eine Reihe an Methoden entwickelt worden, welche in bezug auf ihre Steuerparameter und die eingesetzten Nachbarschaftsdefinitionen unterschiedliche Strategien verfolgen. Zu den bekanntesten zählen hierbei Simulated Annealing (vgl. Kirkpatrick et al., 1983), Evolutionäre Algorithmen (vgl. Holland, 1975) und Tabu Search (vgl. Glover, 1986).

Entsprechende Anwendungen auf mehrkriterielle Flow Shop Probleme existieren (vgl. z. B. Murata et al., 1996, Basseur et al., 2002, Ishibuchi et al., 2003), wobei insbesondere den Evolutionären Algorithmen ein besonderer Forschungsschwerpunkt zukommt, indem hier die Identifikation bzw. Approximation einer Menge effizienter Alternativen durch eine Suchmenge in einem einzigen Optimierungslauf erfolgversprechend erscheint.

2.2 Leistungsfähigkeit von Nachbarschaften

Ein wichtiger Aspekt bei der Ausgestaltung eines Lösungskonzeptes ist die Wahl eines geeigneten Nachbarschaftsoperators N. Für permutationsbasierte Alternativenrepräsentationen wie im vorliegenden Fall kommen neben verschiedener Rekombinationsoperatoren Evolutionärer Algorithmen drei grundlegende Nachbarschaften in Frage (vgl. Reeves, 1999).

1. Die Exchange Nachbarschaft $\mathbf{N}_{ex}$, welche die Positionen zweier Elemente π_j und π_k miteinander vertauscht.

2. Die Forward-Shift Nachbarschaft $\mathbf{N}_{fsh}$, welche einen Auftrag an der Position π_j entnimmt und an die Position π_k mit $k > j$ einfügt, somit bezüglich des Zeithorizonts der Planung nach „vorne“ schiebt.

3. Die Backward-Shift Nachbarschaft $\mathbf{N}_{bsh}$, die dem Forward-Shift Operator in umgekehrter Richtung der Positionsänderung von π_j nach π_k mit $k < j$ entspricht.

Die Untersuchung der Leistungsfähigkeit verschiedener Nachbarschaftsdefinitionen ist unter Einsatz eines lokalen Suchverfahrens wie in Abbildung 3 beschrieben möglich. Hierzu wird im Verlauf der Suche eine Approximation P^{approx} der Pareto Menge P erzeugt, welche alle Alternativen, für

die keine sie dominierenden Lösungen gefunden werden, speichert. Die Suche terminiert hierbei, wenn die Nachbarschaften aller Elemente in P^{approx} untersucht wurden.

1: Initialisiere Steuerparameter: Lege Nachbarschaft **N** fest
2: Generiere eine Ausgangslösung x
3: Setze $P^{approx} = \{x\}$
4: **Wiederhole**
5: Wähle $x \in P^{approx}$ mit noch nicht untersuchter Nachbarschaft
6: Erzeuge $\mathbf{N}(x)$
7: Aktualisiere P^{approx} mit allen $x' \in \mathbf{N}(x)$
8: **Wenn** $x \in P^{approx}$ **dann**
9: Markiere Nachbarschaft von x als untersucht
10: **Ende Wenn**
11: **Bis** $\nexists x \in P^{approx}$ mit noch nicht untersuchter Nachbarschaft

Abbildung 3: Mehrkriterielle lokale Nachbarschaftssuche

Die Leistungsfähigkeit der erläuterten Nachbarschaften $\mathbf{N}_{ex}$, $\mathbf{N}_{fsh}$ und $\mathbf{N}_{bsh}$ wurde innerhalb einer Implementierung des Pseudo-codes aus Abbildung 3 untersucht. Hierzu wurden für unterschiedliche Kombinationen an Zielfunktionen jeweils 100 Probleminstanzen der Größe $n = m = 10$ generiert, für die die enumerative Bestimmung der effizienten Ergebnisse in akzeptabler Zeit möglich ist. Die betrachteten Zielfunktionen können hierbei Tabelle 1 entnommen werden.

Tabelle 1: Übersicht der getesteten Problemklassen.

Bezeichung	Anzahl Ziele	Zielfunktionen
γ_1	2	C_{max}, T_{max}
γ_2	2	C_{max}, C_{sum}
γ_3	2	C_{max}, T_{sum}
γ_4	2	T_{max}, T_{sum}
γ_5	2	C_{sum}, T_{max}
γ_6	2	C_{sum}, T_{sum}
γ_7	3	$C_{max}, T_{max}, T_{sum}$
γ_8	3	$C_{max}, C_{sum}, T_{max}$
γ_9	3	$C_{max}, C_{sum}, T_{sum}$
γ_{10}	3	$C_{sum}, T_{max}, T_{sum}$
γ_{11}	4	$C_{max}, C_{sum}, T_{max}, T_{sum}$

Auf der Grundlage von jeweils 100 Testläufen mit unterschiedlichen Ausgangslösungen konnten Durchschnittswerte der Approximationsqualität für die von Czyżak und Jaszkiewicz (1998) vorgeschlagenen Metriken D_1 der durschnittlichen Abweichung von P^{approx} zu P sowie der maximalen Abweichung D_2 ermittelt werden. Tabelle 2 gibt die Anzahl der Modellinstanzen an, für die unter Einsatz der jeweiligen Nachbarschaftsoperatoren die besten Ergebnisse erzielt wurden.

Es erweist sich keine einzelne Nachbarschaft als den anderen in allen untersuchten Fällen überlegen. Auch ein Einfluss der gewählten Optimalitätskriterien auf die erzielten Resultate ist nicht offensichtlich. Vielmehr lassen sich Modellinstanzen identifizieren, für die eine bestimmte Nachbarschaft vorteilhaft ist, während für andere gegenteilige Schlussfolgerungen zu ziehen sind.

Tabelle 2: Ergebnisse der Testläufe für D_1 und D_2

	D_1			D_2		
	N_{ex}	N_{fsh}	N_{bsh}	N_{ex}	N_{fsh}	N_{bsh}
γ_1	45	55	0	31	68	0
γ_2	37	22	39	40	22	36
γ_3	46	24	30	46	27	25
γ_4	34	56	10	36	56	8
γ_5	37	58	5	35	57	7
γ_6	40	3	57	42	6	49
γ_7	31	62	7	30	61	6
γ_8	32	60	8	23	64	8
γ_9	31	31	38	37	33	29
γ_{10}	39	44	16	34	41	23
γ_{11}	32	55	12	26	55	13

Im Ergebnis bedeutet dies, dass für die untersuchten Modellinstanzen keine dominierende Nachbarschaft existiert. Die erfolgreiche Lösung mehrkriterieller Flow Shop Probleme durch den Einsatz einer einzelnen Nachbarschaft erscheint vor diesem Hintergrund wenig aussichtsreich.

2.3 Variable Nachbarschaftssuche

Zur Überwindung der in Abschnitt 2.2 identifizierten Problematik der optimalen Auswahl einer Nachbarschaft bietet sich die Kombination einzelner

Nachbarschaften an. Abbildung 4 beschreibt eine mögliche und hier exemplarisch untersuchte Vorgehensweise.

1: Lege Nachbarschaften $\mathbf{N}_1, \ldots, \mathbf{N}_k$ fest
2: Generiere eine Ausgangslösung x
3: Setze $P^{approx} = \{x\}$
4: **Wiederhole**
5: Wähle $x \in P^{approx}$ mit noch nicht untersuchter Nachbarschaft
6: Wähle eine Nachbarschaft $\mathbf{N}_i$ aus der Menge der in Schritt 1 festgelegten Nachbarschaften
7: Erzeuge $\mathbf{N}_i(x)$
8: Aktualisiere P^{approx} mit allen $x' \in \mathbf{N}_i(x)$
9: **Wenn** $x \in P^{approx}$ **dann**
10: Markiere Nachbarschaft von x als untersucht
11: **Ende Wenn**
12: **Bis** $\nexists x \in P^{approx}$ mit noch nicht untersuchter Nachbarschaft

Abbildung 4: Mehrkriterielle Variable Nachbarschaftssuche (MOVNS)

Wesentliches Element dieses Vorschlags ist, dass die möglichen Nachbarschaften nicht additiv sondern lediglich alternativ zur Anwendung kommen (vgl. Schritt 6), wodurch sich der erforderliche Rechenaufwand des Verfahrens nicht erhöht. MOVNS kombiniert somit die drei grundlegenden Nachbarschaften aus Abschnitt 2.2. Die Auswahl der anzuwendenden Nachbarschaft in Schritt 6 des Algorithmus erfolgt hierbei mit einer Wahrscheinlichkeit von jeweils $\frac{1}{3}$ je Operator.

Nach der Durchführung von Testläufen und Ermittlung durchschnittlicher Werte für D_1 und D_2 wird deutlich, dass der alternative, wechselnde Einsatz verschiedener Nachbarschaften einem einzelnen Operator in der überwiegenden Anzahl der untersuchten Modellinstanzen vorzuziehen ist. Tabelle 3 gibt hierzu die Ergebnisse wieder.

Die Bedeutung der Ergebnisse wird um so deutlicher, zieht man in Betracht, dass die Verbesserung der Approximationsgüte, abgesehen von der vernachlässigbaren zufälligen Auswahl der Nachbarschaft, ohne zusätzlichen Rechenaufwand erzielt wurde. Variationen der Suchstrategie im Alternativenraum erweisen sich für den untersuchten Fall der mehrkriteriellen Flow Shop Probleme unabhängig von den relevanten Zielfunktionen als günstig.

Tabelle 3: Ergebnisse der Testläufe für D_1 und D_2 im Vergleich zu MOVNS

	D_1				D_2			
	$\mathbf{N}_{ex}$	$\mathbf{N}_{fsh}$	$\mathbf{N}_{bsh}$	MOVNS	$\mathbf{N}_{ex}$	$\mathbf{N}_{fsh}$	$\mathbf{N}_{bsh}$	MOVNS
γ_1	4	17	0	79	4	23	0	73
γ_2	0	5	12	83	0	6	12	82
γ_3	0	4	5	91	2	8	7	82
γ_4	1	15	1	83	5	22	1	72
γ_5	0	19	0	81	1	27	1	70
γ_6	5	0	27	68	6	1	24	69
γ_7	2	19	1	78	2	20	1	75
γ_8	0	22	0	78	0	23	1	75
γ_9	0	10	6	84	2	10	6	79
γ_{10}	1	15	3	81	3	17	5	75
γ_{11}	1	23	0	76	1	22	1	75

3 Nachbarschaftssuche im Ergebnisraum

Während die Nachbarschaftssuche im Alternativenraum naturgemäß problemspezifisch erfolgen muss, ist die Suche im Ergebnisraum problemunabhängig. Wir gehen davon aus, dass die Menge nichtdominierter Ergebnisse gegeben ist und realisieren eine Nachbarschaftssuche in dem k-dimensionalen Ergebnisraum. Wir werden im Folgenden unsere Ausführungen an einem zweikriteriellen Flow Shop Problem mit 10 Maschinen und 10 Aufträgen verdeutlichen. Als Ziele werden hier betrachtet die Zykluszeit (1. Ziel) und die Summe der Terminüberschreitungen (2. Ziel). Das Problem besitzt die folgende Menge nichtdominierter Ergebnisse:

$$N(Y) = \left\{ \begin{array}{l} \binom{1181}{2355}, \binom{1126}{2802}, \binom{1130}{2756}, \binom{1122}{2962}, \binom{1288}{2127}, \binom{1176}{2402}, \binom{1189}{2348}, \\ \binom{1137}{2463}, \binom{1105}{3005}, \binom{1131}{2728}, \binom{1257}{2201}, \binom{1309}{2098}, \binom{1205}{2285}, \binom{1277}{2199}, \\ \binom{1312}{2071}, \binom{1237}{2217}, \binom{1112}{2992}, \binom{1158}{2404}, \binom{1203}{2302}, \binom{1236}{2229}, \binom{1136}{2489}, \\ \binom{1190}{2325}, \binom{1157}{2430}, \binom{1224}{2266}, \binom{1278}{2174}, \binom{1311}{2082}, \binom{1180}{2367}, \binom{1106}{3000} \end{array} \right\}$$

In diesem Beispiel enthält $N(Y)$ nur wenige Elemente, so dass man bei der Durchführung von Suchverfahren Nachbarschaften identifizieren könnte, indem man $N(Y)$ jeweils sequenziell durchsucht. Es erscheint jedoch sinnvoll, $N(Y)$ so zu strukturieren, dass die verwendeten Suchverfahren effizient realisiert werden können.

3.1 Speicherformen der nichtdominierten Ergebnisse

Wir werden dafür zwei Datenstrukturen vorstellen, die dies leisten können. Das sind Quadbäume und kd-Bäume. Diese werden daraufhin untersucht, wie sie die Identifizierung bestimmter Nachbarschaften unterstützen.

3.1.1 Effiziente Quadbäume

Quadbäume wurden von Finkel und Bentley (vgl. Finkel und Bentley, 1974) zur Suche in zweidimensionalen Suchräumen eingeführt. Habenicht hat diese Datenstruktur für höherdimensionale Räume verallgemeinert und ihre besondere Eignung zur Unterstützung der Auswertung der Dominanzrelation herausgearbeitet (vgl. Habenicht, 1984).

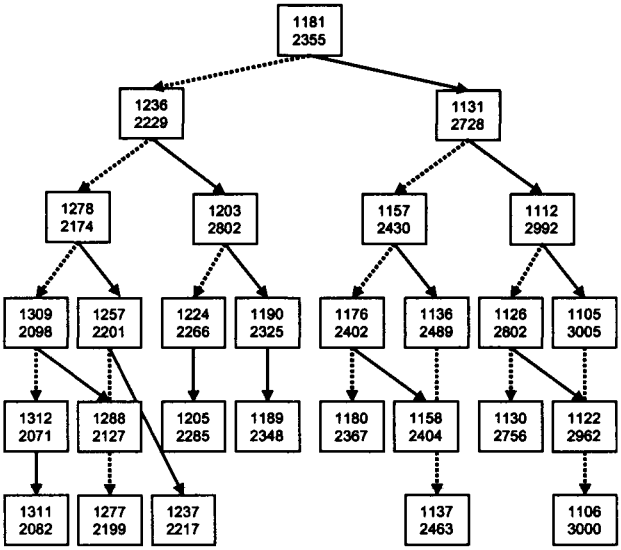


Abbildung 5: Quadbaum

Abbildung 5 zeigt einen Quadbaum für unser Beispielproblem und in Abbildung 6 ist dargestellt, wie dieser Quadbaum den Ergebnisraum strukturiert. Die Wurzel des Baums zerlegt den Raum in die vier Quadranten (im k-dimensionalen Raum in die 2^k Orthanten), die die Wurzel als Spitze besitzen. Jeder weitere Knoten zerlegt den Bereich, in dem er sich befindet wiederum entsprechend in 4 (2^k) Bereiche usf. Enthält ein Quadbaum nur effiziente Vektoren, dann sind zwei der Quadranten (Orthanten) leer, denn sie enthielten Vektoren, die den Knoten dominieren, bzw. von dem Knoten dominiert werden. In diesem Fall hat damit jeder Knoten eines Quadbaums im m-dimensionalen Raum maximal $2^m - 2$ Söhne. In unserem Beispiel wird daher der Quadbaum zu einem binären Baum.

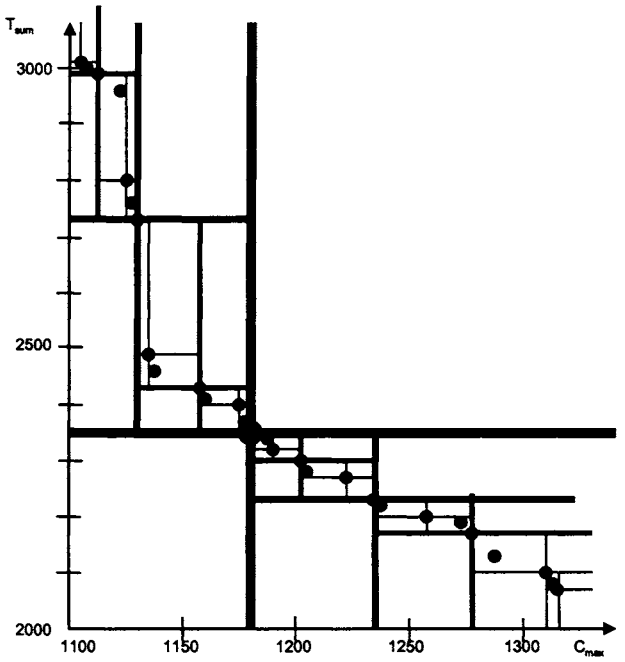


Abbildung 6: Bereichssegmentierung durch den Quadbaum

Der Vorteil dieser Datenstruktur liegt darin, dass sie die Auswertung der Dominanzrelation unterstützt. Das bedeutet, dass man sich bei der Auswertung der Dominanzrelation auf die Überprüfung einer geringen Teilmenge der Knoten beschränken kann. (Eine ausführliche Darstellung des Einsatzes von Quadbäumen findet man in Habenicht, 1984, S. 38 ff.)

3.1.2 kd-Bäume

kd-Bäume unterstützen ebenfalls Bereichssuchen in mehrdimensionalen Räumen. Hier findet aber eine andere Segmentierung des Suchraums statt. Ein Knoten zerlegt den Suchraum nicht in Quadranten bzw. Orthanten sondern in Halbräume. Die Dimensionen des Raums werden also nicht simultan sondern sequenziell zur Zerlegung des Raums herangezogen. Ein Vorteil der kd-Bäume gegenüber Quadbäumen liegt in ihrer besseren (statischen) Balancierbarkeit. Abbildung 7 zeigt einen kd-Baum für unser Beispiel. Die fettgedruckten Zahlen zeigen an, welche Komponente zur Zerlegung herangezogen wurde. Abbildung 8 zeigt die durch diesen Baum induzierte Zerlegung des Ergebnisraums.

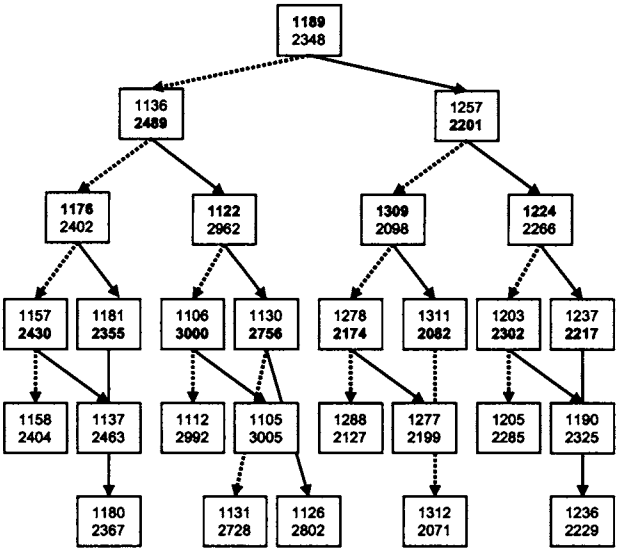


Abbildung 7: kd-Baum

3.2 Nachbarschaften

Die generelle Vorgehensweise der Nachbarschaftssuche im Ergebnisraum ist in Abbildung 9 dargestellt.

Beginnend mit einem Ergebnis y^* soll über eine Abfolge von Nachbarschaften das beste Ergebnis ermittelt werden. Die dabei einzusetzenden Nachbarschaften sind unter folgenden Aspekten zu bewerten:

- Aufwand

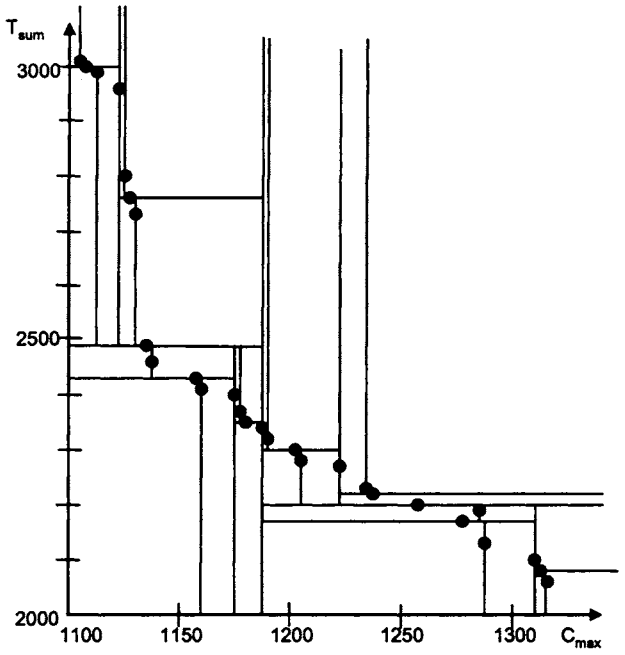


Abbildung 8: Bereichssegmentierung durch den kd-Baum

- Erreichbarkeit
- Informationsbedarf
- Steuerbarkeit

Unter Aufwand verstehen wir einfach den Aufwand, der nötig ist, um alle Nachbarschaftselemente eines Ergebnisses y^* zu bestimmen. Dieser hängt insbesondere davon ab, welche Datenstrukturen ausgenutzt werden können. Erreichbarkeit ist eine sehr wichtige Eigenschaft. Sie bezeichnet die Tatsache, ob jedes Element der Suchmenge von jedem anderen aus durch eine Abfolge von Nachbarschaften erreichbar ist. Unter dem Informationsbedarf verstehen wir, ob in die Nachbarschaftsdefinition Präferenzinformationen des Entscheidungsträgers eingehen. Unter Steuerbarkeit betrachten wir die Frage, inwieweit die Kardinalität der Nachbarschaft a priori abgeschätzt werden kann.

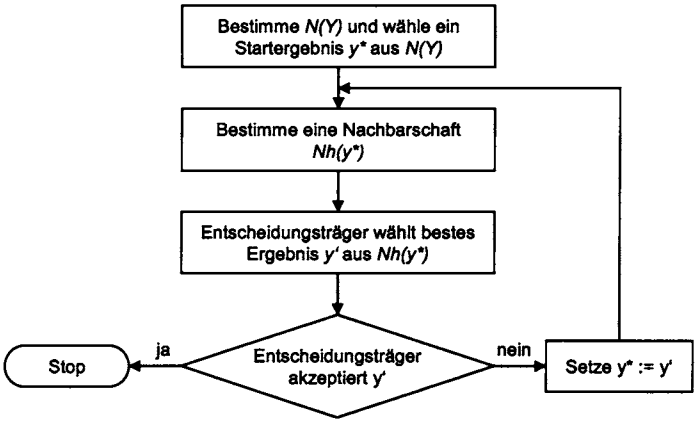


Abbildung 9: Nachbarschaftssuche im Ergebnisraum

3.2.1 Quadbaumnachbarschaft

Ist auf der Ergebnismenge die Datenstruktur Quadbaum implementiert, dann besteht eine nahe liegende Vorgehensweise darin, diese Baumstruktur unmittelbar zur Suche heranzuziehen. In diesem Fall wären die Nachbarn eines Knoten i. d. R. seine Söhne (eine genaue Definition der (Quad-) Baumsuche findet man in Habenicht, 2000, S. 187 f.)

Die Quadbaumnachbarschaft ist offenbar mit minimalem Identifizierungsaufwand zu realisieren. Wie Abbildung 10 zeigt, führt diese sehr starre Nachbarschaftssuche i. A. aber nur zur Auswahl suboptimaler Ergebnisse. Dies liegt insbesondere an der eingeschränkten Erreichbarkeit. Von einem Knoten aus sind nur die Knoten des Teilbaums erreichbar, dessen Wurzel er ist.

Im linken Teil der Abbildung 10 haben wir Indifferenzkurven im Ergebnisraum eingetragen. Verhält sich der Entscheidungsträger entsprechend dieser Indifferenzkurven, so würde er als Suchweg die im Quadbaum fett gerahmten Knoten durchlaufen und den Knoten (1105; 3005) auswählen. Das optimale Ergebnis wäre dagegen (1312; 2071).

3.2.2 Distanznachbarschaft

Eine Nachbarschaft, die ebenfalls die Quadbaumstruktur sehr gut ausnutzen kann, ist die Distanznachbarschaft. Diese wird gebildet durch jene Ergebnisse, die von y^* gemäß eines Abstandsmaßes nicht weiter als α Einheiten entfernt sind. Wählt man als Abstandsmaß die Tschebytscheff-Norm,

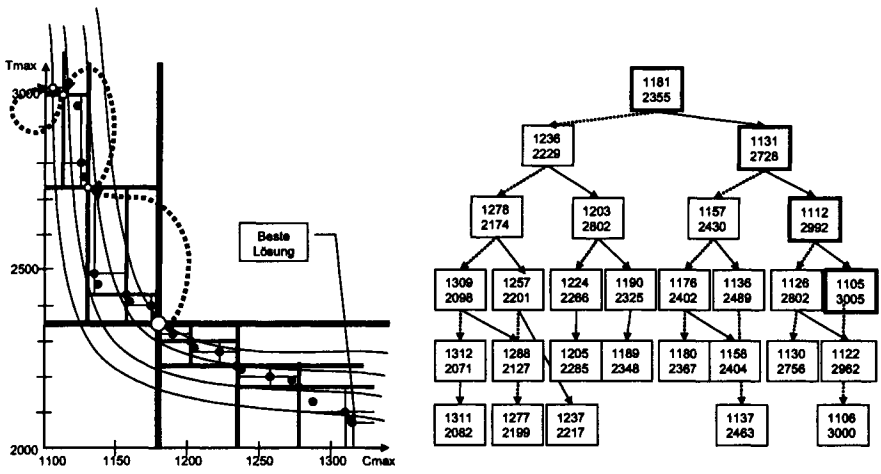


Abbildung 10: Quadbaumnachbarschaft

dann lässt sich das Problem der Identifizierung der Nachbarn formulieren als die Suche nach den Ergebnissen, die den Vektor $(y_1^* + \alpha, y_2^* + \alpha, \ldots, y_m^* + \alpha)^T$ dominieren und von dem Vektor $(y_1^* - \alpha, y_2^* - \alpha, \ldots, y_m^* - \alpha)^T$ dominiert werden.

In Abbildung 11 ist die Ermittlung der Distanznachbarschaft von $y^* = (1288; 2127)^T$ für $\alpha = 50$ im Quadbaum dargestellt. Wie man sieht, ist auch hier der Aufwand zur Ermittlung recht begrenzt. Die fett gerahmten Knoten sind jene, die zur Ermittlung der Nachbarschaft untersucht werden müssen; die grau hinterlegten gehören zur Nachbarschaft.

Unter dem Aspekt der Steuerbarkeit ist diese Nachbarschaft kritisch zu bewerten. Die Anzahl der Nachbarn ist a priori nicht abschätzbar. Wird α zu klein gewählt, kann die Nachbarschaft leer sein. Wird es dagegen zu groß gewählt, enthält die Nachbarschaft zu viele Elemente. Die Größe der Nachbarschaft sollte die Informationsverarbeitungskapazität des Entscheidungsträgers berücksichtigen. Sie sollte daher kaum mehr als etwa sieben Elemente enthalten.

Außerdem ist die Erreichbarkeit bei dieser Nachbarschaft nicht gewährleistet. Man sieht an diesem Beispiel anhand der linken Darstellung in Abbildung 11, dass bspw. die Knoten im linken oberen Bereich bei $\alpha = 50$ von der Wurzel aus nicht erreichbar wären.

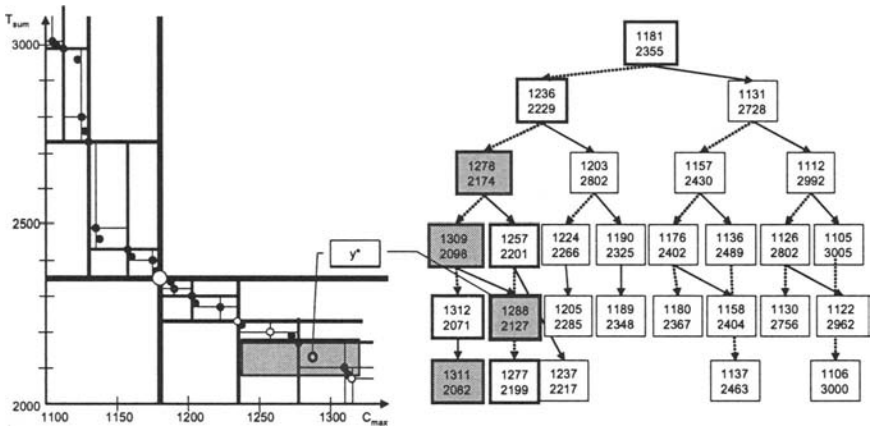


Abbildung 11: Distanznachbarschaft ($\alpha = 50$)

3.2.3 Kardinalitätsnachbarschaft

Die Steuerbarkeit der Distanznachbarschaft lässt sich verbessern, indem man die Nachbarschaft durch die p nächstgelegenen Ergebnisse bildet. Hierbei handelt es sich gewissermaßen um eine dynamische Form der Distanznachbarschaft, die wir *Kardinalitätsnachbarschaft* nennen wollen. Die Ermittlung dieser Nachbarschaft ist etwas aufwändiger als bei der (statischen) Distanznachbarschaft. Man wählt zunächst p (beliebige) Ergebnisse aus und ermittelt als α den maximalen Abstand zu y^*. Mit diesem α beginnt man die Ermittlung der Distanznachbarschaft. Wird ein weiterer Nachbar mit geringerer Distanz als α gefunden, ersetzt dieser den bisherigen Nachbarn mit Distanz α und α wird ggf. angepasst.

Die Erreichbarkeit ist auch bei dieser Form der Distanznachbarschaft nicht gewährleistet.

3.3 Richtungsnachbarschaft

Nachbarschaften, die die Erreichbarkeitsforderung erfüllen sind Richtungsnachbarschaften, von denen wir hier zwei Formen vorstellen wollen. Die Erfüllung der Erreichbarkeitsforderung wird bei diesen durch ein weiteres Ansteigen des Ermittlungsaufwands erkauft. Neben dem Quadbaum eignen sich zur Realisierung dieser Nachbarschaften auch kd-Bäume.

Die erste Form dieser Nachbarschaft, die wir hier vorstellen, können wir als *Kegelnachbarschaft* bezeichnen. Bei dieser Nachbarschaft betrachten wir die $2^m - 2$ Orthanten, die von y^* ausgehen und nicht dominierte Nachbarn

enthalten können. In diesen bestimmen wir jeweils das gemäß eines Abstandsmaßes – wir werden auch hier die Tschebytscheff-Norm verwenden – nächstgelegene Ergebnis.

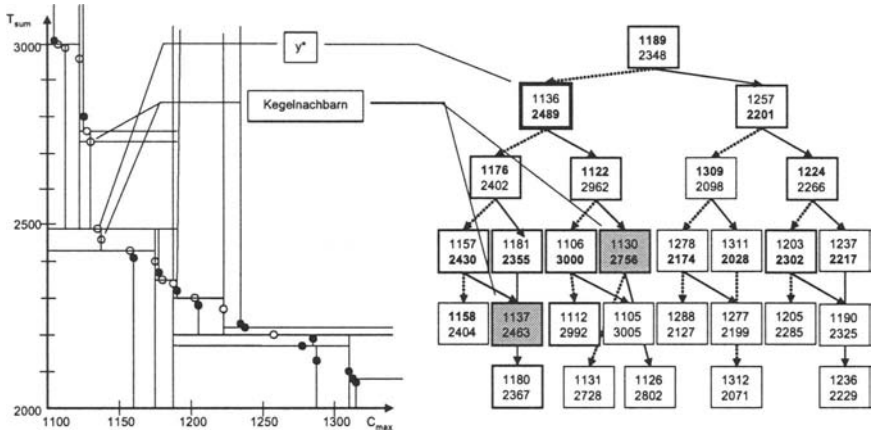


Abbildung 12: Kegelnachbarschaft

In Abbildung 12 ist die Ermittlung der Kegelnachbarschaft von $y^* = (1136; 2489)^T$ dargestellt. Die grau hinterlegten Knoten des Baums müssen dazu überprüft werden. Es sei darauf hingewiesen, dass bei einem m-dimensionalen Problem die Nachbarschaft bis zu $2^m - 2$ Ergebnisse enthalten kann. Für $m > 3$ müsste man hier ggf. ein mehrstufiges Auswahlverfahren vorsehen.

Bei den bisher betrachteten Nachbarschaften wurden in der Nachbarschaftsdefinition keine Präferenzinformationen benutzt. Abschließend führen wir mit der *Richtungsnachbarschaft i. e. S.* eine Nachbarschaft ein, die auf der Formulierung eines „Präferenzkegels" basiert. Der Entscheidungsträger wird gefragt, bzgl. welcher Ergebniskomponenten er eine Verbesserung gegenüber y^* wünscht und bzgl. welcher Komponenten er eine Verschlechterung akzeptiert. Diese Angaben definieren einen Kegel, in dem die p nächsten Ergebnisse die Nachbarschaft bilden.

4 Ausblick

Wir haben in diesem Beitrag das Konzept der Nachbarschaftssuche am Beispiel mehrkriterieller Ablaufplanungsprobleme dargestellt. Dabei haben wir gezeigt, dass im Falle von Permutations Flow Shop Problemen ein variables Nachbarschaftskonzept, bei dem die Exchange Nachbarschaft mit der

Forward- und Backward- Shift Nachbarschaft kombiniert wird, zu einer Verbesserung der Identifizierung der Menge effizienter Ablaufpläne führt.

Zur Realisierung des Auswahlprozesses innerhalb der Menge effizienter Ablaufpläne haben wir verschiedene Nachbarschaftsbegriffe eingeführt und unter verschiedenen Gesichtspunkten bewertend verglichen. Es zeigte sich, dass verschiedene Baumstrukturen diese Nachbarschaftskonzepte in unterschiedlicher Form unterstützen.

Um aus den hier dargestellten Komponenten ein akzeptables Entscheidungsunterstützungssystem zu entwickeln fehlt als weitere Komponente ein benutzerfreundliches Visualisierungskonzept.

Literatur

[1] Basseur M, Seynhaeve F, El-ghazali T (2002) Design of multi-objective evolutionary algorithms: Application to the flow-shop scheduling problem. In: Congress on Evolutionary Computation (CEC'2002), Band 2. IEEE Service Center, Piscataway NJ, May 2002, S. 1151–1156

[2] Błażewicz J, Ecker KH, Pesch E, Schmidt G, Węglarz J (2001) Scheduling Computer and Manufacturing Processes, 2. Auflage. Springer Verlag, Berlin, Heidelberg, New York

[3] Brucker P (2001) Scheduling Algorithms, 3. Auflage. Springer Verlag, Berlin

[4] Conway RW, Maxwell WL, Miller LW (1967) Theory of Scheduling. Addison-Wesley, Reading, MA

[5] Czyżak P, Jaszkiewicz A (1998) Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization. Journal of Multi-Criteria Decision Analysis 7:34–47

[6] Daniels RL, Chambers RJ (1990) Multiobjective flow-shop scheduling. Naval Research Logistics 37:981–995

[7] Finkel RA, Bentley JL (1974) Quad-trees. A data structure for retrieval on composite keys. Acta Informatica 4:1–9

[8] Glover F (1986) Future paths for integer programming and links to artificial intelligence. Computers & Operations Research 13(5):533–549

[9] Habenicht W (1984) Interaktive Lösungsverfahren für diskrete Vektoroptimierungsprobleme unter besonderer Berücksichtigung von Wegeproblemen in Graphen. Anton Hain, Königstein

[10] Habenicht W (2000) Mehrzielkonzepte zur Unterstützung strategischer Entscheidungen. In: Foschiani S, Habenicht W, Schmid U und Wäscher G (Hrsg.) Strategisches Management im Zeichen von Umbruch und Wandel. Schäffer-Poeschel, Stuttgart, S. 175–195

[11] Habenicht W, Scheubrein B, Scheubrein R (2002) Multiple criteria decision making. In: Derigs et al. (Hrsg.) Encyclopedia of Life Support Systems (EOLLS). EOLLS Publishers, Oxford

[12] Holland JH (1975) Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI

[13] Ishibuchi H, Yoshida T, Murata T (2003) Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Transactions on Evolutionary Computation 7(2):204–223

[14] Kirkpatrick S, Gellat CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680

[15] Liao CJ, Yu WC, Joe CB (1997) Bicriterion scheduling the two-machine flowshop. Journal of the Operational Research Society 48:929–935

[16] Murata T, Ishibuchi H, Tanaka H (1996) Multi-objective genetic algorithm and its application to flowshop scheduling. Computers & Industrial Engineering 30(4):957–968

[17] Reeves CR (1999) Landscapes, operators and heuristic search. Annals of Operations Research 86:473–490

[18] Rinnooy Kan AHG (1976) Machine Scheduling Problems: Classification, Complexity and Computations. Martinus Nijhoff, The Hague

[19] T'kindt V, Billaut JC (2002) Multicriteria Scheduling: Theory, Models and Algorithms. Springer Verlag, Berlin, Heidelberg, New York

The Solution of a Generalization of a Bayesian Stopping Problem of MacKinnon

Karl Hinderer and Michael Stieglitz

Institut für Mathematische Stochastik
Universität Karlsruhe

Dedicated to Prof. Dr. Klaus Neumann, Karlsruhe

Abstract

We consider a substantial generalization of a problem proposed by MacKinnon (2003). Within the setting of Bayesian Markovian decision processes we derive for the maximal expected N–stage reward d_N for a random initial state an integral recursion and an algorithmic recursion. From the former we obtain results about the dependence of d_N on several parameters while the latter serves the same purpose, but also yields a numerical solution. An optimal policy is given in the form of an optimal stopping time. The model with a random initial state is dealt with by an appropriate choice of the state space.

Keywords: Bayesian control models, Bayesian optimal stopping

MS 2000 Classification: 90C40, 60G40

1 Introduction

Nick MacKinnon (2003) proposed for the special case $N = 3$ the following ***problem MKP$_N$, $N \geq 2$.*** In a game a number θ is chosen randomly in the interval $(0, 1)$ according to the uniform distribution, and it is not disclosed until the game is over. Then N numbers $z_1, z_2, \cdots, z_N$ are chosen independently at random according to the uniform distribution on the interval $(0, 1/\theta)$. As each successive number is offered, the contestant must accept or reject it, accepting z_N if all previous numbers have been rejected. If z_t is accepted at time $1 \leq t \leq N$, the number θ is revealed, the contestant receives θz_t dollars, and the game ends. According to which stopping time does one obtain the undiscounted maximal expected reward d_N? This means to find those values of z_1 which should be accepted, those values of z_2 which should be accepted in case z_1 has been rejected, $\cdots$, those values of z_{N-1} which should be accepted in case $z_1, z_2, \cdots, z_{N-2}$ have been rejected.

We sketched a solution for $N = 3$ in Hinderer/Stieglitz (2005). In the present paper we give a complete proof and many new results for problem GMKP_N, a *generalization* of the MKP_N in the following four respects: (i) The number $N \geq 2$ of maximal offers is arbitrary; this turns out to require substantially more effort than the case $N = 3$. (ii) Instead of the uniform prior for θ we admit more generally any power distribution $Pow(\alpha, b)$, defined in (2.6) below as a special Beta–distribution. This is a natural generalization since the posterior distributions for uniform prior $Pow(1,1)$ are special power distributions. (iii) We admit costs $\gamma \in \mathbb{R}_+$ for each offer. (iv) We admit a discount factor $\beta \in (0,1]$.

The reader may speculate about the maximal expected reward $d_N = d_N(\alpha, b)$. In MKP_N the very simple stopping time "accept always the first offer z_1" yields the expected reward $\int_0^1 (\int_0^{1/\theta} \theta^2 z \, dz) d\theta = 1/2$, which is a lower bound of d_N. On the other hand, since $z_t \leq 1/\theta$, we have $\theta z_t \leq 1$, hence $d_N \leq 1$. Intuitively one also expects that d_N is increasing in N, which is indeed true, cf. 3.4(a). A substantiated speculation about an optimal stopping time seems to be difficult: since large values of z_t indicate a small value of θ, a reasonable judgement about θz_t seems to be difficult.

Our problem resembles the well–known Bayesian version of the standard asset selling problem (one object for sale and no recall), but the probability distribution of the offers and the reward structure (θz_t instead of simply z_t) differ considerably, and — as our study shows — also the tools needed differ largely. Both problems are special cases of the *Bayesian stopping problem with i.i.d. offers* of Example 1 below. In the literature such problems are mostly treated for the case of a *known* initial offer s_0; see e.g. DeGroot (1970). The more realistic case where s_0 is random with the p.d. $Q(\theta)$ is mostly only treated cursorily. Here we give a rigorous and simple approach, using a somewhat tricky choice of the state space (cf. Example 1 below) in a Bayesian control model (***BCM*** for short) with *known* initial state. Martingale methods, often used for optimal stopping problems, are not considered here.

In section 2 we carefully present our theoretical basis. Theorem 2.2 gives the standard method of reducing the Bayesian problem with deterministic initial state to a certain Markovian decision process MDP′; this part is largely based on Rieder (1988); see also Rieder (1975). The result is applied in 2.3 to a Bayesian stopping problem which includes both the asset selling problem and the GMKP_N. The latter is studied in section 3, which is the core of the paper. Here methods taylored to the specific problem rather than standard methods are required. At first sight, the recursive algorithm for d_N seems to resist a numerical solution due to the uncountability of the state space. It is crucial for the overcoming of this

obstacle that $d_N(\alpha, b)$ turns out to be independent of b. In 3.3 we give the two recursions for d_N (see (3.6) and (3.7)), stated in the abstract, and an optimal stopping time. As a consequence, $d_N = 1/2 - \gamma$ if $\gamma \geq 1/2$. Our next results 3.4 and 3.5, which concern the case $0 \leq \gamma \leq 1/2$, are instructive contributions to the topic "structure of solutions": the rather involved dependence of d_N on α, β and γ is studied in 3.4 by considering properties such as bounds, monotonicity, convexity and (Lipschitz–) continuity. Similar results hold for $d := \lim_{N\to\infty} d_N$ (which in case $\beta < 1$ is the geometric limit of $(d_N)_1^\infty$); see 3.5. Computations reveal that in general d_N is neither convex nor concave in α; see Figure 1. Finally an explicit solution for the case $N = 3, \beta = 1, \gamma = 0$ and arbitrary power p.d. as prior is given in 3.6. In particular, for the original problem MKP_3 we get $d_3 = 553/864 \approx 0.6400$, and the following optimal stopping time of control limit type: Accept z_1 iff $z_1 > 11/12$; otherwise accept z_2 iff $z_2 > 2/3$; otherwise accept z_3.

Notations. $\mathbb{R}^+$ is the set of positive reals. The n-fold cartesian product of a set X is denoted by X^n, and its elements are denoted by $x^n := (x_i)_1^n := (x_1, x_2, \cdots, x_n)$. A *function* is a mapping into $\mathbb{R}$. For reals x, y we use $x \wedge y := \min(x, y)$. The Kronecker symbol is denoted by $\delta_{i,j}$. The indicator function $1_B : \Omega \to \{0, 1\}$ of a subset B of Ω is defined by $1_B(x) := 1$ iff $x \in B$. We use $\sigma_n(x) := \sum_{j=0}^{n-1} x^j$, $x \in \mathbb{R}_+$. The abbreviation p.d. is used for " probability distribution". Sometimes we write $\mu(dx)$ for a p.d. μ on the σ–algebra $\mathfrak{X}$, and $\int \mu(dx)\, f(x)$ instead of $\int f(x)\, \mu(dx)$; similarly for iterated integrals. A transition p.d. P from a measurable space $(X,\mathfrak{X})$ into a measurable space $(Y,\mathfrak{Y})$ is a function $(x, B) \mapsto P(x, B)$ on $X \times \mathfrak{Y}$ which is measurable in x and a p.d. in B. A transition density of P w.r. to a σ–finite measure ρ on $\mathfrak{Y}$ is a measurable function $p \geq 0$ on $X \times Y$ such that $P(x, B) = \int_B p(x, y)\, \rho(dy)$, $x \in X, B \in \mathfrak{Y}$. Which σ–algebras are involved in statements of measurability of sets or mappings will be clear from the context. If $(X,\mathfrak{X})$ and $(Y,\mathfrak{Y})$ are measurable spaces, we endow $X \times Y$ with the product–σ–algebra $\mathfrak{X} \otimes \mathfrak{Y}$. Questions of measurability are only treated briefly.

2 The Model for a Binary Bayesian Problem

We consider the following ***Bayesian control problem*** $BCM_N(\mu_0)$, defined by a horizon $N \geq 1$ and a data set $(\Theta, S, Z, Q, T, r, V_0, \beta)$ of the following meaning: The state s_t of a system evolves at times $t = 0, 1, \cdots, N-1$ randomly in the state space S under the influence of actions a_t taken from the action space $A := \{0, 1\}$. (A more general A could be admitted but is not required for our purpose.) Because of this form of A the $BCM_N(\mu_0)$ is called ***"binary"***. Put $D = S \times A$. The sets Θ, S and Z are endowed with

σ–algebras $\mathfrak{T}$, $\mathfrak{S}$ and $\mathfrak{Z}$, respectively. At the end of each period $[t, t+1)$ (i.e. after action a_t has been taken) the decision maker observes a value z_{t+1} from the observation space Z, which arises according to the transition p.d. $Q(\theta, dz)$ from Θ into Z. Let $z^t := (z_i)_{i=1}^t$ be the sequence of observations up to time $1 \leq t \leq N$. The parameter θ in the parameter space Θ is unknown; according to the Bayes principle we assume that we have information on it in form of a "prior" p.d. μ_0. The sequence of observations is stochastically independent, conditional on θ. The random transition from s_t to s_{t+1} under action a_t is determined by a measurable function T (called the transition function) from $D \times Z$ into S as $s_{t+1} := T(s_t, a_t, z_{t+1})$. At time $t = 0$ and at the times $1 \leq t \leq N-1$ the decision maker can base his action on the state s_0 and on (z^t, s_t), respectively. Thus the action a_t is selected according to an *N–stage observation history–dependent* policy $\pi = (\pi_t)_0^{N-1}$ as $a_0 := \pi_0(s_0)$ and $a_t := \pi_t(z^t, s_t)$, $1 \leq t \leq N-1$. Here the decision rules π_0 and π_t are measurable mappings from S and from $Z^t \times S := (\times_1^t Z) \times S$ into $\{0,1\}$, respectively. The set of such policies is denoted by Δ_N. At each time $0 \leq t \leq N-1$ a reward $r(\theta, s_t, a_t) \in \mathbb{R}$ and at time N a terminal reward $V_0(\theta, s_N) \in \mathbb{R}$ is obtained, which are discounted back to time $t = 0$ by the discount factor $\beta \in (0, 1]$. Both r and V_0 are assumed to be lower bounded and measurable. The special case where the process stops after the first time $0 \leq t \leq N-1$ where action $a_t := 1$ is taken, models Bayesian ***stopping*** problems; see Example 1.

Denote by $\mathbf{P}(\mathfrak{T})$ the set of all *priors*, i.e. of all p.d.'s on $\mathfrak{T}$. Our N-stage problem defines on Δ_N our *objective function* $\pi \mapsto v_{N\pi}(\mu, s_0) :=$ the expectation of the sum of discounted rewards, obtained under policy $\pi \in \Delta_N$, prior $\mu \in \mathbf{P}(\mathfrak{T})$ and initial state $s_0 \in S$; a definition in terms of the model is given in (2.2) below.

For the formal definition of $v_{N\pi}$ we use the sample space $\Theta \times Z^N$ with the product–σ–algebra $\mathfrak{T} \otimes \mathfrak{Z}^N$ and the generic element (θ, z^N). The random state $\zeta_{t\pi} = \zeta_{t\pi}(s_0, z^t)$ at time $1 \leq t \leq N$ under policy $\pi \in \Delta_N$ is determined recursively by $\zeta_{1\pi} := T(s_0, \pi_0(s_0), z_1)$ and $\zeta_{t+1,\pi} := T(\zeta_{t\pi}, \pi_t(z^t, \zeta_{t\pi}), z_{t+1})$, $1 \leq t \leq N-1$. Then the N–stage random reward function on $\Theta \times Z^N$ under policy π and initial state $s_0 \in S$ is given by

$$\begin{aligned}(\theta, z^N) \quad &\mapsto \quad G_{N\pi}(\theta, s_0, z^N) := r(\theta, s_0, \pi_0(s_0)) \\ &+ \quad \sum_{t=1}^{N-1} \beta^t \cdot r(\theta, \zeta_{t\pi}, \pi_t(z^t, \zeta_{t\pi})) + \beta^N \cdot V_0(\theta, \zeta_{N\pi}).\end{aligned} \tag{2.1}$$

Put $Q^N(\theta, dz^N) := \times_{t=1}^N Q(\theta, dz_t)$, which is a p.d. on $\mathfrak{Z}^N$. Since the observations are i.i.d., conditional on θ, with p.d. $Q(\theta, dz)$, we use for $\mu \in \mathbf{P}(\mathfrak{T})$

on $\mathfrak{T}\otimes\mathfrak{Z}^N$ the p.d.

$$B \mapsto \mathbb{Q}_N^\mu(B) := \int \mu(d\theta) \int Q^N(\theta, dz^N)\, 1_B(\theta, z^N).$$

Since r and V_0 are lower bounded, there exists for $\pi \in \Delta_N$ and $s_0 \in S$

$$v_{N\pi}(\mu, s_0) := \int \mathbb{Q}_N^\mu(d(\theta, z^N))\, G_{N\pi}(\theta, s_0, z^N) \in (-\infty, \infty]. \tag{2.2}$$

It is called the N–stage ***Bayes reward of policy*** $\pi \in \Delta_N$ *w.r. to prior* $\mu \in \mathbf{P}(\mathfrak{T})$ *and initial state* s_0. *A BCM,* $N \geq 1$ *and a prior* $\mu_0 \in \mathbf{P}(\mathfrak{T})$ *define the* ***Bayesian problem*** $\boldsymbol{BCM}_N(\mu_0)$ as follows:

(i) Find for each initial state $s_0 \in S$ the ***maximal N–stage Bayes reward*** within the set of observation history–dependent policies, i.e.

$$v_N(\mu_0, s_0) := \sup\{v_{N\pi}(\mu_0, s_0) : \pi \in \Delta_N\}.$$

(ii) Find a policy $\pi^* \in \Delta_N$ which is ***Bayes–optimal*** for $\mathrm{BCM}_N(\mu_0)$ in the sense that it maximizes $\pi \mapsto v_{N\pi}(\mu_0, s_0)$ on Δ_N for all $s_0 \in S$.

Example 1: A $\boldsymbol{BCM}_N$ for a Bayesian stopping problem with random initial offer. We now present a BCM_N as above (i.e. with deterministic initial state) which models a Bayesian stopping problem (comprising both the GMKP_N and the asset–selling problem) with a *random* initial offer. The BCM_N is defined as follows: $Z := \mathbb{R}^+$ is the set of offers; $S = Z + \{\overline{s}, \hat{s}\}$; $s_0 \in Z$ denotes an initial offer, while $s_t \in Z$ for some $1 \leq t \leq N$ means that no offer has been accepted before time t and that the offer at time t equals s_t; $s_t = \hat{s}$ for some $1 \leq t \leq N$ means that an offer has been accepted before time t; $s_0 := \hat{s}$ occurs only for formal reasons and may be interpreted that no offers at all are received and no costs arise; the state $\overline{s}$ is crucial for our goal of computing by (2.3) below $d_N(\mu_0)$, the maximal expected discounted reward when N random offers arise and the prior is μ_0. The state $\overline{s}$ can occur only as initial state upon which a transition to some offer $z \in Z$ is enforced. Then the game evolves from z at time $t = 1$ on as it would do from time $t = 0$ on with an offer $z \in Z$. We use $a_t = 1\,(= 0)$ if the offer at time $0 \leq t \leq N-1$ is accepted (rejected). An accepted offer of amount $z \in \mathbb{R}^+$ yields the reward $\psi(\theta) \cdot z$, where $\psi \geq 0$ is a measurable function on Θ; in addition, each offer (not only the rejected ones, as sometimes assumed in the literature) costs $\gamma \geq 0$; the state $\overline{s}$ yields neither a gain nor a cost; $Q(\theta, dz)$ is the p.d. of each of the offers. The evolution of states is determined by a transition function T as follows: if $s_t \in Z$, the next state is $s_{t+1} := T(s_t, a_t, z_{t+1})$; thus the

system moves from a state $s_0 \in Z$ to states $s_1, s_2, \cdots, s_{t-1}$ within Z until either it is "absorbed" in $s_t := \hat{s}$ for some $1 \leq t \leq N$ or until some $s_N \in Z$ is reached (no absorption).

Altogether we have to choose

$$T(s,a,z) = \begin{cases} z \cdot \delta_{a,0} + \hat{s} \cdot \delta_{a,1} & , \text{if } s \in Z, \\ z & , \text{if } s = \overline{s}, \\ \hat{s} & , \text{if } s = \hat{s}; \end{cases}$$

$$r(\theta,s,a) = \begin{cases} \psi(\theta) \cdot s \cdot \delta_{a,1} - \gamma & , \text{if } s \in Z, \\ 0 & , \text{if } s \in \{\overline{s}, \hat{s}\}. \end{cases}$$

$$V_0(\theta,s) = \begin{cases} \psi(\theta) \cdot s - \gamma & , \text{if } s \in Z, \\ 0 & , \text{if } s \in \{\overline{s}, \hat{s}\}. \end{cases}$$

Special cases: For ***asset selling*** one uses arbitrary Θ and Q, while $\psi \equiv 1$. For $\boldsymbol{GMKP}_N$ we use $\Theta := (0,1)$ and $\psi(\theta) := \theta$, $\theta \in \Theta$; $Q(\theta, dz)$ is the uniform p.d. on $(0, 1/\theta)$. □

It is intuitively clear and follows formally from (2.2) that $v_N(\mu_0, \hat{s}) = 0$ and that $v_N(\mu_0, s_0)$ for $s_0 \in Z$ is the *maximal expected discounted reward* when the decision process starts with a ***known*** offer s_0. Moreover, inserting T and r into (2.1) and using (2.2), we see that we have to define

$$d_N(\mu_0) := v_N(\mu_0, \overline{s})/\beta. \tag{2.3}$$

(Here the denominator β takes care of the fact that the rewards, when starting in $s_0 = \overline{s}$, are discounted to time $t = 1$.) The importance of (2.3) stems from the fact that we can use it for computing $d_N(\mu_0)$ within our BCM_N with deterministic initial state. For this purpose we have to find (i) $v_N(\mu_0, \overline{s})$ (for which we also need $v_n(\mu_0, z)$, $1 \leq n \leq N-1, z \in Z$, cf. (2.14) below) and (ii) a policy π^* which is Bayes–optimal for $\mathrm{BCM}_N(\mu_0)$, hence in particular maximizes $\pi \mapsto v_{N\pi}(\mu_0, \overline{s})$ on Δ_N.

A standard approach for solving problem $\mathrm{BCM}_N(\mu_0)$ uses the following device: the prior μ_0 is embedded into a family $\hat{\mu}(i)$, $i \in I$, of p.d.'s on $\mathfrak{T}$. Then, if $\mu_0 = \hat{\mu}(i_0)$ for some $i_0 \in I$, one constructs an MDP′ with state space $I \times S$ and value functions V'_N, $N \geq 1$, such that $V_N(\mu_0, s) = V'_N(i_0, s), s \in S$, and such that a Bayes–optimal $\pi^* \in \Delta_N$ can be constructed (cf. 2.2.(c) below) from a certain N–stage optimal Markovian policy in MDP′.

For a rigorous presentation of this approach we need some preparations. Put $\mathbb{Q}^\mu := \mathbb{Q}^\mu_1$, which is the joint p.d. of the two coordinate variables $\tilde{\theta}$ and

Z_1 on $\Theta \times Z$. A mapping Φ from $\mathbf{P}(\mathfrak{T}) \times Z$ into $\mathbf{P}(\mathfrak{T})$ which transforms each prior μ under the observation z into a version of its posterior

$$\begin{aligned} \Phi(\mu, z, d\theta) &= \mathbb{Q}^{\mu}_{\tilde{\theta}|Z_1}(z, d\theta) \\ &= \text{the conditional p.d. of } \tilde{\theta}, \text{ given } Z_1 = z, \end{aligned}$$

is called a version of the ***Bayes operator*** (generated by Q). It is common to consider $\Phi(\mu, z, d\theta)$ as an update of μ, given the observation z.

The next result is well–known, see e.g. Rieder (1988).

Lemma 2.1. (Computation of the Bayes operator) *Assume that Q has a transition density $q : \Theta \times Z \to \mathbb{R}_+$ w.r. to some σ–finite measure on $\mathfrak{Z}$. For $\mu \in \mathbf{P}(\mathfrak{T})$ and $z \in Z$ put $q_\mu(z) := \int \mu(d\theta')\, q(\theta', z)$. Then*

$$(\mu, z) \mapsto \Phi(\mu, z, B) := \begin{cases} \int_B q(\theta, z)\mu(d\theta)/q_\mu(z),\ B \in \mathfrak{Z}, & \text{, if } 0 < q_\mu(z) < \infty, \\ \mu(B), B \in \mathfrak{Z}, & \text{, otherwise,} \end{cases}$$

is a version of the Bayes operator.

Let $(I,\mathfrak{I})$ be a measurable space and $\hat{\mu}$ a transition p.d. from I into Θ. From now on we consider only the case where μ belongs to the family $\hat{\mu}(i), i \in I$. *For simplicity of notation we often write $v_{N\pi}(i,s)$ and $v_N(i,s)$ instead of $v_{N\pi}(\hat{\mu}(i), s)$ and $v_N(\hat{\mu}(i), s)$, respectively.* Thus (2.2) shows that for $\pi \in \Delta_N$ and all N, i, s there holds

$$\begin{aligned} v_{N\pi}(i,s) &= \int \mathbb{Q}_N^{\hat{\mu}(i)}(d(\theta, z^N))\, G_{N\pi}(\theta, s, z^N), \qquad (2.4) \\ v_N(i,s) &= \sup_{\pi \in \Delta_N} v_{N\pi}(i,s). \end{aligned}$$

We call $v_N : I \times S \to [0, \infty]$ the N*–stage* ***Bayes value function.***

A measurable mapping ϕ from $I \times Z$ into I is called a ***sufficient statistic*** (w.r. to Q) for the family $\hat{\mu}$ if

$$\Phi(\hat{\mu}(i), z, d\theta) = \hat{\mu}(\phi(i,z), d\theta), \quad (i,z) \in I \times Z.$$

Then the posterior of each prior $\hat{\mu}(i)$ belongs to $\hat{\mu}$, and we also call (BCM, $\hat{\mu}, \phi$) a ***Bayesian control model.*** If ϕ is sufficient we define the t–fold iterated *posterior index* $i_t(i, z^t)$, $t \geq 1, (i, z^t) \in I \times Z^t$, of the prior $\hat{\mu}(i)$ recursively, using $i_0(i, z^0) := i$, by

$$i_{t+1}(i, z^{t+1}) := \phi(i_t(i, z^t), z_{t+1}), \quad t \geq 0. \qquad (2.5)$$

Example 2. The model $(BCM, \hat{\mu}, \phi)$ for the GMKP. From the BCM in Example 1, specialized to the GMKP, we obtain a model $(BCM, \hat{\mu}, \phi)$ with $\hat{\mu}$ and ϕ as follows. Put $I := \mathbb{R}^+ \times (0,1]$ with elements $i = (\alpha, b)$ and endowed with its Borel–σ–algebra. Let $\hat{\mu}(\alpha, b)$ be the ***power p.d.*** $Pow(\alpha, b)$ on $\mathfrak{T}$, defined by the density

$$\theta \mapsto h_{\alpha,b}(\theta) := \alpha \cdot \theta^{\alpha-1} \cdot 1_{(0,b)}(\theta)/b^{\alpha}. \tag{2.6}$$

(It is easily seen that $\hat{\mu}(\alpha, b, B)$ is measurable in (α, b)). Note that $U(0,1)$ is embedded as $\hat{\mu}(1,1)$ into the family $\hat{\mu}$.

A simple computation derives from 2.1 that $\Phi(\hat{\mu}(\alpha, b), z)$ has the density $h_{\alpha+1, b\wedge(1/z)}$. This verifies sufficiency of the measurable mapping

$$(\alpha, b, z) \mapsto \phi(\alpha, b, z) := (\alpha + 1, b \wedge (1/z)). \tag{2.7}$$

Note that $\phi(\alpha, b, z) \in I$ since $b \leq 1$. Put

$$M_t(b, z^t) := \min(b, 1/z_1, 1/z_2, \cdots, 1/z_t), \quad 1 \leq t \leq N, (b, z^t) \in (0,1] \times Z^t.$$

Then induction on $t \geq 1$ shows that the posterior index $i_t(\alpha, b, z^t)$ is given by

$$i_t(\alpha, b, z^t) = (\alpha + t, M_t(b, z^t)), \quad t \geq 1, (\alpha, b, z^t) \in I \times Z^t. \tag{2.8}$$

By the way, this also shows that power p.d.'s are natural in GMKP: if one starts as in the MKP with a uniform prior, all posteriors are power distributions. □

Let $\mathbf{F}'$ be the set of measurable mappings f' from $I \times S$ into $\{0,1\}$. Thus $(\mathbf{F}')^N$ is the set of N–stage Markovian policies in MDP'. If ϕ is sufficient we say that $\pi^i = (\pi_t^i)_{t=0}^{N-1} \in \Delta_N$ is *generated* by i and by $\pi' = (\pi_t')_0^{N-1} \in (\mathbf{F}')^N$ if $\pi_0^i(s_0) := \pi_0'(i, s_0)$, $s_0 \in S$, and $\pi_t^i(z^t, s_t) := \pi_t'(i_t(i, z^t), s_t)$, $1 \leq t \leq N-1, (z^t, s_t) \in Z^t \times S$.

Let W be a mapping from $I \times D$ into $[-\infty, \infty]$. A mapping $f' \in \mathbf{F}'$ is called a *maximizer of* W, if $f'(i, s)$ is a maximum point of $a \mapsto W(i, s, a)$ for all (i, s). The next result can be derived from known results in the literature; closest to our setting are Satz 3.2 and Satz 3.4 in Rieder (1988).

Theorem 2.2. (The Basic Theorem for a binary Bayesian control model (BCM,$\hat{\mu}, \phi$)) *For* $(i,s,a) \in I \times D$ *put*

$$\begin{aligned} Q'(i,B) &:= \int \hat{\mu}(i,d\theta)\, Q(\theta,B), B \in \mathfrak{Z}, \\ r'(i,s,a) &:= \int \hat{\mu}(i,d\theta)\, r(\theta,s,a),\ v_0(i,s) := \int \hat{\mu}(i,d\theta)\, V_0(\theta,s), \\ Lv(i,s,a) &:= r'(i,s,a) + \beta \cdot \int Q'(i,dz)\, v(\phi(i,z), T(s,a,z)). \end{aligned}$$

(These terms are measurable in i*, in* (i,s) *and in* (i,s,a)*, respectively.) Then there holds:*

(a) *The Bayes value functions are measurable and lower bounded, and they can be found recursively by the value iteration*

$$\begin{aligned} v_n(i,s) &= \max[Lv_{n-1}(i,s,0),\ Lv_{n-1}(i,s,1)], \\ &\qquad n \geq 1, (i,s) \in I \times S. \end{aligned} \tag{2.9}$$

(b) *There exists for each* $n \geq 1$ *a (pointwise) smallest maximizer of* Lv_{n-1}.

(c) *Fix* $N \geq 1$. *Let* f'_n *be a maximizer of* Lv_{n-1}, $1 \leq n \leq N$. *Then for fixed* $i \in I$ *the observation history-dependent policy* $\pi^i = (\pi^i_t)_{t=0}^{N-1} \in \Delta_N$ *which is generated by* i *and* $(\pi'_t)_0^{N-1} := (f'_N, f'_{N-1}, \cdots, f'_1)$ *is Bayes-optimal for problem* $BCM_N(\hat{\mu}(i), \phi)$. □

The core of each policy $\pi \in \Delta_N$, when applied for initial state $\overline{s}$, may be described by its *stopping time*, i.e. the time at which π prescribes to accept the momentary offer. It is defined by

$$z^N \mapsto \tau_{N\pi}(z^N) := \min\{1 \leq t \leq N-1 : \pi_t(z^t, z_t) = 1\}, \tag{2.10}$$

where $\min \emptyset := N$. (Note that $\tau_{N\pi}$ does not depend on π_0.) It is intuitively clear that, using $\tau_\pi := \tau_{N\pi}(z^N)$, there holds for all $N, \pi \in \Delta_N$ and (θ, z^N)

$$G_{N\pi}(\theta, \overline{s}, z^N) = \beta^{\tau_\pi} \cdot \psi(\theta) \cdot z_{\tau_\pi} - \gamma \cdot \beta \cdot \sigma_{\tau_\pi}(\beta). \tag{2.11}$$

Thus it suffices to find an *optimal stopping time* τ^*_N, defined as the stopping time $\tau_{N\pi^*}$ of any Bayes optimal $\pi^* \in \Delta_N$. We sketch a proof of (2.11): Fix z^N and put $m := \tau_\pi(z^N)$. From the definition of τ_π we get $\pi_\nu(z^\nu, z_\nu) = 0$ for $1 \leq \nu \leq m-1$, and $\pi_m(z^m, z_m) = 1$ if $m < N$. Next, the form of T implies that $\zeta_{\nu\pi}(\overline{s}, z^\nu) = z_\nu$, $1 \leq \nu \leq m$, and $\zeta_{\nu\pi}(\overline{s}, z^\nu) = \hat{s}$ for $m+1 \leq \nu \leq N$. Now the assertion follows from (2.1) and the form of r. □

In the next result note that $d_1(i)$ equals the expected reward one gets under prior $\hat{\mu}(i)$ when one obtains a single offer with cost γ which is accepted.

Proposition 2.3. (The stopping problem of Example 1 with random initial offer) *Consider the model $(BCM, \hat{\mu}, \phi)$ for Example 1. Assume that $G(i) := \int \hat{\mu}(i, d\theta)\, \psi(\theta)$, $i \in I$, is finite. Fix some $i_0 \in I$.*

(a) *The maximal expected reward $d_N(i_0)$ for prior $\hat{\mu}(i_0)$ can be obtained recursively, starting with $d_1(i) := \int Q'(i, dz)\, z \cdot G(\phi(i,z)) - \gamma$, $i \in I$, as follows: For $n \geq 1, i \in I$, there holds*

$$d_{n+1}(i) = \int Q'(i, dz)\ \max\left[\beta \cdot d_n(\phi(i,z)),\ z \cdot G(\phi(i,z))\right] - \gamma. \tag{2.12}$$

(b) *For prior $\hat{\mu}(i_0)$ the mapping*

$$z^N \mapsto \tau_N^*(i_0, z^N) := \min\{1 \leq t \leq N-1 : \beta \cdot d_{N-t}(i_t(i_0, z^t)) < z_t \cdot G(i_t(i_0, z^t))\}, \tag{2.13}$$

where $\min \emptyset := N$, is an optimal stopping time.

Proof. We apply 2.2 to the $(BCM, \hat{\mu}, \phi)$ from Example 1.

(a) Firstly for $i \in I$ and $s \in Z$ we have $r'(i,s,1) = v_0(i,s) = s \cdot G(i) - \gamma$, $r'(i,s,0) = -\gamma$, and $r'(i, \overline{s}, a) = r'(i, \hat{s}, a) = v_0(i, \overline{s}) = v_0(i, \hat{s}) = 0$, $a \in \{0,1\}$. From (2.9) we obtain for $n \geq 1, i \in I$ that $v_n(i, \hat{s}) = \beta \cdot \int Q'(i, dz)\, v_{n-1}(\phi(i,z), \hat{s})$, which implies by induction on $n \geq 0$ that $v_n(i, \hat{s}) = 0$, $n \geq 0$. Now one obtains from (2.9) for $i \in I$ and $n \geq 1$

$$\begin{aligned} d_n(i) = v_n(i, \overline{s})/\beta &= \int Q'(i, dz)\, v_{n-1}(\phi(i,z), z)), \\ v_{n-1}(i,z) &= \max\left[\beta \cdot d_{n-1}(i), z \cdot G(i)\right] - \gamma, \quad z \in Z. \end{aligned} \tag{2.14}$$

The combination of these two equations yields (2.12).

(b) The proof of (2.14) also shows that for fixed $N \geq 1$ the mapping f'_n, defined by $f'_n(i, \overline{s}) := f'_n(i, \hat{s}) := 0$ (= reject) and

$$f'_n(i,z) := 1 \text{ iff } \beta \cdot d_n(i) < z \cdot G(i), \quad z \in Z, i \in I, \tag{2.15}$$

is the smallest maximizer of Lv_{n-1}, $1 \leq n \leq N$. Then the policy $\pi^* = (\pi_t^*)_0^{N-1} \in \Delta_N$, generated by i_0 and by $(f'_n)_1^N$, is Bayes–optimal by 2.2(b). For $1 \leq t \leq N-1$ and $z^t \in Z^t$ we have $\pi_t^*(z^t, z_t) = f'_{N-t}(i_t(i_0, z^t), z_t)$. Now the assertion follows from (2.15) and the definition $\tau_N^* := \tau_{N\pi^*}$. □

3 Solution of the Generalization of MacKinnon's Problem

In this section we consider the GMKP. The maximal expected discounted n–stage reward d_n clearly depends on the data parameter tripel $x := (\alpha, \beta, \gamma) \in \mathbb{R}^+ \times (0,1] \times \mathbb{R}_+$. In 3.1 through 3.3 we keep β and γ fixed, and write then $d_n(\alpha)$ instead of $d_n(x)$.

One cannot expect that (2.12) leads to an explicit representation of $d_N(\alpha, b)$. Also a numerical solution seems to be impossible since $d_N(\alpha, b)$ requires due to (2.12) to find $d_n(i_{N-n}(\alpha, b, z^{N-n}))$ for $1 \leq n \leq N-1$ and for the uncountable set of observation–histories $z^{N-n} \in Z^{N-n}$. Fortunately, it turns out (see 3.3(a) below) that $d_n(\alpha, b)$, $n \geq 1, (\alpha, b) \in I$, does not depend on b, hence, due to (2.5) and (2.7), $d_n(i_{N-n}((\alpha, b, z^{N-n}))$ does not depend on (b, z^{N-n}). This is the reason why we can find in (3.7) a relatively simple recursion for $d_n(\alpha)$ and in 3.3(d) the optimal stopping time. We need some preparations. Recall the definition $M_t(b, z^t) := \min(b, 1/z_1, 1/z_2, \cdots, 1/z_t) \leq 1$, $1 \leq t \leq N, (b, z^t) \in (0,1] \times Z^t$, and put $Q'(\alpha, dz) := Q'(\alpha, 1, dz)$, $\alpha \in \mathbb{R}^+$.

A simple integration, using from 2.3 the definition of G and from (2.6) the density $h_{\alpha,b}$ of the power distribution $Pow(\alpha, b)$, yields

$$\begin{aligned} G(\alpha, b) &= g(\alpha) \cdot b, \\ \text{where } g(\alpha) &:= \alpha/(\alpha+1), \quad (\alpha, b) \in \mathbb{R}^+ \times (0,1]. \end{aligned} \tag{3.1}$$

Lemma 3.1. (a) *$Q'(\alpha, b, dz)$ has the distribution function*

$$z \mapsto F_{\alpha,b}(z) := \tag{3.2}$$
$$\begin{cases} b \cdot g(\alpha) \cdot z^+ & , \quad \text{if } -\infty < z \leq 1/b, \\ 1 - (\alpha+1)^{-1} \cdot (bz)^{-\alpha} & , \quad \text{if } z > 1/b. \end{cases}$$

Moreover, $Q'(\alpha, b, dz)$ is stochastically decreasing in α.

(b) *For each measurable $v \geq 0$ on $\mathbb{R}^+$ there holds*

$$\int Q'(\alpha, b, dz)\, v(z) =$$
$$g(\alpha) \cdot \left[\int_0^1 v(z/b)\, dz + \int_1^\infty v(z/b)/z^{\alpha+1}\, dz \right] \tag{3.3}$$
$$\int Q'(\alpha, b, dz)\, v(z) = \int Q'(\alpha, dz)\, v(z/b). \tag{3.4}$$

Proof. (a) Fix (α, b) and $z \in \mathbb{R}$. We have $Q(\theta, (-\infty, z]) = (\theta z^+) \wedge 1$ and

$$F_{\alpha,b}(z) = Q'(\alpha, b, (-\infty, z]) = \int Pow(\alpha, b, d\theta)\, Q(\theta, (-\infty, z]).$$

Thus $F_{\alpha,b}(z) = 0, z \leq 0$, and $F_{\alpha,b}(z) = \alpha \cdot b^{-\alpha} \cdot \int_0^b \theta^{\alpha-1} \cdot (\theta z) \wedge 1 \, d\theta, z > 0$. This easily yields $F_{\alpha,b}(z)$ for $0 < z \leq 1/b$, and also, splitting $\int_0^b \theta^{\alpha-1} \cdot (\theta z) \wedge 1 \, d\theta$ into $\int_0^{1/z}(\cdots) + \int_{1/z}^b(\cdots)$, for $z > 1/b$.

In addition, $\alpha \mapsto F_{\alpha,b}(z)$ is increasing for all $z \in \mathbb{R}$, which proves that $\alpha \mapsto Q'(\alpha, b, dz)$ is stochastically decreasing.

(b) From (a) we see that $Q'(\alpha, b, dz)$ has the density $y \mapsto b \cdot g(\alpha)/(b \cdot \max(y, 1))^{\alpha+1} \cdot 1_{(0,\infty)}(y)$. Therefore

$$\int Q'(\alpha, b, dz)\, v(z) = \\ b \cdot g(\alpha) \cdot \left[\int_0^{1/b} v(y)\, dy + \int_{1/b}^{\infty} v(y)/(by)^{\alpha+1}\, dy \right].$$

Now the substitution $z = by$ yields (3.3). Finally (3.4) follows from (3.3) by first setting $b := 1$ and then replacing $v(z)$ by $v(z/b)$. □

The proof of the following auxiliary result is given in the appendix.

Lemma 3.2. *Let $c(\cdot) \geq 0$ be a function on $\mathbb{R}^+$ and put*

$$E(\alpha) := \int Q'(\alpha, dz) \max[c(\alpha), g(\alpha+1) \cdot (z \wedge 1)], \alpha \in \mathbb{R}^+.$$

Then there holds:

(a)

$$E(\alpha) = \begin{cases} \frac{1}{2} \cdot (1 - (\alpha+1)^{-2}) \cdot c(\alpha)^2 + \frac{1}{2} & , \text{if } c(\alpha) \leq g(\alpha+1), \\ c(\alpha) & , \text{if } c(\alpha) \geq g(\alpha+1). \end{cases} \tag{3.5}$$

(b) *If $c(\cdot)$ is increasing then $E(\cdot)$ is increasing.*

(c) *If $c(\cdot)$ is continuous then $E(\cdot)$ is continuous.*

Theorem 3.3. (The solution of the generalized problem of MacKinnon) *Assume $N \geq 2, \beta \in (0,1], \gamma \in \mathbb{R}_+$ and an arbitrary power distribution $Pow(\alpha, b)$, $(\alpha, b) \in \mathbb{R}^+ \times (0,1]$ as prior.*

(a) *There holds $d_1(\alpha, b) = 1/2 - \gamma =: d_1(\alpha)$. The maximal expected discounted N-stage reward $d_N(\alpha, b) =: d_N(\alpha)$ does not depend on b, it is finite and $(d_n)_2^\infty$ satisfies the recursion*

$$d_{n+1}(\alpha) = \int Q'(\alpha, dz) \cdot \max[\beta \cdot d_n(\alpha+1), g(\alpha+1) \cdot (z \wedge 1)] - \gamma, \\ n \geq 1, \alpha \in \mathbb{R}^+. \tag{3.6}$$

(b) *If* $\gamma \geq 1/2$ *then* $d_N(\alpha) = 1/2 - \gamma$ *for all* α.

(c) *If* $\gamma \leq 1/2$ *then* $d_N(\alpha)$ *can be computed recursively for all* α *as follows: for* $n \geq 1$ *there holds:*

$$d_{n+1}(\alpha) = \begin{cases} \frac{1}{2} \cdot (1 - (\alpha+1)^{-2}) \cdot (\beta \cdot d_n(\alpha+1))^2 + \frac{1}{2} - \gamma, \\ \quad \text{if } \beta \cdot d_n(\alpha+1) \leq g(\alpha+1), \\ \beta \cdot d_n(\alpha+1) - \gamma, \\ \quad \text{if } \beta \cdot d_n(\alpha+1) \geq g(\alpha+1). \end{cases} \tag{3.7}$$

(d) *An optimal stopping time is given, using* $\min \emptyset := N$, *by*

$$\begin{aligned} z^N \mapsto \tau_N^*(\alpha, b, z^N) &= \min\{1 \leq t \leq N-1 : \\ \beta \cdot d_{N-t}(\alpha+t) &< z_t \cdot g(\alpha+t) \cdot M_t(b, z^t)\}. \end{aligned} \tag{3.8}$$

In particular, if $\gamma \geq 1/2$ *then* $\tau_N^*(\alpha, b, z^N) = 1$ *for all* (α, b).

Proof. (a1) We prove that $d_1(\alpha, b) = 1/2 - \gamma = d_1(\alpha)$. From 2.3, (3.1), (2.7) and (3.3) we get for $(\alpha, b) \in I$

$$\begin{aligned} d_1(\alpha, b) + \gamma &= \int Q'(\alpha, b, dz)\, z \cdot G(\phi(\alpha, b, z)) \\ &= g(\alpha+1) \cdot \int Q'(\alpha, b, dz)\, (bz \wedge 1) \\ &= g(\alpha+1) \cdot g(\alpha) \cdot \left[\int_0^1 z\, dz + \int_1^\infty dz/z^{\alpha+1} \right] = 1/2. \end{aligned}$$

(a2) We verify by induction on $n \geq 1$ the assertion (I_n) that d_n is independent of b and that $d_{n+1}(\alpha)$ equals the r.h.s. of (3.6). For the proof we note that by (2.12), (2.7), (3.1) and (3.4) there holds $d_{n+1}(\alpha, b) = \int Q'(\alpha, dz)\, v_n(z/b) - \gamma$, where $v_n(z) := \max\,[\beta \cdot d_n(\alpha+1, b \wedge (1/z)), g(\alpha+1) \cdot (bz \wedge 1)]$, $n \geq 1$. Now (I_1) holds by (a1), and (I_n) for some $n \geq 1$ easily implies (I_{n+1}).

(a3) Finiteness of d_n holds since a simple proof, using (3.6), shows by induction on $n \geq 1$ that $-\gamma \leq d_n \leq 1$.

(b) Using $d_1(\alpha)$ from (a), we prove by induction on $n \geq 1$ the assertion (I_n) that $d_n(\alpha) = 1/2 - \gamma$. (I_1) holds by definition of $d_1(\alpha)$. Assume (I_n) for some $n \geq 1$. Then $\beta \cdot d_n(\alpha+1) = \beta \cdot (1/2 - \gamma) \leq 0 \leq g(\alpha+1) \cdot z \wedge 1$. Now (3.6) yields $d_{n+1}(\alpha) = g(\alpha+1) \cdot \int Q'(\alpha, dz)\, z \wedge 1 - \gamma$, which equals $1/2 - \gamma$ as shown in (a1).

(c) We obtain (3.7) from (3.6) by applying 3.2(a) with $c(\alpha) := \beta \cdot d_n(\alpha+1) \geq 0$, since then $d_{n+1} = E(\alpha) - \gamma$.

(d) Equation (3.8) follows, since $d_1(\alpha) := 1/2-\gamma$ by (a), from (b), 2.3(b), (2.8) and (3.1) with α replaced by $\alpha + t$, and b replaced by $M_t(b, z^t)$. The assertion in case $\gamma \geq 1/2$ holds since then $d_{N-t}(\alpha + t) \leq 0$. □

Remark 1. (a) Part (b) of 3.3 expresses the remarkable fact (quite different from the asset selling problem) that for "large" costs γ (namely $\gamma \geq 1/2$) having several chances to stop is no advantage over the case where only a single offer arises.

(b) In contrast to d_N the optimal stopping time τ_N^* *does* depend on b, and its computation requires $(d_n(\alpha+N-n))_{n=1}^{N-1}$, which is known from (b) and (c).

(c) Our numerical computation of d_n in Table 1 below is based on the recursion (3.7), while both (3.6) and (3.7) are used for studying the dependence of d_n on n and on the parameters α, β and γ in 3.4 and 3.5 below.

(d) If $\gamma = 0$ then there also holds

$$d_N(\alpha) = \beta^{N-n^*(N,\alpha)} \cdot d_{n^*(N,\alpha)}(\alpha + N - n^*(N,\alpha)), \tag{3.9}$$

where $n^*(N,\alpha)$ is the smallest $1 \leq n \leq N-1$ for which $\beta \cdot d_n(\alpha+N-n) \geq (\alpha+N-n)/(\alpha+N-n+1)$, if such an n exists, and $n^*(N,\alpha) := N$, else.

For the *proof* put $k := \alpha+N-n-1$. If $\beta \cdot d_n(\alpha+N-n) \geq (k+1)/(k+2)$, then the second line of (3.7) implies $\beta \cdot d_{n+1}(\alpha+N-n-1) \geq k/(k+1)$. Thus $d_{n+1}(\alpha+N-n-1) = \beta \cdot d_n(\alpha+N-n)$ for $n^*(N,\alpha) \leq n \leq N-1$. This easily implies (3.9). □

We now study the dependence of the maximal expected discounted n–stage reward $d_n(x)$ on n and on $x := (\alpha, \beta, \gamma) \in X := \mathbb{R}^+ \times (0,1] \times [0,1/2]$. Note the difference between the dependence on α on the one hand and the dependence on β and γ on the other hand: While $d_n(x)$ is convex in β and also in γ, Figure 1 shows that in general $d_n(x)$ is neither convex nor concave in α.

As usual, for a function u on $\mathbb{R}$ the smallest number $l \in [0,\infty]$ such that $|u(x)-u(y)| \leq l \cdot |x-y|$ for all $x, y \in \mathbb{R}$ is called the Lipschitz module $l(u)$ of u, and u is said to be Lipschitz continuous if $l(u)$ is finite.

Proposition 3.4. (The dependence of the functions $d_n(\cdot)$ on $n \geq 1$ and $x = (\alpha, \beta, \gamma) \in \mathbb{R}^+ \times (0,1] \times [0,1/2]$) *There holds:*

(a) $d_n(x)$ *is increasing in* n *for* $x \in X$.

Moreover, there holds for $n \geq 1$*:*

(b) $0 \leq 1/2-\gamma = d_1(x) < 1-\gamma$, *and* $0 \leq 1/2-\gamma \leq d_{n+1}(x) \leq g(\alpha+n)-\gamma < 1-\gamma$.

(c) $d_n(x)$ *is increasing and continuous in* α, *hence there exists* $u_n(\beta,\gamma) := \lim_{\alpha\to\infty} d_n(x) = \sup_{\alpha\in\mathbb{R}^+} d_n(x)$ *and* $e_n(\beta,\gamma) := \lim_{\alpha\to 0} d_n(x) = \inf_{\alpha\in\mathbb{R}^+} d_n(x)$.

(d) *The numbers* $u_n(\beta,\gamma)$, *can be found recursively, using* $u_1(\beta,\gamma) = 1/2-\gamma$, *by*

$$u_{n+1}(\beta,\gamma) = (\beta\cdot u_n(\beta,\gamma))^2/2 + 1/2 - \gamma. \tag{3.10}$$

Moreover, using $\inf\ \emptyset := \infty$, *we get* $2 \le n_0 := \inf\{k\in\mathbb{N} : \beta\cdot d_k(1) > 1/2\} \le \infty$, *and for* $n \ge 1$ *we have*

$$e_{n+1} = \begin{cases} 1/2-\gamma & , \text{if } 1\le n < n_0, \\ \beta\cdot d_n(1)-\gamma & , \text{if } n\ge n_0. \end{cases}$$

In addition, $e_{n+1} = 1/2-\gamma$ *for all* $n\ge 1$ *if* $\beta\cdot(5-8\gamma)\le 4$.

(e) $d_n(x)$ *is increasing and convex in* β.

(f) $d_n(x)$ *is Lipschitz continuous (hence uniformly continuous) in* β, *and* $l(d_n(\alpha,\cdot,\gamma)) \le (n-1)\cdot(1-\gamma)$.

(g) $d_n(x)$ *is decreasing and convex in* γ.

(h) $d_n(x)$ *is Lipschitz continuous (hence uniformly continuous) in* γ, *and* $l(d_n(\alpha,\beta,\cdot)) \le \sigma_n(\beta)$.

Table 1: Some values of $d_n(1,1,0)$

n	2	3	5	10	20	30
$d_n(1,1,0)$	0.5938	0.6400	0.7056	0.7966	0.8739	0.9083

Remark 2. (a) It follows from 3.3(b) and 3.4(b) that for all $N\ge 2$ and all $x\in X$ the N–stage game is favorable, fair or unfavorable, respectively, for the contestant (i.e. $d_N(x)$ is positive, zero or negative) iff $0\le\gamma<1/2, \gamma = 1/2$ or $\gamma > 1/2$.

(b) $d_n(x)$ is for $\gamma < 1/2$ and all $n\ge 1$ only "moderately" increasing in α in the sense that for all α the relative increase $[d_n(\alpha+\delta)-d_n(\alpha)]/d_n(\alpha)$ tends to zero for $\alpha\to\infty$, uniformly w.r. to $\delta\in\mathbb{R}^+$. In fact, we show that

$$\frac{d_n(\alpha+\delta)-d_n(\alpha)}{d_n(\alpha)} \le 1/\alpha, \quad n\ge 1, \alpha,\delta\in\mathbb{R}^+. \tag{3.11}$$

Proof. (i) Fix (β,γ). We prove by induction on $n\ge 1$ the assertion (I_n) that $f_n(\alpha) := \beta\cdot d_n(\alpha)/g(\alpha)$ is decreasing in α. Note that $d_n \ge d_1 > 0$

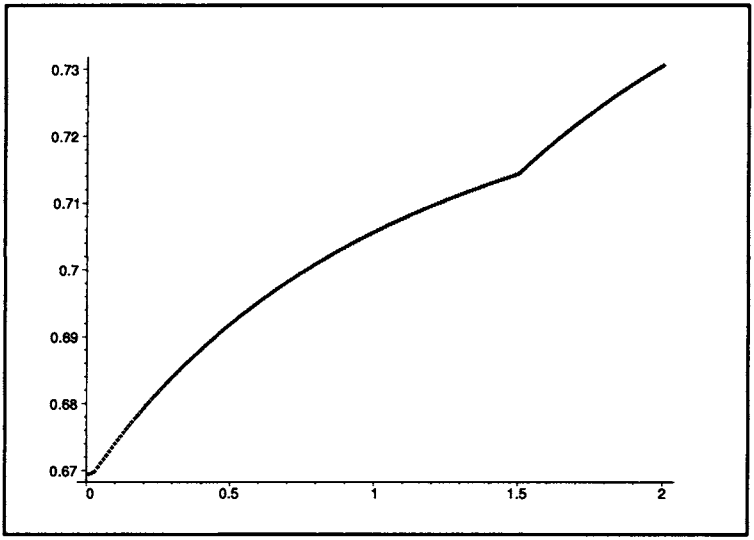


Figure 1: $\alpha \to d_5(\alpha, 1, 0)$

since $0 \leq \gamma < 1/2$. (I_1) holds since $g(\cdot)$ is increasing. Assume (I_n). Then, by a simple calculation one derives from (3.6) that

$$f_{n+1}(\alpha) = \beta \cdot g(\alpha+1) \cdot g(\alpha)^{-1} \cdot \int Q'(\alpha, dz)\ v(\alpha, z) - \gamma,$$

where $v(\alpha, z) := \max\,[f_n(\alpha+1), z \wedge 1] \geq 0$. Let $\alpha < \overline{\alpha}$. Observing that $v(\overline{\alpha}, z) \leq v(\alpha, z)$, that $v(\alpha, z)$ is increasing in z and that $\alpha \mapsto Q'(\alpha, 1, dz)$ is stochastically decreasing by 3.1(a), we obtain

$$\int Q'(\overline{\alpha}, dz)\ v(\overline{\alpha}, z) \leq \int Q'(\overline{\alpha}, dz)\ v(\alpha, z) \leq \int Q'(\alpha, dz)\ v(\alpha, z).$$

Thus $\int Q'(\alpha, dz)\ v(\alpha, z) \geq 0$ is decreasing in α. Then (I_{n+1}) follows since also $0 \leq g(\alpha+1)/g(\alpha) = (1 + [\alpha \cdot (\alpha+2)]^{-1})$ is decreasing in α.

(ii) Now (3.11) follows from antitonicity of $\alpha \mapsto f_n(\alpha)$ by a simple calculation. □

Proof of 3.4. (a) Using 3.3(c) we prove by induction on $n \geq 1$ the assertion (I_n) that $d_{n+1}(\cdot) \geq d_n(\cdot)$. Fix $x \in X$. For the proof of (I_1) recall that $d_1(\alpha) = 1/2 - \gamma$.

Case 1: $\beta d_1(\alpha+1) < g(\alpha+1)$. Then we get from (3.7)

$$d_2(\alpha) = \frac{1}{2} \cdot (1 - (\alpha+1)^{-2}) \cdot (\beta \cdot d_1(\alpha+1))^2 + d_1(\alpha) \geq d_1(\alpha).$$

Case 2: $\beta d_1(\alpha+1) \geq g(\alpha+1) = 1 - 1/(\alpha+2)$. Then $d_2(\alpha) = \beta \cdot d_1(\alpha+1) - \gamma \geq 1 - 1/(\alpha+2) - \gamma > 1/2 - \gamma = d_1(\alpha)$. Thus (I_1) is true.

If (I_n) holds for some $n \geq 1$ then also (I_{n+1}) holds by (3.6).

(b) The bounds for d_1 hold since $d_1(x) = 1/2 - \gamma$ by 3.3(a). Now we turn to the bounds for $d_{n+1}, n \geq 1$. The lower bound holds by (a), and $g(\alpha+n) - \gamma < 1 - \gamma$ is obvious. It remains to verify the assertion (I_n), $n \geq 1$, that $d_{n+1}(\alpha) \leq g(\alpha+n) - \gamma$ for all α, which is done by induction on n. Firstly note that $1/2 < g(\alpha+1) < 1$ for all α and that $g(\cdot)$ is increasing. Next, due to $g(\alpha+1) \cdot (z \wedge 1) \leq g(\alpha+n)$, we see from (3.6) that

$$d_{n+1}(\alpha) \leq \max\left[\beta \cdot d_n(\alpha+1), g(\alpha+n)\right] - \gamma, \quad n \geq 1, \tag{3.12}$$

which verifies (I_1) since

$$\begin{aligned} \max\left[\beta \cdot d_1(\alpha+1), g(\alpha+1)\right] &\leq \max\left[\beta \cdot (1/2 - \gamma), g(\alpha+1)\right] \\ &\leq \max\left[1/2, g(\alpha+1)\right] = g(\alpha+1). \end{aligned}$$

It also shows that (I_n) implies (I_{n+1}), since (I_n) implies $\beta \cdot [d_{n+1}(\alpha+1) - \gamma] \leq g(\alpha+1+n)$.

(c) The assertion (I_n) is true for $n = 1$ since d_1 is constant in α. Assume that (I_n) holds for some $n \geq 1$. Now the assertion follows from 3.2(b) and 3.2(c) with $c(\alpha) := \beta \cdot d_n(\alpha+1)$ since then $d_{n+1}(\alpha) = E(\alpha) - \gamma$ by (3.6). Note that $c(\alpha) \geq 0$ by (b).

(d1) We prove the assertion about u_{n+1}. Fix some $n \geq 1$. From (3.6) and (3.3) with $b := 1$ we get, using $v(\alpha, z) := \max\left[\beta \cdot d_n(\alpha+1), g(\alpha+1) \cdot (z \wedge 1)\right]$,

$$\begin{aligned} d_{n+1}(\alpha) + \gamma &= \int Q'(\alpha, dz)\, v(\alpha, z) \\ &= g(\alpha) \cdot \left[\int_0^1 v(\alpha, z)\, dz + \int_1^\infty v(\alpha, z)/z^{\alpha+1}\, dz\right]. \end{aligned}$$

We have $0 \leq v(\alpha, z) \leq 1$ since $d_n \leq 1$ by (b). Now

$$\int_1^\infty v(\alpha, z)/z^{\alpha+1}\, dz \leq \int_1^\infty 1/z^{\alpha+1}\, dz = 1/\alpha \to 0 \quad \text{for } \alpha \to \infty.$$

Moreover $v(\alpha, z)$ converges for $\alpha \to \infty$ increasingly towards $v(z) := \max\left[\beta \cdot u_n, z \wedge 1\right]$. By the theorem on the convergence of integrals $\int_0^1 v(\alpha, z)\, dz$ converges for $\alpha \to \infty$ to

$$\int_0^1 v(z)\, dz = \int_0^{\beta u_n} \beta u_n\, dz + \int_{\beta u_n}^1 z\, dz = (\beta^2 u_n^2 + 1)/2.$$

Here we used that $0 \le u_n \le 1$ since $0 \le d_n \le 1$. Now the proof is complete.

(d2) We prove the assertion about e_{n+1}.

(d2i) We have $\beta \cdot d_1(1) = \beta \cdot (1/2 - \gamma) \le 1/2$, hence $2 \le n_0 \le \infty$.

(d2ii) Fix $n \ge 1$. We show that $e_{n+1} = \max[\beta \cdot d_n(1), 1/2] - \gamma$. In the proof we use without mention that $d_n(x)$ is increasing and continuous in α by (c). Let $(\alpha_k)_1^\infty \subset \mathbb{R}^+$ be a sequence converging for $k \to \infty$ to zero. Then there holds at least one of the following cases:

Case 1: The set $B_\le := \{\alpha_k : \beta \cdot d_n(\alpha_k + 1) \le g(\alpha_k + 1)\}$ is infinite. Using (3.7) and letting $B_\le \ni \alpha_k \to 0$ we obtain $e_{n+1} = 1/2 - \gamma$. This equals $\max[\beta \cdot d_n(1), 1/2] - \gamma$ since convergence of $g(\alpha_k)$ to $1/2$ for $\alpha_k \to 0$ yields $\beta \cdot d_n(1) \le 1/2$.

Case 2: The set $B_> := \{\alpha_k : \beta \cdot d_n(\alpha_k + 1) > g(\alpha_k + 1)\}$ is infinite. Using (3.7) and letting $B_> \ni \alpha_k \to 0$ we obtain $e_{n+1} = \beta \cdot d_n(1) - \gamma$. This equals $\max[\beta \cdot d_n(1), 1/2] - \gamma$ since convergence of $g(\alpha_k)$ to $1/2$ for $\alpha_k \to 0$ yields $\beta \cdot d_n(1) \ge 1/2$.

(d2iii) Now (d2ii) and (a) verifies the value given for e_{n+1}.

(d2iv) The assertion that $e_{n+1} = 1/2 - \gamma$ if $\beta \cdot (5 - 8\gamma) \le 4$ requires 3.5(d) below and is therefore included in the proof of 3.5(d).

(e) The assertion (I_n) is trivially true for $n = 1$. Assume that it is true for some $n \ge 1$ and fix (α, γ). Then $\beta \mapsto \beta \cdot d_n(\alpha + 1, \beta, \gamma)$ is increasing and convex since the product of the two non–negative, increasing and convex functions $\beta \mapsto \beta$ and $\beta \mapsto d_n(\alpha, \beta, \gamma)$ on $(0, 1]$ is increasing and convex; see e.g. Roberts/Varberg (1970), Theorem 13 C. Now (I_{n+1}) follows from (3.6) since the maximum of the two increasing and convex functions $\beta \cdot d_n(\alpha, \beta, \gamma)$ and $\beta \mapsto g(\alpha + 1) \cdot (z \wedge 1)$ is increasing and convex and since this property is preserved under integration by $Q'(\alpha, dz)$.

(f) Fix γ. We use induction on $n \ge 1$ for the assertion (I_n) that $l(d_n(\alpha, \cdot, \gamma)) \le (n - 1) \cdot (1 - \gamma)$ for all α. Obviously (I_1) is true. Assume (I_n) for some $n \ge 1$. Put $u(\beta) := \beta$ and $h(\beta) := d_n(\alpha + 1, \beta, \gamma)$. Using $|\max(x_1, x_2) - \max(y_1, y_2)| \le \max[|x_1 - y_1|, |x_2 - y_2|]$ for $x_1, x_2, y_1, y_2 \in \mathbb{R}$ we get from (3.6) for $\beta, \beta' \in (0, 1]$

$$|d_{n+1}(\alpha, \beta, \gamma) - d_{n+1}(\alpha, \beta', \gamma)|$$
$$\le \int Q'(\alpha, dz)\, |u(\beta) \cdot h(\beta) - u(\beta') \cdot h(\beta')| \le l(u \cdot h) \cdot |\beta - \beta'|.$$

From (b) we know that $0 \le h(\cdot) \le 1 - \gamma$, which implies $\sup_\beta h(\beta) \le 1 - \gamma$. Next, from a well–known and easy to prove upper bound for the Lipschitz

module of the product of two functions we get

$$\begin{aligned} l(d_{n+1}(\alpha,\cdot,\gamma)) &\leq l(u\cdot h) \leq \sup_\beta |u(\beta)|\cdot l(h) \\ +\sup_\beta |h(\beta)|\cdot l(u) &\leq l(d_n(\alpha+1,\cdot,\gamma)+1-\gamma \\ \leq (n-1)\cdot(1-\gamma) &+ 1-\gamma = n\cdot(1-\gamma). \end{aligned}$$

This proves (I_{n+1}).

(g) This follows easily by induction on n, using (3.6) and the fact that the maximum and also the sum of two decreasing and convex functions is decreasing and convex and that antitonicity and convexity in γ of a function $(\gamma, z) \mapsto w(\gamma, z)$, which is integrable w.r. to $Q'(\alpha, dz)$, is preserved under integration by $Q'(\alpha, dz)$.

(h) We use induction on $n \geq 1$ for the assertion (I_n) that $l(d_n(\alpha,\beta,\cdot)) \leq \sigma_n(\beta)$. Obviously (I_1) is true. Assume (I_n) for some $n \geq 1$. Similar as in the proof of (f) we obtain (I_{n+1}) since for $\gamma, \gamma' \in \mathbb{R}_+$

$$\begin{aligned} &|d_{n+1}(\alpha,\beta,\gamma) - d_{n+1}(\alpha,\beta,\gamma')| \leq \\ &\quad \int Q'(\alpha,dz)\,\beta\cdot|d_n(\alpha+1,\beta,\gamma) - d_n(\alpha+1,\beta,\gamma')| + |\gamma-\gamma'| \\ &\quad \leq (\beta\cdot\sigma_n(\beta)+1)\cdot|\gamma-\gamma'| = \sigma_{n+1}(\beta)\cdot|\gamma-\gamma'|. \end{aligned}$$ □

In the next result we use $[\beta < 1] := \{(\alpha,\beta,\gamma) \in X : \beta < 1\}$.

Proposition 3.5. (Properties of the limit function $d(\cdot)$ on $\mathbb{R}^+ \times (0,1] \times [0,1/2]$) *There holds:*

(a) *There exists* $d(x) := \lim_{n\to\infty} d_n(x), x \in X$, *and* $0 \leq 1/2 - \gamma \leq d(x) \leq 1-\gamma$.

(b) $d(x)$ *is increasing in* α, *hence there exists* $u(\beta,\gamma) := \lim_{\alpha\to\infty} d(x) = \sup_{\alpha\in\mathbb{R}^+} d(x)$.

(c) $\sup_{(\alpha,\gamma)\in\mathbb{R}^+\times[0,1/2]} |d_n(x) - d(x)| \leq \beta^n/2$ *for* $n \geq 1$. *Thus* $d_n(x)$ *converges for* $n \to \infty$ *uniformly and geometrically to* $d(x)$ *on* $[\beta < 1]$, *and there* $d(x)$ *is continuous in* (α,γ).

(d) $u(1,0) = 1$ *and* $u(\beta,\gamma) = \left(1 - \sqrt{1-\beta^2\cdot(1-2\gamma)}\right)/\beta^2 < 1-\gamma$ *if* $(\beta,\gamma) \neq (1,0)$.

(e) $d(\alpha,1,0) = 1$, $\alpha \in \mathbb{R}^+$.

(f) $d(x)$ *is increasing and convex in* β.

(g) $d(x)$ *is decreasing and convex in* γ.

(h) $d(x)$ *is Lipschitz continuous in* γ *on* $[\beta < 1]$ *and there holds* $l(d(\alpha,\beta,\cdot)) \leq 1/(1-\beta)$.

Proof. (a), (b), (f) and (g) follow from parts (a) $\wedge$ (b), (c), (e) and (g) of 3.4, respectively, using elementary facts about the preservation of properties of functions under the limit $n \to \infty$.

(c) If $n \to \infty$ then $v_n(z,\alpha,\gamma) := \max[\beta \cdot d_n(\alpha+1,\gamma), g(\alpha+1)\cdot(z \wedge 1)] \geq 0$ converges by (a) and 3.4(a) increasingly towards $v(z,\alpha,\gamma) := \max[\beta \cdot d(\alpha+1,\gamma), g(\alpha+1)\cdot(z \wedge 1)]$. Now (3.6) and the theorem on monotone convergence of integrals show that $d(\alpha,\gamma) = \int Q'(\alpha,dz)\, v(z,\alpha,\gamma) - \gamma$ for all (α,γ). Next, using $|\max(x_1,x_2) - \max(y_1,y_2)| \leq \max[|x_1-y_1|, |x_2-y_2|]$ for $x_1,x_2,y_1,y_2 \in \mathbb{R}$, and that both d_n and d have the lower bound $1/2-\gamma$ and the upper bound $1-\gamma$, we get from (3.6)

$$\begin{aligned} |d(\alpha,\gamma) - d_{n+1}(\alpha,\gamma)| &= \\ &\left| \int Q'(\alpha,dz)\,[v(z,\alpha,\gamma) - v_n(z,\alpha,\gamma)] \right| \\ &\leq \beta \cdot [d(\alpha+1,\gamma) - d_n(\alpha+1,\gamma)] \\ &\leq \ldots \leq \beta^n \cdot [d(\alpha+n,\gamma) - d_1(\alpha+n,\gamma)] \leq \beta^n/2. \end{aligned}$$

This proves the first assertion, from which the second one follows since the uniform limit of a sequence of continuous functions on $\mathbb{R}^+$ is continuous.

(d1) Fix (β,γ). Since d_n is increasing in n by 3.4(a), the same holds for u_n. Therefore $u = \sup_\alpha \sup_n d_n(\alpha) = \sup_n \sup_\alpha d_n(\alpha) = \sup_n u_n = \lim_n u_n$. Now (3.10) shows that u satisfies $2u = (\beta u)^2 + 1 - 2\gamma$. This quadratic equation yields

$$u = (1 - \sqrt{1-\beta^2(1-2\gamma)})/\beta^2, \tag{3.13}$$

since the other solution $\eta := (1+\sqrt{1-\beta^2(1-2\gamma)})/\beta^2$ satisfies $\beta\eta > 1$ which contradicts $\beta u \leq 1$. Now (3.13) verifies the value of $u(x)$ for all $x \in X$. Finally $u < 1-\gamma$ in case $(\beta,\gamma) \neq (1,0)$ and $u = 1$ else, are easily proven.

(d2) We now add the proof for the assertion made in 3.4(d) that $e_{n+1} = 1/2-\gamma$ if $\beta \cdot (5-8\gamma) \leq 4$. Firstly, $u(\beta,\gamma) = \left(1-\sqrt{1-\beta^2\cdot(1-2\gamma)}\right)/\beta^2$ obviously also holds for $\beta = 1$ and $\gamma = 0$. Now $n_0 = \infty$ iff $\beta \cdot d_n(1) \leq 1/2$, $n \geq 1$, iff $\beta \cdot d(1) \leq 1/2$ if $\beta \cdot u(\beta,\gamma) \leq 1/2$, which holds iff $\beta\cdot(5-8\gamma) \leq 4$.

(e) This follows from (d) for $\beta = 1$ and $\gamma = 0$ and from the fact that $\delta(\alpha) := d(\alpha,1,0)$ is a constant. In fact, from (3.6) and 3.4(c) we get

$$d_{n+1}(\alpha) \geq \int Q'(\alpha,dz)\, d_n(\alpha+1) = d_n(\alpha+1) \geq d_n(\alpha).$$

Now (a) implies $\delta(\alpha) = \delta(\alpha+1)$ for all α, which together with isotonicity of $\delta(\cdot)$, ensured by (b), yields the assertion.

(h) It is easily seen that for each sequence of functions ψ_n on $\mathbb{R}_+$, converging to some function ψ, there holds $l(\psi) \leq \liminf_{n\to\infty} l(\psi_n)$. Now the assertion follows from (a) and from 3.4(h) with $\psi_n(\gamma) := d_n(\alpha, \beta, \gamma)$ and $\psi(\gamma) := d(\alpha, \beta, \gamma)$. □

From 3.3(c) and (d) we now get the solution to the original problem of MacKinnon, generalized by allowing any power p.d. $Pow(\alpha, b)$ as prior.

Corollary 3.6. (Solution of MacKinnon's problem for the case $N = 3, \beta = 1, \gamma = 0$ and arbitrary prior $Pow(\alpha, b)$, $(\alpha, b) \in I$) *There holds:*

(*a*) *The maximal expected undiscounted 3–stage reward has the following value:*

(*a*1) *If* $\alpha > \alpha_0 := (\sqrt{13} - 2)/3 \approx 0.5352$ *then*

$$d_3(\alpha) = \frac{\alpha \cdot (\alpha + 2)}{128(\alpha + 1)^2} \cdot \left(5 - (\alpha + 2)^{-2}\right)^2 + \frac{1}{2}. \tag{3.14}$$

In particular, for the uniform prior $Pow(1, 1)$ *one gets* $d_3(1) = 121/864 + 1/2 \approx 0.6400..$

(*a*2) *If* $\alpha \leq \alpha_0$ *then*

$$d_3(\alpha) = \left(5 - (\alpha + 2)^{-2}\right)/8. \tag{3.15}$$

(*b*) *The optimal stopping time is as follows:*

(*b*1) *Assume* $\alpha > \alpha_0$. *Then accept* z_1 *iff* $z_1 > [5(\alpha + 2)^2 - 1]/[8b \cdot (\alpha + 1) \cdot (\alpha + 2)]$; *otherwise accept* z_2 *iff* $z_2 > (\alpha + 3)(2b(\alpha + 2))^{-1}$; *otherwise accept* z_3.

In particular, for the uniform prior one gets: accept z_1 *iff* $z_1 > 11/12$; *otherwise accept* z_2 *iff* $z_2 > 2/3$; *otherwise accept* z_3.

(*b*2) *Assume* $\alpha \leq \alpha_0$. *Then never accept* z_1; *accept* z_2 *iff* $z_2 > (\alpha + 3)(2(\alpha + 2))^{-1} \cdot \max(1/b, z_1)$; *otherwise accept* z_3.

Proof. Firstly note that 3.3(c) for $n := 1$ implies, observing that $d_1(\alpha+1) = 1/2 < g(\alpha + 1)$,

$$d_2(\alpha) = \frac{\alpha \cdot (\alpha + 2)}{8(\alpha + 1)^2} + \frac{1}{2} = \frac{1}{8} \cdot (5 - (\alpha + 1)^{-2}), \quad \alpha \in \mathbb{R}^+.$$

(a) The assertion follows by simple computations from 3.3(c) with $n := 2$, observing that

$$d_2(\alpha + 1) < g(\alpha + 1) \text{ iff } \alpha > \alpha_0. \tag{3.16}$$

(b) From 3.3(d) with $N = 3$ we know that there holds, using $\min \emptyset := 3$,

$$\tau_3^*(\alpha, b, z_1, z_2) = \min\{1 \le t \le 2 : d_{3-t}(\alpha + t) < g(\alpha + t) \cdot z_t \cdot M_t(b, z^t)\}. \tag{3.17}$$

(bi) Fix $\alpha, z_1 \in \mathbb{R}^+$. It follows from (3.17) and (3.16) that one should accept z_1 iff $[z_1 > (\alpha+2)(b \cdot (\alpha+1))^{-1} \cdot d_2(\alpha+1)] \wedge [\alpha > \alpha_0]$. This shows for both cases $\alpha > \alpha_0$ and $\alpha \le \alpha_0$ when to accept z_1.

(bii) Fix $\alpha, z_1, z_2 \in \mathbb{R}^+$, and assume that z_1 has been rejected. It follows from (3.17) for $t := 2$ that one should accept z_2 iff $\min(bz_2, z_2/z_1) > H(\alpha) := (\alpha+3) \cdot (2 \cdot (\alpha+2))^{-1}$, where $H(\alpha) < 1$.

Case 1: $\alpha > \alpha_0$. Then $bz_1 \le (\alpha+2)(\alpha+1)^{-1} \cdot d_2(\alpha+1)$, hence $bz_1 \le 1$ by (3.16), hence $z_2/z_1 > bz_2$. Thus one should accept z_2 iff $z_2 > H(\alpha)/b$.

Case 2: $\alpha \le \alpha_0$. Then one should accept z_2 iff $\min(bz_2, z_2/z_1) > H(\alpha)$ iff $z_2 > H(\alpha) \cdot \max(1/b, z_1)$.

Now the proof of (b) is complete. □

Appendix. *Proof of Lemma 3.2.* (a) Fix α and put $v(z) := \max[c(\alpha), g(\alpha+1) \cdot (z \wedge 1)]$. Then we get from (3.3) with $b := 1$

$$E(\alpha) = \int Q'(\alpha, dz)\, v(z) = g(\alpha) \cdot \left[\int_0^1 v(z)\, dz + \int_1^\infty v(z)/z^{\alpha+1}\, dz\right].$$

Case 1: $c(\alpha) \le g(\alpha+1)$, which is the first case in (3.5). Then $\rho := c(\alpha)/g(\alpha+1) \le 1$, and simple manipulations yield

$$\begin{aligned}\int_0^1 v(z)\, dz &= \int_0^\rho c(\alpha)\, dz + \int_\rho^1 g(\alpha+1) \cdot z\, dz \\ &= \left[c(\alpha)^2/g(\alpha+1) + g(\alpha+1)\right]/2.\end{aligned}$$

Moreover, since $g(\alpha+1) \cdot z \ge g(\alpha+1) \ge c(\alpha)$ if $z \ge 1$, we have $\int_1^\infty v(z)dz/z^{\alpha+1} = g(\alpha+1) \cdot \int_1^\infty dz/z^{\alpha+1} = g(\alpha+1)/\alpha$. Now (3.5) follows in case 1.

Case 2: $c(\alpha) \ge g(\alpha+1)$, which is the second case in (3.5). Then $c(\alpha) \ge g(\alpha+1) \ge g(\alpha+1) \cdot (z \wedge 1)$, hence

$$\begin{aligned}\int_0^1 v(z)\, dz &+ \int_1^\infty v(z)/z^{\alpha+1} dz = \int_0^1 c(\alpha)\, dz \\ &+ \int_1^\infty c(\alpha)/z^{\alpha+1} dz = c(\alpha) \cdot (\alpha+1)/\alpha.\end{aligned}$$

Now (3.5) follows in case 2.

(b) Fix $0 < \alpha < \overline{\alpha}$. Note that $g(\alpha) = 1 - 1/(\alpha+1)$ is increasing in α on $\mathbb{R}^+$. In order to show that $E(\alpha) \leq E(\overline{\alpha})$ we apply (3.5) and consider four cases, as follows.

Case 1: $c(\alpha) < g(\alpha+1)$ and $c(\overline{\alpha}) < g(\overline{\alpha}+1)$. Then

$$\begin{aligned} 2 \cdot E(\alpha) &= (1-(\alpha+1)^{-2}) \cdot c(\alpha)^2 + 1 \\ &\leq (1-(\overline{\alpha}+1)^{-2}) \cdot c(\overline{\alpha})^2 + 1 = 2 \cdot E(\overline{\alpha}). \end{aligned}$$

Case 2: $c(\alpha) < g(\alpha+1)$ and $c(\overline{\alpha}) \geq g(\overline{\alpha}+1)$. Then, using $0 \leq c(\alpha) \leq g(\alpha+1) \leq g(\overline{\alpha}+1) \leq c(\overline{\alpha})$, we get $c(\alpha)^2 \leq g(\alpha+1)^2$, hence

$$\begin{aligned} 2 \cdot E(\alpha) &= (1-(\alpha+1)^{-2}) \cdot c(\alpha)^2 + 1 \\ &\leq (1-(\overline{\alpha}+1)^{-2}) \cdot g(\overline{\alpha}+1)^2 + 1 = g(\overline{\alpha}+1)^2 \\ &- (\overline{\alpha}+2)^{-2} + 1 = (g(\overline{\alpha}+1)-1)^2 + 2 \cdot g(\overline{\alpha}+1) \\ &- (\overline{\alpha}+2)^{-2} \leq [(\overline{\alpha}+1)(\overline{\alpha}+2)^{-1} - 1]^2 + 2 \cdot c(\overline{\alpha}) \\ &- (\overline{\alpha}+2)^{-2} = 2 \cdot E(\overline{\alpha}). \end{aligned}$$

Case 3: $c(\alpha) \geq g(\alpha+1)$ and $c(\overline{\alpha}) < g(\overline{\alpha}+1)$. Then, using $0 \leq c(\alpha) \leq c(\overline{\alpha}) \leq g(\overline{\alpha}+1) < 1$, we get $c(\alpha)^2 \leq g(\overline{\alpha}+1)^2$, hence

$$\begin{aligned} 2 \cdot E(\overline{\alpha}) &= (1-(\overline{\alpha}+1)^{-2}) \cdot c(\overline{\alpha})^2 + 1 \\ &\geq (1-(\overline{\alpha}+1)^{-2}) \cdot c(\alpha)^2 + 1 - 2c(\alpha) + 2c(\alpha) \\ &= (1-c(\alpha))^2 - c(\alpha)^2/(\overline{\alpha}+1)^2 + 2 \cdot E(\alpha) \\ &\geq (1-g(\overline{\alpha}+1))^2 - g(\overline{\alpha}+1)^2/(\overline{\alpha}+1)^2 + 2 \cdot E(\alpha) \\ &= 2 \cdot E(\alpha). \end{aligned}$$

Case 4: $c(\alpha) \geq g(\alpha+1)$ and $c(\overline{\alpha}) \geq g(\overline{\alpha}+1)$. Then $E(\alpha) = c(\alpha) \leq c(\overline{\alpha}) = E(\overline{\alpha})$.

(c) Put $H(\alpha) := g(\alpha+1)$, $\alpha \in \mathbb{R}^+$. Consider on $\mathbb{R}^+$ the continuous functions $e(\alpha) := (1-(\alpha+1)^{-2})/2 \cdot c(\alpha)^2 + 1/2$ and $c(\cdot)$. Then $E = e$ on $[c < H] := \{\alpha \in \mathbb{R}^+ : c(\alpha) < H(\alpha)\}$ and $E = c$ on $[c \geq H]$. From (a) we know that $e = c$ on $[c = H]$. Since c and H are continuous, the sets $[c < H]$ and $[c > H]$ are open.

Select $\alpha_0 \in \mathbb{R}^+$. We show that E is continuous in α_0. This is obvious if α_0 belongs to either of the two open sets $[c < H]$ or $[c > H]$. Thus let $\alpha_0 \in [c = H]$ and select a sequence $(\alpha_n) \subset \mathbb{R}^+ = [c < H] \cup [c \geq H]$ which converges to α_0 for $n \to \infty$.

Case 1: $\alpha_n \in [c < H]$ for n large enough. Then $E(\alpha_n) = e(\alpha_n) \to e(\alpha_0) = c(\alpha_0) = E(\alpha_0)$ by continuity of e.

Case 2: $\alpha_n \in [c \geq H]$ for n large enough. Then $E(\alpha_n) = c(\alpha_n) \to c(\alpha_0) = E(\alpha_0)$ by continuity of c.

Case 3: Infinitely many α_n's, denoted by α_n^-, belong to $[c < H]$ and infinitely many α_n's, denoted by α_n^+, belong to $[c \geq H]$. Then $\lim_n E(\alpha_n) = E(\alpha_0)$ since

$$\begin{aligned} \lim E(\alpha_n^+) &= \lim c(\alpha_n^+) = c(\alpha_0) = E(\alpha_0) = e(\alpha_0) \\ &= \lim e(\alpha_n^-) = \lim E(\alpha_n^-). \end{aligned}$$

□

References

[1] DeGroot MH (1970) Optimal Statistical Decisions. McGraw-Hill, New York

[2] Hinderer K, Stieglitz M (2005) Solution to Problem 11051 in the Amer. Math. Monthly. To be published

[3] MacKinnon N (2003) Problem 11051. Amer. Math. Monthly 101:957

[4] Rieder U (1975) Bayesian dynamic programming. Adv. Appl. Prob. 7:330–348

[5] Rieder U (1988) Bayessche Kontrollmodelle. Vorlesungsausarbeitung, Universität Ulm

[6] Roberts AW, Varberg DE (1973) Convex Functions. Academic Press, New York

A Remark on the Formulation of Dual Programs Based on Generalized Convexity

Diethard Pallaschke

Institut für Statistik und Mathematische Wirtschaftstheorie
Universität Karlsruhe

Dedicated to Klaus Neumann

Abstract

In this paper we present a general concept for the formulation of the dual program which is based on generalized convexity. This is done in a purely algebraic way where no topological assumptions are made. Moreover all proofs are presented in an extreme simple way. A complete presentation of this subject can be found in the book of D. Pallaschke and S. Rolewicz [14].

Key words: Generalized Convexity, Value Function, Lagrange Duality

AMS(MOS) Subject Classification: 52A07, 26A27, 90C30.

1 Introduction

In this paper, we will follow the presentation of S. Dolecki, S. Kurcyusz and S. Rolewicz (see [3],[4],[5],[6],[7],[10],[11]) on abstract convexity and will study a generalized duality theory for optimization problems. All material of this paper is taken from the book of D. Pallaschke and S. Rolewicz [14] and the survey paper [15], in particular the sections 1 - 4 overlap with the paper [15]. For further references we refer to the books of A. Rubinov [16] and I. Singer [17].

We begin with the formulation of an optimization problem in a general context. Therefore let X, Y be nonempty sets denoted as the *space of domain* and the *space of parameters* and let

$$\Gamma : Y \longrightarrow \mathbf{2}^X$$

be a set-valued mapping. Let $\Gamma^{-1}(x) = \{y \in Y \mid x \in \Gamma(y)\}$ be the inverse set-valued mapping.

Using this notation, we shall write

$$(P) \qquad \begin{array}{l} \min f(x) \\ \text{under} \\ x \in \Gamma(y_0) \ , \ y_0 \in Y. \end{array}$$

for a minimization problem. If the optimization problem is given as usually by

$$(P') \qquad \begin{array}{l} \min f(x) \\ \text{under} \\ g_i(x) \ \leq \ \alpha_i \quad i \in \{1, ..., k\}. \end{array}$$

where $f, g_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ are continuous functions, we take for $X = \mathbb{R}^n$, $Y = \mathbb{R}^k$ and

$$\Gamma : Y \longrightarrow \mathbf{2}^X \quad \text{with} \quad (\alpha_1, ..., \alpha_k) \mapsto \{x \in U \mid g_i(x) \leq \alpha_i, \ i \in \{1, ..., k\} \ \}.$$

Throughout this paper we will use the symbol X if we deal in the space of domain and the symbol Y if we consider the space of parameters.

Denote by $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ the extended reals. Then the *value function* of the above minimization problem is given by

$$f\Gamma : Y \longrightarrow \bar{\mathbb{R}} \quad \text{with} \quad f\Gamma(y_0) = \inf\{f(x) \mid x \in \Gamma(y_0) \ \}.$$

In order to study local properties of the value function, we will introduce now the notion of a *reference system.*

Denote by $\mathcal{F}(Y)$ the linear space of all real-valued functions defined on Y.

Definition 1.1.
A set $\Phi \subseteq \mathcal{F}(Y)$ is called a reference system *if the following two conditions are satisfied:*

(i) for all $c \in \mathbb{R}$ and all $\phi \in \Phi$ holds $\phi + c \in \Phi$,

(ii) for all $\lambda \geq 0$ and all $\phi \in \Phi$ holds $\lambda\phi \in \Phi$.

Natural examples for reference systems are affine functions, convex functions or *DC*-functions, i.e. the differences of convex functions, defined on a linear space.

2 Generalized Convexity

An essential role in convex analysis is played by the Fenchel conjugation and the operation of convexification. It can be shown that both operations do not depend on the linearity of the domain of definition, so that they can be extended for arbitrary reference systems.

Definition 2.1.
Let $\Phi \subseteq \mathcal{F}(Y)$ be a reference system. For a function $f : Y \longrightarrow \bar{\mathbb{R}}$ the function

$$f^* : \Phi \longrightarrow \bar{\mathbb{R}} \quad \textit{with} \quad f^*(\phi) = \sup_{y \in Y}(\phi(y) - f(y))$$

is said to be the Φ-Fenchel conjugate *of f. The* second Φ-Fenchel conjugate *is defined as*

$$f^{**} : Y \longrightarrow \bar{\mathbb{R}} \quad \textit{with} \quad f^{**}(y) = \sup_{\phi \in \Phi}(\phi(y) - f^*(\phi)).$$

The following properties of the Φ-Fenchel conjugation can be shown.

Proposition 2.2.
Suppose that Y is a nonempty set, $\Phi \subseteq \mathcal{F}(Y)$ is a reference system, $f, g : Y \longrightarrow \bar{\mathbb{R}}$, $c \in \mathbb{R}$ and $\phi \in \Phi$. Then the following holds:

(i) $g \geq f$ *implies* $f^* \geq g^*$;

(ii) $(f^* + c) = f^*(\phi - c) = f^* - c$;

(iii) $f(y) + f^*(\phi) \geq \phi(y)$ *for all* $y \in Y$ *and all* $\phi \in \Phi$ *(Fenchel-Moreau Inequality).*

Proof. (i) From $g \geq f$ it follows $-f \geq -g$. Thus for every $\phi \in \Phi$ we have $\phi - f \geq \phi - g$, which implies $f^* \geq g^*$.

(ii) Clearly,

$$(f + c)^*(\phi) = \sup_{y \in Y}(\phi(y) - f(y) - c) = \sup_{y \in Y}((\phi(y) - c) - f(y)) = f^*(\phi - c).$$

On the other hand, we have

$$(f+c)^*(\phi) = \sup_{y\in Y}(\phi(y) - f(y) - c) = \sup_{y\in Y}(\phi(y) - f(y)) - c = f^*(\phi) - c.$$

(iii) By definition of the Φ-Fenchel conjugate we have $f^*(\phi) = \sup_{y\in Y}(\phi(y) - f(y))$. Hence for all $y \in Y$ and all $\phi \in \Phi$ we have $f^*(\phi) \geq \phi(y) - f(y)$, i.e. $f^*(\phi) + f(y) \geq \phi(y)$. □

For a function $f : Y \longrightarrow \bar{\mathbb{R}}$ we denote by

$$f^o : Y \longrightarrow \bar{\mathbb{R}} \quad \text{with} \quad f^o(y) = \sup_{\substack{\phi\in\Phi \\ \phi\leq f}} \phi(y)$$

the Φ-*convexification of* f.

Theorem 2.3. (Kutateladze and Rubinov (1971),(1972),(1976), Elster and Nehse (1974), Dolecki-Kurcyusz (1978))
Let Y be a nonempty set, and $\Phi \subseteq \mathcal{F}(Y)$ a reference system. Then for every $f : Y \longrightarrow \bar{\mathbb{R}}$

$$f^{**} = f^o = \sup_{\substack{\phi\in\Phi \\ f^*(\phi)=0}} \phi(y).$$

Proof. First we will show that the following formula holds:

$$f^o(y) = \sup_{\substack{\phi\in\Phi \\ f^*(\phi)=0}} \phi(y).$$

Observe that from the definition of f^o and f^* the following formula follows:

$$f^o(y) = \sup_{\substack{\phi\in\Phi \\ \phi\leq f}} \phi(y) = \sup_{\substack{\phi\in\Phi \\ f^*(\phi)\leq 0}} \phi(y).$$

We will now show that for every $a > 0$ and every $\phi \in \Phi$ with $f^*(\phi) = -a$ there exists a $\hat{\phi} \in \Phi$ with $\phi \leq \hat{\phi}$ and $f^*(\hat{\phi}) = 0$. Namely, if $f^*(\phi) = -a$, then we put $\hat{\phi} = \phi + a$ and from Proposition 2.2 (ii) it follows the assertion.

Next, we will show that

$$f^{**}(y) = \sup_{\substack{\phi\in\Phi \\ f^*(\phi)=0}} \phi(y).$$

From the definition of f^{**} it follows that:

$$f^{**}(y) = \sup_{\phi\in\Phi}(\phi(y) - f^*(\phi)) \geq \sup_{\substack{\phi\in\Phi \\ f^*(\phi)=0}} (\phi(y) - f^*(\phi)) = \sup_{\substack{\phi\in\Phi \\ f^*(\phi)=0}} \phi(y).$$

Hence we have

$$f^{**}(y) \geq \sup_{\substack{\phi \in \Phi \\ f^*(\phi)=0}} \phi(y).$$

In order to prove the equality, we will show that

$$\sup_{\phi \in \Phi}(\phi(y) - f^*(\phi)) = \sup_{\substack{\phi \in \Phi \\ f^*(\phi)=0}} (\phi(y) - f^*(\phi)).$$

Therefore let us assume that for some $a \in \mathbb{R}$ and $\phi \in \Phi$ we have $f^*(\phi) = a$. Then put $\hat{\phi} = \phi - a$ and from Proposition 2.2 (ii) it follows that $f^*(\hat{\phi}) = 0$. Since for every $y \in Y$

$$\hat{\phi}(y) - f^*(\hat{\phi}) = (\phi(y) - a) - f^*(\phi - a) = \phi(y) - a - (f^*(\phi) - a) = \phi(y) - f^*(\phi),$$

the required equality is proved. □

3 Augmented Lagrangians

The usual theory of Lagrangian multipliers is a way to transform a constrained optimization problem into an unconstrained one in such a manner that a neglect of a constraint is imposed by a penalty.

In this section we will present a general method of constructing augmented Lagrangian from a given reference system Φ.

Definition 3.1.
Let A be a nonempty subset of a set X. A set of functions

$$\mathcal{P}_A = \{p \mid p : X \longrightarrow \bar{\mathbb{R}}\}$$

is called a penalty system *for $A \subseteq X$ if the following conditions are satisfied:*

(i) for all $\lambda \geq 0$ and all $p \in \mathcal{P}_A$ we have $\lambda \cdot p \in \mathcal{P}_A$;

(ii) for all $x \in A$ and all $p \in \mathcal{P}_A$ we have $p(x) \leq 0$;

(iii) for all $x \in X \setminus A$ there exists a $p \in \mathcal{P}_A$ with $p(x) > 0$;

(iv) for all $x \in A$ there exists a $p \in \mathcal{P}_A$ with $p(x) > -\infty$.

For a penalty system the following two simple propositions hold.

Proposition 3.2.
Let $A \subset X$ be a nonempty set, let $f : X \longrightarrow \mathbb{R}$ and let $\mathcal{P}_A$ be a penalty system for A. Then

$$\inf_{x \in A} f(x) = \inf_{x \in X} [\sup_{p \in \mathcal{P}_A} (f(x) + p(x))].$$

Proof. Let us evaluate the expression $L(x) = \sup_{p \in \mathcal{P}_A} (f(x) + p(x))$ at a point $x_0 \in X$.

If $x_0 \in X \setminus A$ then, by condition (iii) of Definition 3.1, there exists a $p_0 \in \mathcal{P}_A$ with $p_0(x_0) > 0$. From condition (i) of Definition 3.1 follows then, that $L(x_0) = +\infty$.

If $x_0 \in A$ then, by condition (ii) of Definition 3.1, we know that for every $p \in \mathcal{P}_A$ we have $p_0(x_0) \leq 0$. Hence, by condition (i), we conclude that $L(x_0) = f(x_0)$. □

Proposition 3.3.
Let $A \subset X$ be a nonempty set, let $f : X \longrightarrow \mathbb{R}$ and let $\mathcal{P}_A$ be a penalty system for A. Then

$$\sup_{p \in \mathcal{P}_A} \inf_{x \in A} (f(x) + p(x)) \leq \inf_{x \in X} [\sup_{p \in \mathcal{P}_A} (f(x) + p(x))].$$

Proof. Write

$$\underline{M} = \sup_{p \in \mathcal{P}_A} \inf_{x \in A} (f(x) + p(x)) \quad \text{and} \quad \overline{M} = \inf_{x \in X} (\sup_{p \in \mathcal{P}_A} (f(x) + p(x))).$$

In order to show that $\underline{M} \leq \overline{M}$ we show that for every $\varepsilon > 0$ we have $\underline{M} - \varepsilon < \overline{M}$.

By definition of the supremum, for every $\varepsilon > 0$ there exists a $p_0 \in \mathcal{P}_A$ with $\underline{M} - \varepsilon \leq \inf_{x \in X} (f(x) + p_0(x))$. Hence for all $x \in X$ we have

$$\underline{M} - \varepsilon \leq f(x) + p_0(x) \leq \sup_{p \in \mathcal{P}_A} (f(x) + p(x)).$$

Thus

$$\underline{M} - \varepsilon \leq \inf_{x \in X} \sup_{p \in \mathcal{P}_A} (f(x) + p(x)) = \overline{M}.$$ □

Now we will come to the construction of a penalty system from a given reference system. Therefore we introduce the following

Definition 3.4.
Let Y be a nonempty set and let $\Phi \subseteq \mathcal{F}(Y)$ be a reference system. A subset $C \subset Y$ is called Φ-convex *if there exists a nonempty subset $\tilde{\Phi} \subseteq \Phi$ such that for every $\phi \in \tilde{\Phi}$ there is a real number $a_\phi \in \mathbb{R}$ with the property*

$$C = \bigcap_{\phi \in \tilde{\Phi}} \{ y \in Y \mid \phi(y) \leq a_\phi \}.$$

For a given reference system we can construct a penalty system in the following way.

Theorem 3.5. (Dolecki-Kurcyusz (1978))
Let X, Y be a nonempty sets and let $\Phi \subseteq \mathcal{F}(Y)$ be a reference system. Moreover, let

$$\Gamma : Y \longrightarrow \mathbf{2}^X$$

be a set-valued mapping such that for every $x \in X$ the set $\Gamma^{-1}(x)$ is Φ-convex.
Then for every $y_0 \in Y$ the set

$$\mathcal{P}_{\Gamma(y_0)} = \{ p : X \longrightarrow \bar{\mathbb{R}} \mid p(x) = - \sup_{y \in \Gamma^{-1}(x)} \phi(y) + \phi(y_0) , \ \phi \in \Phi \}$$

is a penalty system for $\Gamma(y_0)$.

Proof. We have to verify the conditions of Definition 3.1 for a penalty system.

(i) $\lambda \cdot p \in \mathcal{P}_{\Gamma(y_0)}$ for all $\lambda \geq 0$ and all $p \in \mathcal{P}_{\Gamma(y_0)}$. This follows directly from the definition of p as

$$p(x) = - \sup_{y \in \Gamma^{-1}(x)} \phi(y) + \phi(y_0) , \ \phi \in \Phi$$

and from condition (ii) of Definition 1.1.

(ii) $p(x) \leq 0$ for all $x \in \Gamma(y_0)$ and all $p \in \mathcal{P}_{\Gamma(y_0)}$. To prove this statement, suppose that $x \in \Gamma(y_0)$ is given. Then $y_0 \in \Gamma^{-1}(x)$. Hence for every

$\phi \in \Phi$ we have $\phi(y_0) \leq \sup_{y \in \Gamma^{-1}(x)} \phi(y)$, which means that for every $p \in \mathcal{P}_{\Gamma(y_0)}$ we have

$$p(x) = - \sup_{y \in \Gamma^{-1}(x)} \phi(y) + \phi(y_0) \leq 0.$$

(iii) for all $x_0 \in X \setminus \Gamma(y_0)$ there exists a $\hat{p} \in \mathcal{P}_{\Gamma(y_0)}$ with $\hat{p}(x_0) > 0$. Since $\Gamma^{-1}(x_0)$ is Φ-convex and $y_0 \notin \Gamma^{-1}(x_0)$, there exists a $\hat{\phi} \in \Phi$ with $\sup_{y \in \Gamma^{-1}(x_0)} \hat{\phi}(y) < \hat{\phi}(y_0)$. Hence

$$\hat{p}(x) = - \sup_{y \in \Gamma^{-1}(x)} \hat{\phi}(y) + \hat{\phi}(y_0) > 0.$$

(iv) for all $x_0 \in \Gamma(y_0)$ there exists a $p_0 \in \mathcal{P}_{\Gamma(y_0)}$ with $p_0(x) > -\infty$. This means that from $x_0 \in \Gamma(y_0)$ it follows that $y_0 \in \Gamma^{-1}(x_0)$. Since $\Gamma^{-1}(x_0)$ is Φ-convex, there exists a representation of $\Gamma^{-1}(x_0)$ as

$$\Gamma^{-1}(x_0) = \bigcap_{\phi \in \tilde{\Phi}} \{ y \in Y \mid \phi(y) \leq a_\phi \}$$

for some $\tilde{\Phi} \subseteq \Phi$. Hence there exists a $\phi_0 \in \tilde{\Phi}$ with $\sup_{y \in \Gamma^{-1}(x_0)} \phi_0(y) \leq a_{\phi_0}$. This implies

$$p_0(x_0) = - \sup_{y \in \Gamma^{-1}(x_0)} \phi_0(y) + \phi_0(y_0) \geq -a_{\phi_0} + \phi_0(y_0) > -\infty$$

□

Definition 3.6.
Let Y be a nonempty set and let $\Phi \subseteq \mathcal{F}(Y)$ be a reference system. Then

$$L(x, \phi, y_0) = f(x) - \sup_{y \in \Gamma^{-1}(x)} \phi(y) + \phi(y_0)$$

is called the augmented Lagrangian *for the minimization problem*

$$(P_1) \qquad \begin{array}{l} \min f(x) \\ \text{under} \\ x \in \Gamma(y_0) \ , \ y_0 \in Y, \end{array}$$

with respect to the reference system Φ.

Although this definition of a Lagrangian looks quite unfamiliar, it is a natural generalization of the usual Lagrangian-function for the minimization problem

$$\min f(x)$$

under

$$g_i(x) \leq \alpha_i \quad i \in \{1, ..., k\}$$

if we take for $X = \mathbb{R}^n$ and $Y = \mathbb{R}^k$. In this case a corresponding set-valued mapping is

$$\Gamma : \mathbb{R}^k \longrightarrow \mathbf{2}^{(\mathbb{R}^n)} \text{ with } \Gamma(\alpha_1, ..., \alpha_k) = \{x \in \mathbb{R}^n \mid g_i(x) \leq \alpha_i, \quad i = 1, .., k\}.$$

For the reference system of 'positive affine functions', i.e. for

$$\Phi = \{\phi : \mathbb{R}^m \longrightarrow \mathbb{R} \mid \phi(x) = -\langle \lambda, x \rangle + c \text{ with } \lambda \in \mathbb{R}^k_+, \quad c \in \mathbb{R}\},$$

and $\phi \in \Phi$, $\alpha = (\alpha_1, ..., \alpha_k) \in \mathbb{R}^k$ and for a fixed $\alpha^0 = (\alpha^0_1, ..., \alpha^0_k)$ we have

$$\begin{aligned}
L(x, \phi, \alpha^0) &= f(x) - \sup_{\alpha \in \Gamma^{-1}(x)} \phi(\alpha) + \phi(\alpha^0) \\
&= f(x) - \sup_{\alpha \in \Gamma^{-1}(x)} (-\langle \lambda, \alpha \rangle + c) - \langle \lambda, \alpha^0 \rangle + c \\
&= f(x) - \sup_{\alpha \in \Gamma^{-1}(x)} (-\langle \lambda, \alpha \rangle) - \langle \lambda, \alpha^0 \rangle \\
&= f(x) + \inf_{\alpha \in \Gamma^{-1}(x)} (\langle \lambda, \alpha \rangle) - \langle \lambda, \alpha^0 \rangle \\
&= f(x) + \sum_{i=1}^{k} \lambda_i g_i(x) - \sum_{i=1}^{k} \lambda_i \alpha^0_i \\
&= f(x) + \sum_{i=1}^{k} \lambda_i (g_i(x) - \alpha^0_i),
\end{aligned}$$

since

$$\Gamma^{-1}(x) = \{\alpha \in \mathbb{R}^k \mid g_i(x) \leq \alpha_i, \ i = 1, .., k\}$$

and

$$\inf_{\alpha \in \Gamma^{-1}(x)} (\langle \lambda, \alpha \rangle) = \langle \lambda, (g_1(x), ..., g_k(x)) \rangle.$$

4 Duality

We consider the following situation:

Let X, Y be a nonempty sets and let $\Phi \subseteq \mathcal{F}(Y)$ be a reference system and

$$\Gamma : Y \longrightarrow \mathbf{2}^X$$

be a set-valued mapping. Moreover, let

$$(P) \qquad \begin{array}{l} \min f(x) \\ \text{under} \\ x \in \Gamma(y_0) \ , \ y_0 \in Y, \end{array}$$

be a minimization problem under consideration.

Theorem 4.1. (Dolecki-Kurcyusz (1978))
Let X, Y be a nonempty sets and let $\Phi \subseteq \mathcal{F}(Y)$ be a reference system. Moreover, let

$$\Gamma : Y \longrightarrow \mathbf{2}^X$$

be a set-valued mapping such that for every $x \in X$ the set $\Gamma^{-1}(x)$ is Φ-convex.

Then for the minimization problem

$$(P) \qquad \begin{array}{l} min\ f(x) \\ under \\ x \in \Gamma(y_0) \ , \ y_0 \in Y, \end{array}$$

the following statements hold:

(1) $\inf\limits_{x \in X} L(x, \phi, y_0) = -(f\Gamma)^*(\phi) + \phi(y_0)$

(2) $\sup\limits_{\phi \in \Phi} \inf\limits_{x \in X} L(x, \phi, y_0) = (f\Gamma)^o(y_0)$

(3) $\inf\limits_{x \in X} \sup\limits_{\phi \in \Phi} L(x, \phi, y_0) = (f\Gamma)(y_0)$

Proof. These formulas can be proved as follows:

(1) We have:

$$\begin{aligned}
\inf_{x\in X} L(x,\phi,y_0) &= \inf_{x\in X}(f(x) - \sup_{y\in\Gamma^{-1}(x)} \phi(y) + \phi(y_0)) \\
&= \inf_{x\in X}(f(x) - \sup_{y\in\Gamma^{-1}(x)} -(-(\phi(y))) + \phi(y_0)) \\
&= \inf_{x\in X}(f(x) + \inf_{y\in\Gamma^{-1}(x)}(-\phi(y)) + \phi(y_0)) \\
&= \inf_{x\in X}\inf_{y\in\Gamma^{-1}(x)}(f(x) - \phi(y) + \phi(y_0)) \\
&= \inf_{(x,y)\in\{y\in\Gamma^{-1}(x)\ |\ x\in X\}}(f(x) - \phi(y)) + \phi(y_0) \\
&= \inf_{(x,y)\in\{x\in\Gamma(y)\ |\ y\in Y\}}(f(x) - \phi(y)) + \phi(y_0) \\
&= \inf_{y\in Y}\inf_{x\in\Gamma(y)}(f(x) - \phi(y) + \phi(y_0)) \\
&= \inf_{y\in Y}(\inf_{x\in\Gamma(y)}(f(x) - \phi(y)) + \phi(y_0) \\
&= \inf_{y\in Y}(f\Gamma(y) - \phi(y)) + \phi(y_0) \\
&= -\sup_{y\in Y}(\phi(y)) - f\Gamma(y)) + \phi(y_0) \\
&= -(f\Gamma)^*(\phi) + \phi(y_0)
\end{aligned}$$

(2) Since by formula (1) we have

$$\inf_{x\in X} L(x,\phi,y_0) = -(f\Gamma)^*(\phi) + \phi(y_0),$$

it follows from Theorem 2.3 that

$$\sup_{\phi\in\Phi}\inf_{x\in X} L(x,\phi,y_0) = \sup_{\phi\in\Phi}(\phi(y_0) - (f\Gamma)^*(\phi)) = (f\Gamma)^{**}(y_0) = (f\Gamma)^o(y_0).$$

(3) Since by Theorem 3.5

$$\mathcal{P}_{\Gamma(y_0)} = \{p : X \longrightarrow \bar{\mathbb{R}} \mid p(x) = -\sup_{y\in\Gamma^{-1}(x)} \phi(y) + \phi(y_0)\ ,\ \ \phi\in\Phi\ \},$$

is a penalty system for $\Gamma(y_0)$, by Proposition 3.2 follows that

$$\begin{aligned}
\inf_{x\in X} L(x,\phi,y_0) &= \inf_{x\in X}\sup_{\phi\in\Phi}(f(x) - \sup_{y\in\Gamma^{-1}(x)} \phi(y) + \phi(y_0)) \\
&= \inf_{x\in X}\sup_{p\in\mathcal{P}_{\Gamma(y_0)}}(f(x) + p(x)) \\
&= \inf_{x\in\Gamma(y_0)} f(x) = f\Gamma(y_0).
\end{aligned}$$

□

5 The Dual Program

To

$$\sup_{\phi\in\Phi}\inf_{x\in X} L(x,\phi,y_0) = (f\Gamma)^o(y_0)$$

corresponds the following optimization problem:

$$(f\Gamma)^o(y_0) = \sup_{\substack{\phi\leq(f\Gamma)\\ \phi\in\Phi}} \phi(y_0) = \sup_{\substack{(f\Gamma)^*(\phi)=0\\ \phi\in\Phi}} \phi(y_0) = \sup_{\substack{\inf_{x\in X} L(x,\phi,y_0)=\phi(y_0)\\ \phi\in\Phi}} \phi(y_0),$$

because

$$\inf_{x\in X} L(x,\phi,y_0) = -(f\Gamma)^*(\phi) + \phi(y_0)$$

holds. Hence the dual program to

$$(P) \qquad \begin{array}{l} \min f(x) \\ \text{under} \\ x\in\Gamma(y_0)\ ,\ y_0\in Y, \end{array}$$

is

$$(D) \qquad \begin{array}{l} \max \phi(y_0) \\ \text{under} \\ \inf_{x\in X} L(x,\phi,y_0) = \phi(y_0) \\ \phi\in\Phi \end{array}$$

Let us now consider the case of a linear program.

Let $x, c \in \mathbb{R}^n$ and $b, u \in \mathbb{R}^k$ be given vectors and let A be an (n,k)-matrix. We consider the linear programs in the form:

$$(\mathrm{P}) \qquad \begin{array}{l} \max\langle c,x\rangle \\ \text{under} \\ Ax\leq b,\quad x\geq 0 \end{array}$$

To apply our approach, we rewrite this problem as:

$$(\tilde{\mathrm{P}}) \qquad \begin{array}{l} \min\langle \tilde{c},x\rangle \\ \text{under} \\ \tilde{A}x\leq \tilde{b} \end{array}$$

with $\tilde{c} = -c$, and $\tilde{A} = \begin{pmatrix} A \\ -Id_n \end{pmatrix}$, $\tilde{b} = \begin{pmatrix} b \\ 0 \end{pmatrix}$, where Id_n is the $n \times n$-unit matrix and $0 \in \mathbb{R}^n$.

For the reference system of 'positive affine functions', i.e. for

$$\Phi = \{\phi : \mathbb{R}^{m+n} \longrightarrow \mathbb{R} \mid \phi(x) = -\langle \tilde{\lambda}, x \rangle + \alpha \text{ with } \tilde{\lambda} \in \mathbb{R}_+^{m+n}, \ \alpha \in \mathbb{R} \},$$

we have the following Lagrangian-function:

$$L(x, \phi, \tilde{b}) = \langle \tilde{c}, x \rangle + \langle \tilde{\lambda}, \tilde{A}x - \tilde{b} \rangle,$$

where $\tilde{\lambda} = (\lambda, \rho) \in \mathbb{R}^m \times \mathbb{R}^n$.

Since for fixed $\tilde{\lambda}$ the function:

$$x \mapsto L(x, \phi, \tilde{b}) = \langle \tilde{c}, x \rangle + \langle \tilde{\lambda}, \tilde{A}x - \tilde{b} \rangle$$

is linear, it achieves a finite minimum to satisfy the the condition: $\inf_{x \in X} L(x, \phi, \tilde{b}) = \phi(\tilde{b})$ if and only if gives:

$$\tilde{c} + \tilde{A}^{\mathrm{tr}} \tilde{\lambda} = 0$$

and substituting this condition in the above minimum condition gives

$$\phi(\tilde{b}) = -\langle \tilde{\lambda}, b \rangle + \alpha = \langle \tilde{c}, x \rangle + \langle \tilde{\lambda}, \tilde{A}x - \tilde{b} \rangle$$

$$= \langle \underbrace{-\tilde{A}^{\mathrm{tr}} \tilde{\lambda}}_{\tilde{c}}, x \rangle + \langle \tilde{\lambda}, \tilde{A}x - \tilde{b} \rangle = -\langle \tilde{\lambda}, \tilde{b} \rangle$$

i.e. $\alpha = 0$.

Hence the dual problem is:

$$(\tilde{\mathrm{D}}) \qquad \begin{array}{l} \max -\langle \tilde{b}, \lambda \rangle \\ \text{under} \\ -\tilde{A}^{\mathrm{tr}} \tilde{\lambda} = \tilde{c} \\ \lambda \leq 0, \quad \alpha \geq 0 \end{array}$$

or in a different formulation

$$\begin{array}{c} -\min\langle b, \lambda\rangle \\ \text{under} \\ (-A^{\mathrm{tr}}, Id_n)\begin{pmatrix} \lambda \\ \rho \end{pmatrix} = -c \\ \lambda \geq 0, \end{array}$$

because $\rho \geq 0$ and therefore $-A^{\mathrm{tr}}\lambda + \rho = -c$ gives $A^{\mathrm{tr}}\lambda \geq c$. This leads to the problem:

$$\text{(D)} \qquad \begin{array}{c} \min\langle b, \lambda\rangle \\ \text{under} \\ A^{\mathrm{tr}}\lambda \geq c \\ \lambda \geq 0. \end{array}$$

which is the usual dual of a linear program.

Next we consider the case of a convex quadratic program:

Let $x, c \in \mathbb{R}^n$ and $b, u \in \mathbb{R}^k$ be given vectors and let A be an (n,k)-matrix and Q a symmetric positive semi-definite (n,n)-matrix.

We consider the convex quadratic program in the form:

$$\text{(P)} \qquad \begin{array}{c} \min \tfrac{1}{2}\langle x, Qx\rangle + \langle c, x\rangle \\ \text{under} \\ Ax \geq b. \end{array}$$

To apply our approach, we rewrite this problem as:

$$(\tilde{\text{P}}) \qquad \begin{array}{c} \min \tfrac{1}{2}\langle x, Qx\rangle + \langle c, x\rangle \\ \text{under} \\ -Ax \leq -b. \end{array}$$

For the reference system of 'positive affine functions', i.e. for

$$\Phi = \{\phi : \mathbb{R}^m \longrightarrow \mathbb{R} \mid \phi(x) = -\langle \lambda, x\rangle + \alpha \ \text{ with } \ \lambda \in \mathbb{R}^m_+, \ \ \alpha \in \mathbb{R} \ \},$$

we have the following Lagrangian-function:

$$L(x, \phi, -b) = \min \frac{1}{2}\langle x, Qx\rangle + \langle c, x\rangle + \langle \lambda, b - Ax\rangle.$$

In order to determine

$$\inf_{x \in X} L(x, \phi, -b) = \phi(-b) \qquad (*)$$

let us observe that for fixed λ the function:

$$x \mapsto L(x, \phi, -b) = \frac{1}{2}\langle x, Qx\rangle + \langle c, x\rangle + \langle \lambda, b - Ax\rangle$$

achieves its minimum at those $x \in \mathbb{R}^n$ which satisfy the the condition:

$$Qx + c - A^{\mathrm{tr}}\lambda = 0 \quad \text{and} \quad \lambda \geq 0.$$

Substituting this condition in the above minimum condition (*) gives

$$-\frac{1}{2}\langle x, Qx\rangle + \langle \lambda, b\rangle = \langle \lambda, b\rangle + \alpha,$$

which implies that $\alpha = -\frac{1}{2}\langle x, Qx\rangle$.

Hence the dual problem is:

$$(\tilde{\mathrm{D}}) \qquad \begin{array}{c} \max \left(\langle \lambda, b\rangle - \frac{1}{2}\langle x, Qx\rangle\right) \\ \text{under} \\ Qx + c - A^{\mathrm{tr}}\lambda = 0 \\ \lambda \geq 0. \end{array}$$

or in a different formulation

$$\begin{array}{c} \min_{x \in \mathbb{R}^n} \max_{\lambda \in \mathbb{R}^m} \left(\langle \lambda, b\rangle - \frac{1}{2}\langle x, Qx\rangle\right) \\ \text{under} \\ Qx + c = A^{\mathrm{tr}}\lambda \\ \lambda \geq 0, \end{array}$$

because $\langle x, Qx\rangle \geq 0$, for all $x \in \mathbb{R}^n$. This is the usual dual program for a convex quadratic program (see [9], Section 3.3.2.3). It follows from [8], Theorem 5.3 that the duality gap is zero (see also [13], Theorem 4.5.5).

References

[1] Balder EJ (1977) An extension of duality – stability relations to nonconvex optimization problems. SIAM Jour. Contr. Optim. 15:329–343

[2] Clarke FH (1989) Methods of dynamic and nonsmooth optimization. SIAM Regional Conference Series in Applied Mathematics, Capital City Press, Montpellier, Vermont

[3] Dolecki S (1978) Semicontinuity in constrained optimization I. Metric spaces. Control and Cybernetics 7(2):5–16

[4] Dolecki S (1978b) Semicontinuity in constrained optimization Ib. Normed spaces. Control and Cybernetics 7(3):17–25

[5] Dolecki S (1978c) Semicontinuity in constrained optimization II. Control and Cybernetics 7(4):51–68

[6] Dolecki S, Kurcyusz S (1978) On Φ-convexity in extremal problems. SIAM Jour. Control and Optim. 16:277–300

[7] Dolecki S, Rolewicz S (1979b) Exact penalties for local minima. SIAM Jour. Contr. Optim. 17:596–606

[8] Elster KH, Nehse R (1974) Zur Theorie der Polarfunktionale. Math. Operationsforsch. und Stat. Ser. Optimization 5:3–21

[9] Gill PhE, Murray W, Wright MH (1981) Practical Optimization. Academic Press, London, New York, Sidney

[10] Kurcyusz S (1976) Some remarks on generalized Lagrangians. Proc. 7-th IFIP Conference on Optimization Technique, Nice, September 1975, Springer-Verlag

[11] Kurcyusz S (1976b) On existence and nonexistence of Lagrange multipliers in Banach spaces. Jour. Optim. Theory and Appl. 20:81–110

[12] Kutateladze SS, Rubinov AM (1971) Some classes of H-convex functions and sets. Soviet Math. Dokl. 12:665–668

[13] Neumann K, Morlock M (2002) Operations research, 2nd Edition (in German). Carl Hanser Verlag, München

[14] Pallaschke D, Rolewicz S (1997) Foundations of Mathematical Optimization. Mathematics and its Applications. Kluwer Acad. Publ., Dordrecht

[15] Pallaschke D, Rolewicz S (1998) Penalty and augmented Lagrangian in general optimization problems. Charlemagne and his heritage. 1200 years of civilization and science in Europe, Vol. 2 (Aachen, 1995), Brepols, Turnhout, pp. 423–437

[16] Rubinov A (2000) Abstract convexity and global optimization. Nonconvex Optimization and its Applications, 44. Kluwer Academic Publishers, Dordrecht

[17] Singer I (1997) Abstract convex analysis. With a foreword by A. M. Rubinov, Canadian Mathematical Society Series of Monographs and Advanced Texts. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York

Part II
Operations Research in Production and Service Management

An Application of Markov Decision Processes to the Seat Inventory Control Problem

Christiane Barz and Karl-Heinz Waldmann

Institut für Wirtschaftstheorie und Operations Research
Universität Karlsruhe (TH)

1 Introduction

Airlines typically divide a pool of identical seats into several booking classes that represent e.g. different discount levels with differentiated sale conditions and restrictions. Assuming perfect market segmentation, mixing discount and higher-fare passengers in the same aircraft compartment offers the airline the potential of gaining revenue from seats that would otherwise fly empty. If too many seats are sold at a discount price, however, the airline company would loose full-fare passengers. If too many seats are protected for higher-fare demand, the flight would depart with vacant seats. Seat inventory control deals with the optimal allocation of capacity to these different classes of demand, forming a substantial part of a revenue management system.

In the early literature on seat inventory control, many models assume that demand for each booking class arrives during a single contiguous time segment. In this case the booking period can be divided into time-periods for which all booking requests belong to the same fare class. Only total demands of the different booking classes are observed. The distributions of the coming booking classes' demands are known and independent of each other (for a exception with only two fare classes see Brumelle et al. (1990)). Many models additionally assume that customer requests for tickets arrive in increasing fare order, e.g. the class willing to pay the fare $\hat{r}_k$ before $\hat{r}_{k-1}$ with $\hat{r}_k < \hat{r}_{k-1} < \ldots < \hat{r}_1$. Static models then answer the question of how many requests of a certain booking class should be accepted.

The static revenue management model was first introduced by Littlewood (1972) for two fare classes and extended heuristically by Belobaba (1987 and 1989). Curry (1990), Wollmer (1992), Brumelle and McGill (1993) and Robinson (1995) provided the exact solution to the case of k fare classes. Lautenbacher and Stidham (1999) stressed the structure of the underlying Markov decision process of that model and show structural similarities to the dynamic model.

Dynamic models answer the question whether or not to accept a particular reservation request at the time of arrival. They relax the assumption that the demand for different fare classes arrives in a certain predetermined order. Instead, they allow for the possibility of interspersed arrivals of several classes. The demand for each fare class is modelled as a time-dependent (Markov) process, where the interarrival times lengthen or shorten as the scheduled departure time approaches. Lee and Hersh (1993) and Lautenbacher and Stidham (1999) provide a solution to the discrete time case using a Markov decision process formulation. Liang (1999) reformulates and solves this model in continuous time. Kleywegt and Papastavrou (1998) as well as van Slyke and Young (2000) demonstrate that the problem can also be formulated as a (stochastic) knapsack problem. Subramanian et al. (1999) extend the underlying Markov decision process to incorporate cancellations, no-shows, and overbooking.

For these basic static or dynamic single leg seat inventory control problems without cancellations and no-shows (or under the assumptions of Subramanian et al. (1999)) it can be shown that the optimal policy can be stated in terms of booking limits.

Booking limits are controls that limit the number of seats that can be reserved (i.e. accepted requests) for a particular class at a given point in time. Thus, a booking limit b for a particular class at a given point in time indicates that it is optimal to accept further requests up to a predetermined level only.

In the dynamic case without cancellations it can be shown that the booking limits increase for higher revenue customers. Booking limits with this property are sometimes called nested booking limits in the revenue management literature.

Lautenbacher and Stidham (1999) introduce a so-called omnibus model that encompasses both the basic static and dynamic models as special cases. In particular, they prove optimality of a booking-limit policy.

Brumelle and Walczak (2003) allow for semi-Markovian multiple-request arrivals. They allow for no-shows and overbooking but do not consider cancellations. Further, they also allow for some correlations between classes. For the special case that group requests may be accepted partially, Brumelle and Walczak (2003) show optimality of a (generalized) booking-limit policy, where the booking limit may depend on the type of the fare class and the size of the demand. In his dissertation Walczak (2001) presents a more general model that may account for cancellations.

In the present paper we introduce a general model similar to the one of Walczak (2001) that allows for cancellations, no-shows and overbooking and enables us to incorporate additional environmental factors (a concept

similar to the state of the market in the dynamic pricing context of Aviv and Pazgal (2005) and stimulated by the control of queuing systems in a random environment as studied in Helm and Waldmann (1984)). We allow for batch arrivals but since it is known that structural properties fail for total accept/deny decisions, we assume that these batches can be partially accepted.

The paper is organized as follows. In section 2, the decision problem is presented rigorously and reduced to a Markov decision model with an absorbing set. Section 3 contains the (mild and natural) conditions our main result, the optimality of a booking-limit rule, i.e. a decision rule f^* with actions $f^*(s,i) = \min\{d,\ b(i) - s\}$ for $s < b(i)$ and 0, otherwise, is then obtained in section 4, where s denotes the reservations already made, d the actual request for seats, and i an environmental state collecting all information about the customers, the final time to departure, the reward associated with each of the requested seats, and, possibly, some additional factors resulting from an economic or statistical environment. Further, in section 4, some structural results of b are given. Finally, sections 5 and 6 are devoted to various applications and examples demonstrating the efficacy and scope of our results.

Notation. We use $\mathbb{N}_0$ ($\mathbb{N}$) to denote the set of all nonnegative (positive) integers. Given an arbitrary set S endowed with some ordering $\leq_S$, a function $v : M \to \mathbb{R}$ is said to be increasing (decreasing) if $s \leq_S s'$ implies $v(s) \leq v(s')$ ($v(s) \geq v(s')$). Further the notation $\|.\|$ will be used to denote the supremum norm, i.e. $\|v\| := \sup_{s \in S} |v(s)|$ for any $v : S \to \mathbb{R}$. Finally, for all $v : \mathbb{N}_0 \to \mathbb{R}$, we use $\Delta v(s) := v(s+1) - v(s)$, $s \in \mathbb{N}_0$, to denote the increase of v.

2 The Decision Model

We consider a nonstop flight of an airplane with a capacity of C seats that is to depart after a certain time T. There are k ($k \in \mathbb{N}$) booking classes with associated fares of $0 < \hat{r}_k < \hat{r}_{k-1} < \ldots < \hat{r}_1$. At random times customers request a certain number of reservations $d \in D := \{0, 1, \ldots, d_{max}\}$ ($d_{max} \in \mathbb{N}$) for seats of a booking class with fare $r \in R := \{0, \hat{r}_k, \ldots, \hat{r}_1\}$. Thus, the $(n+1)$-th request gives information on the number $d_n \in D$ of tickets (the costumer is interested in) and the reward $r_n \in R$ that is offered for each of the d_n tickets. It is to decide how many of these requested tickets should be actually sold.

2.1 *The Environmental Process and the Arrival of Requests*

Let the times of a request result from the realization of an environmental process (E_n) with finite state space E. Some examples will be given later. In particular, let the $(n+1)$-th request correspond to stage n of the environmental process (E_n). We suppose that d_n and r_n are realizations of random variables D_n and R_n, respectively, with joint distribution

$$\eta_{e_n}(d_n, r_n) := P(D_n = d_n, R_n = r_n \mid E_n = e_n),$$

which may depend on the environmental state e_n. On the other hand, the environmental process is assumed to be correlated with the realized demand d_n and the realized reward r_n in such a way that the distribution of E_{n+1},

$$\kappa_{e_n,d_n,r_n}(e_{n+1}) := P(E_{n+1} = e_{n+1} \mid E_n = e_n, D_n = d_n, R_n = r_n),$$

is allowed to depend on d_n and r_n (in addition to e_n).

Note that the random variables D_n, R_n, and E_n do not depend on the control of the process. Therefore we may summarize all information about D_n, R_n, and E_n in an external process (I_n) with finite state space $I = D \times R \times E$, which can be easily verified to be a Markov chain with transition matrix $P = (p_{ij})$, where, for each $i \in I$ and $j = (d', r', e') \in I$ we have $p_{ij} = \kappa_i(e')\eta_{e'}(d', r')$.

To realize a finite number of requests (almost sure), we suppose that the environmental process runs into an absorbing state, indicating the end of the booking process. In the revenue management setting absorbing states indicate that the next observable event will be flight departure: Having entered an absorbing state neither more requests, nor cancellations occur; no additional costs or rewards may be gained. Some conditions on the behavior of the Markov chain (I_n) will be given later. At the moment we only exploit that it has a (nonempty) absorbing set J_0, say, i.e. for all $i \in J_0$ it holds that $\sum_{j \in J_0} p_{ij} = 1$. Further, we introduce $J := \{i \in I \mid i \notin J_0\}$ to be the complement of J_0 in I and refer to J as the essential state space of the external process.

2.2 *Accepting Requests, Cancellations and No-Shows*

Fix $n \in \mathbb{N}_0$. Let $s_n \in S := \mathbb{N}_0$ denote the number of reservations at stage n. Then, given the $(n+1)$-th request with demand d_n and reward r_n, respectively, a decision has to be made about the number $a_n \in A(d_n) := \{0, \ldots, d_n\}$ of seats to be accepted for reservation, leading to a reward $r_n \cdot a_n$ and an increase of the number of reservations from s_n to $s_n + a_n$.

We also consider no-shows and cancellations of reservations. In particular, the number of reservations is assumed to reduce from $s_n + a_n$ to $s_{n+1} \leq s_n + a_n$ between stages n and $n+1$ with probability $q^{i_n}_{s_n+a_n,s_{n+1}}$, say, which additionally may depend on the external state i_n. Note, that for the arrival process we allowed d_n to be equal to 0, what represents a pure cancelling event without any requests.

To make up for the loss of reservations due to cancellations and no-shows, s_n+a_n is not bounded by the capacity C of the airplane. Thus we allow overbooking, which, however, is qualified by a penalty cost $\Gamma_P(s_n, i_n) \geq 0$ and a terminal cost $\Gamma_T(\xi, \zeta) \geq 0$, depending on the extended state $(\xi, \zeta) \in S \times J_0$ upon departure. By making some reasonable assumptions on these penalty and overbooking costs, booking limits are obtained in an implicit way.

For incorporating these costs into our model we consider at each stage not only the above mentioned rewards for accepting requests reduced by the penalty costs for ignoring the capacity restrictions of the airplane, but also expected costs of overbooking weighted with the probability that there will be no more request-events until departure.

As already mentioned, we assume that a customer who has not cancelled in advance still will not show up at departure with a probability of $p^{ns}(\zeta)$, depending on the actual external state ζ at the time of departure. These so called "no-shows", as modelled in Subramanian et al. (1999) and Brumelle and Walczak (2003), can be considered within the terminal costs Γ_T by setting

$$\Gamma_T(\xi, \zeta) = \mathbb{E}_{\xi, 1-p^{ns}(\zeta)}[g_\zeta(\max\{0,\ X - C\})],$$

where $g_\zeta(x) \geq 0$ are the costs that stem from denying x passengers with reservations to board the plane at external state $\zeta \in J_0$. These costs are typically 0 for $x = 0$ and increasing and convex in x. The random variable X is assumed to have a binomial distribution with parameters ξ and $1 - p^{ns}(\zeta)$.

This modelling approach implies the usual assumptions that cancellation and no-show probabilities are the same for all customers and are independent of the time the reservations on hand were accepted. If these probabilities are mutually independent across customers, the equivalent charging scheme (as used in Subramanian et al. (1999)) can be applied for modelling class dependent cancellation and no-show refunds. For simplicity we will not consider any refunds in the following.

2.3 *The Underlying Markov Decision Process*

Our decision problem can be treated as a Markov decision process (MDP) with countable state space $S \times I$, countable action space $A := \mathbb{N}_0$, finite subsets $A(d)$ of admissible actions in $(s,i) \in S \times I$, $i = (d,r,e)$, constraint set $K := \{((s,i),a) \mid (s,(d,r,e)) \in S \times I, a \in A(d)\}$, transition law $\tilde{q}$ from K into $S \times I$, where $\tilde{q}((s,i),a,(s',j)) := q^i_{s+a,s'} p_{ij}$, reward function $\tilde{r} : K \to \mathbb{R}$, given by

$$\tilde{r}((s,i),a) := r \cdot a - c_i(s+a) \quad \text{for } (s,i) \in S \times J,\ a \in A(d)$$

and $\tilde{r}((s,i),a) := 0$, otherwise, where, for $s \in S$, $i \in J$,

$$c_i(s) := \Gamma_P(s,i) + \sum_{s'=0}^{s} q^i_{s,s'} \sum_{j \in J_0} p_{ij} \Gamma_T(s',j),$$

and, finally, discount factor $\beta = 1$.

Let $\mathfrak{F} := \{f : S \times I \to A \mid f(s,(d,r,e)) \in A(d)\}$ be the set of all decision rules. Each $f \in \mathfrak{F}$ is thought of as a stationary policy specifying action $a = f(s,i)$ to be taken in state (s,i).

It will be shown in the proof of Theorem 1 that the assumptions (A1) and (A2) to be introduced below imply the general assumption (GA) and condition (C) in Schäl (1975). Hence the following expressions will be well defined: $V_f(s,i)$, the expected total reward starting in state (s,i) and following decision rule f, i.e.

$$V_f(s,i) := \mathbb{E}_{f,(s,i)} \left(\sum_{n=0}^{\infty} \tilde{r}((S_n,I_n), f(S_n,I_n)) \mid (S_0,I_0) = (s,i) \right),$$

and $V(s,i)$, the maximal expected total reward starting in state (s,i), i.e.

$$V(s,i) := \max_{f \in \mathfrak{F}} V_f(s,i), \quad (s,i) \in S \times I.$$

A decision rule $f^* \in \mathfrak{F}$ is said to be optimal if $V_{f^*}(s,i) = V(s,i)$ holds for all $(s,i) \in S \times I$. Note that $V_f \equiv 0$, $f \in \mathfrak{F}$, and $V \equiv 0$ on $S \times J_0$.

For all $f \in \mathfrak{F}$, $(s,i) \in S \times J$, $i = (d,r,e)$, all $a \in A(d)$, and all $v : S \times J \to \mathbb{R}$ such that the following expressions are well defined, we set

$$\begin{aligned} Lv(s,i,a) &:= a \cdot r - c_i(s+a) + \sum_{s'=0}^{s+a} q^i_{s+a,s'} \sum_{j \in J} p_{ij} v(s',j) \\ U_f v(s,i) &:= Lv(s,i,f(s,i)) \\ Uv(s,i) &:= \max\{Lv(s,i,a) \mid a \in A(d)\} \end{aligned}$$

Then, for example, if (GA) and (C) in Schäl (1975) are fulfilled (cf. Theorem 4.2 and 5.3 there), V is solution to a functional equation (optimality equation)

$$V = UV \tag{1}$$

and may be obtained by successive approximations (value iteration)

$$V = \lim_{n\to\infty} Uv_n \qquad v_0 \equiv 0 \qquad v_n = Uv_{n-1}, \qquad n \in \mathbb{N}.$$

Moreover, each decision rule $f^* \in \mathfrak{F}$ maximizing the right-hand side of (1), i.e. for which $V = U_{f^*}V$ holds, is optimal.

Our objective will be to establish optimality of a decision rule of a (generalized) booking-limit type. A decision rule $f_b \in \mathfrak{F}$ will be called a booking-limit rule, if there exists a function $b : J \to A$ such that for all $s \in S$ and $i = (d, r, e) \in J$

$$f_b(s,i) = \begin{cases} \min\{d,\ b(i) - s\} & s < b(i) \\ 0 & s \geq b(i), \end{cases}$$

which implies that the requests are accepted up to some booking limit $b(i)$, which may depend on the actual state of the external process.

3 Assumptions

Denote by P_J the substochastic matrix resulting from P by dropping the rows and columns of all external states belonging to J_0. Then the n-th power of P_J multiplied by 1, i.e. $P_J^n 1$, can be thought of as the probability that the external process has not yet entered the absorbing set J_0 at time n starting in J.

(A1) There is some $n \in \mathbb{N}$ such that $\|P_J^n 1\| < 1$.

It is shown in Hinderer and Waldmann (2005) (within the more general setting of a nonlinear operator) that assumption (A1) is equivalent to a large number of other conditions. So, e.g., (A1) is equivalent to postulating that the external process enters the absorbing set J_0 in finite time with probability one or that the entrance time has finite moments of any order.

(A2) For all $i = (d, r, e) \in J$

(i) $c_i(\cdot)$ is increasing and convex,

(ii) $\lim_{s\to\infty} \Delta c_i(s) > r$.

Remember that I is finite and that $c_i(\cdot)$ is a nonnegative function. Then, given (A2), it easily follows that $r \cdot a - c_i(s+a) = \sum_{\nu=0}^{a-1}(r - \Delta c_i(s+\nu)) - c_i(s)$ is bounded from above and that there exist critical numbers

$$z(i) := \min\{s \in S \mid \Delta c_i(s) > r\}, \quad i = (d, r, e) \in J,$$

which will be seen to be upper bounds for a booking limit in state (s, i), i.e. $f^*(s, i) \leq z(i) - s$ for all optimal decision rules.

Together with Lemma 1' in Stidham (1978) we also infer from assumption (A2) that, for $i = (d, r, e) \in J$, the function $\max_{0 \leq a \leq d}\{ra - c_i(s+a)\}$ is decreasing and concave (in s), which, together with the following assumption (A3) forms the basis for obtaining the same result for $V(\cdot, i)$, $i \in J$, by induction.

(A3) For all $i \in J$ and all increasing and convex functions $h : S \to \mathbb{R}$ it holds that

$$s \to \sum_{s'=0}^{s} q_{s,s'}^{i} h(s')$$

is increasing and convex.

Assumption (A1) does not seem to be a problem in the revenue management setting, since we have a planning horizon of finite length. Part (i) of (A2) is satisfied by the common assumption on the penalty costs and terminal costs for overbooking to be increasing and convex in s for every $i \in J$. In the case of no-shows as defined above, this is true if $g_\zeta(\cdot)$ is increasing and convex for each external state ζ at departure (see Subramanian et al. (1999)). Part (ii) is satisfied by the assumption that the costs of overbooking by one additional seat are at least as high as the reward earned by accepting the highest-revenue customer - another very common assumption in the literature.

(A3) holds if the transition matrices $(q_{s,s'}^{i})$, $i \in J$, are stochastically increasing and convex in the sense of Shaked and Shanthikumar (1988). This needs to be checked in concrete applications. E.g. it is known to hold if the reduction of the number of customers can be modelled by a binomial distribution (see Example 4.1 in Shaked and Shanthikumar (1988)).

4 Optimality of a Booking-Limit Rule

We are now in a position to state and prove our main results.

Theorem 1. *Assume (A1) to (A3). Then there exists an optimal booking-limit rule, i.e. an optimal decision rule $f_b^* \in \mathfrak{F}$, with*

$$b(i) \leq z(i), \quad i \in J.$$

Proof. Since I is finite, we have $\gamma(s) := \max_{i \in J} c_i(s) < \infty$, $s \in S$, and $z_{max} := \max_{i \in J} z(i) < \infty$. Now, by letting $f \equiv 0$, it easily follows that

$$V(s,i) \geq V_f(s,i) \geq -\gamma(s) \sum_{n=0}^{\infty} P_J^n 1(i), \quad (s,i) \in S \times J.$$

On the other hand, since $\tilde{r}((s,i),a)$ is bounded from above by $\mu < \infty$, say, we have

$$V(s,i) \leq \mu \sum_{n=0}^{\infty} P_J^n 1(i), \quad (s,i) \in S \times J.$$

Finally, for $n = km + \ell$, using $\|P_J^n 1\| \leq (\|P_J^m 1\|)^k \cdot \|P_J^\ell 1\|$, assumption (GA) and condition (C) in Schäl (1975) are easily seen to hold.

Now, by essentially making use of the same arguments as given in the proof of Theorem 4.1 in Helm and Waldmann (1984) for a controlled queueing system, the optimality of a decision rule of a booking-limit type follows by induction. □

Under the following assumption, it can be shown that the booking limits are independent of the observed demands and that they are increasing in the observed reward (for any environmental state).

(A4) $\kappa_i(.) \equiv \kappa_e(.)$, $c_i(.) \equiv c_e(.)$, and $q^i_{ss'} = q^e_{ss'}$, $s, s' \in S$, depend on $i = (d,r,e)$ only through e.

Theorem 2. *Assume (A1) to (A4). Then $b(i) = b(r,e)$ depends on $i = (d,r,e)$ only through r and e and for $r \leq r'$, $e = e'$ we have $b(r,e) \leq b(r',e)$.*

Proof. Fix $i = (d,r,e)$ and $i' = (d',r',e')$ with $r \leq r'$ and $e = e'$. For $s \in S$ set

$$w_e(s) := c_e(s) - \sum_{s''=0}^{s} q^e_{s,s''} \sum_{e'' \in E} \kappa_e(e'') \sum_{(d'',r'') \in D \times R} \eta_{e''}(d'',r'') V(s'',(d'',r'',e'')).$$

Then $V(s,i) = \max_{0 \leq a \leq d}\{ra - w_e(s+a)\}$. Now, $b(i)$ can be characterized as an action a_0, say, such that both

$$ra - w_e(s+a) \leq r(a+1) - w_e(s+a+1), \quad a < a_0,$$

and

$$ra - w_e(s+a) \geq r(a+1) - w_e(s+a+1), \quad a \geq a_0,$$

hold, from which we infer that $b(i)$ is independent of d.

If $b(r,e) = 0$, then $b(r',e) \geq 0$ trivially holds. Thus let $b(r,e) > 0$. Then, for $s \in S$,

$$\Delta w_e(s+a) \leq r, \quad a < b(r,e),$$

and, since $r \leq r'$ by assumption,

$$\Delta w_e(s+a) \leq r', \quad a < b(r,e),$$

too, which, finally, gives $b(r',e) \geq b(r,e)$. □

5 An Application to Standard Models

Both the basic static and dynamic model of seat inventory control can be easily verified to be special cases of our general approach.

5.1 The Static Model

Recall that the decision process in the static model is as follows: At the time the total demand D_m for fare class m (with reward $\hat{r}_m$ for each seat) is observed, the airline determines the number a_m of requests to accept for reservation in order to maximize the total expected reward. Overbooking is not allowed, cancellations as well as no-shows are not considered.

Let $1,\ldots,k$ be the fare classes under consideration. Then the static model results from our general model by choosing $E = \{0,1,\ldots,k\}$ and letting $\kappa_i(e') = \kappa_e(e') = 1$ for $e' = \max\{0,\ e-1\}$ and 0, otherwise. The reward R_n is given (a.s.) by the reward $\hat{r}_m$ of the actual fare class. Hence $\eta_e(d,r) = P(D_e = d)$, if $r = \hat{r}_e$, and 0, otherwise.

Note that $J_0 = \{(d,r,e) \in I \mid e = 0\}$. Since cancellations are not allowed, we have $q^i_{s,s'} = 1$ for $s' = s$ and 0, otherwise ($s \in S$, $i \in J$). To avoid overbooking, let $\Gamma_P(s,i) = \rho \cdot \max\{0,\ s-C\}$, $(s,i) \in S \times J_0$, where $\rho > 0$ is sufficiently large (e.g. $\rho > \hat{r}_1 \cdot C$) that it is suboptimal to make a decision leading to a total number $s+a > C$ of reservations. $\Gamma_T(\xi,\zeta)$ equals 0 for all $\xi \in S$ and $\zeta \in J_0$.

It easily follows that $\|P^k_J 1\| = 0$. Thus assumption (A1) holds. Since $c_i(s) := \Gamma_P(s,i) = \rho \cdot \max\{0,\ s-C\}$ is increasing and convex and it holds that $\Delta c_i(s) = \rho > r$ for $s > C$, also assumption (A2) is fulfilled. (A3) holds trivially, since $q^i_{s,s'} = 1$ for $s = s'$ by assumption. A similar argument verifies (A4) to hold.

Hence, by applying Theorems 1 and 2, there exists an optimal booking-limit rule f^*_b, which can be rewritten as

$$f_m(s_m,d_m) = \begin{cases} \min\{d_m,\ b_m - s_m\} & s_m < b_m \\ 0 & s_m \geq b_m \end{cases}$$

for every class $m = k, \ldots, 1$.

For a more intuitive interpretation: If, for class m ($m = k, k-1, \ldots, 1$), there is a demand d_m, the airline accepts $a_m = \min\{d_m,\ b_m - s_m\}$ requests for reservation (in addition to the s_m reservations already made). The result is well-known (see, e.g., Wollmer (1992) or chapter 2.2 in Talluri and van Ryzin (2004)).

Most static revenue management models do not consider cancellations or no-shows. Talluri and van Ryzin (2004), chapter 4.4, introduce extensions for incorporating those effects under the assumptions that cancellation and no-show probabilities are the same for all customers, are mutually independent across customers and are independent of the time the reservations on hand were accepted. They assume that there is a certain probability p_m with which each of the $s_m + a_m$ reservations on hand is cancelled before the next class requests arrive. Then they can show again the optimality of a booking-limit rule with limits that depend only on the current booking class m.

Note that still $J_0 = \{(d, r, e) \in I \mid e = 0\}$. But since cancellations are allowed, our probabilities $q^{i_m}_{s_m+a_m, s_{m+1}}$ for a reduction from state $s_m + a_m$ to state s_{m+1} result now from a binomial distribution with parameters $s_m + a_m$ and $(1 - p_m)$. Since no-shows and overbooking are allowed, we model Γ_T, as mentioned above, by setting $\Gamma_T(\xi, \zeta) = \mathbb{E}_{\xi, 1-p^{n\bullet}(\zeta)}[g_\zeta(\max\{0,\ X - C\})$. $g_\zeta(\cdot)$ is assumed to be an increasing and convex function with $g_\zeta(0) = 0$ and $\Delta g_\zeta(x) > \rho$, $x = 1, 2, \ldots$, where $\rho > 0$ is sufficiently large (e.g. $\rho > \hat{r}_1 \cdot C$) that it is suboptimal to make decisions leading to a number $x > C$ of reservations at time of departure for every external state ζ.

To reduce the computational burden and to simplify analysis, an overbooking pad σ is often assumed. This maximum overbooking level results in an additional state constraint $0 \leq s \leq C + \sigma$ and can be enforced by setting $\Gamma_P(s, i) = \rho \cdot \max\{0,\ s - C - \sigma\}$, $(s, i) \in S \times J_0$, where $\rho > 0$ is chosen as above.

The new cost function $c_i(s)$ is increasing and convex and it holds that $\Delta c_i(s) = \rho > r$ for $s > C + \sigma$. So assumption (A2) is fulfilled. Since binomial distributed random variables are stochastically increasing convex and thus (A3) is fulfilled, we again get the optimality of a booking-limit rule immediately from Theorem 1. Since (A4) remains to be fulfilled, we get from 2 that the booking limit only depends on the environment, the current booking class m.

5.2 The Dynamic Model

Dynamic models relax the assumption that the demand for different fare classes arrives in a certain predetermined order, but allow for interspersed

arrivals of requests for several classes. This arrival process is assumed to be Markovian and is defined independently of any seat sale policy. The other assumptions of the static model are retained. We only consider the case without group bookings and state the discrete time case as introduced by Lee and Hersh (1993) and Lautenbacher and Stidham (1999).

Thus within the setting of a dynamic model the decisions to be made can be described as follows: A request for (exactly) one seat is observed at times $t = T, T-1, \ldots, 1$ before departure. (The time periods are chosen so that the probability of more than one request per period is 0.) The airline has then to decide whether or not to accept this request knowing the class $m \in \{1, \ldots, k\}$ of the request and thus the associated reward $\hat{r}_m$. For modelling the event of no customer arrival the artificial class 0 is constructed with $\hat{r}_0 = 0$ (so that $\sum_{m=0}^{k} P(R_t = \hat{r}_m) = 1$ for all t).

The dynamic model results from our general model by choosing $E = \{0, 1, \ldots, T\}$ and letting $\kappa_t(e') = \kappa_e(e') = 1$ for $e' = \max\{0,\ e-1\}$ and 0, otherwise. The demand D_n is equal to 1 (a.s.). Hence $\eta_e(d, r) = P(R_e = r)$, if $d = 1$, and 0, otherwise.

Note that $J_0 = \{(d, r, e) \in I \mid e = 0\}$. Since cancellations are not allowed, we have $q^i_{s,s'} = 1$ for $s' = s$ and 0, otherwise ($s \in S$, $i \in J$). To avoid overbooking, let $\Gamma_P(s, i) = \rho \cdot \max\{0,\ s - C\}$, $(s, i) \in S \times J_0$, where $\rho > 0$ is sufficiently large (e.g. $\rho > \hat{r}_1 \cdot C$) such that it is suboptimal to make a decision leading to a total number $s + a > C$ of reservations. There are no terminal costs.

It easily follows that $\|P_J^T 1\| = 0$. Thus assumption (A1) holds. Since $c_i(s) := \Gamma_P(s, i) = \rho \cdot \max\{0,\ s - C\}$ is increasing and convex and it holds that $\Delta c_i(s) = \rho > r$ for $s > C$, also assumption (A2) is fulfilled. (A3) holds trivially, since $q^i_{s,s'} = 1$ for $s = s'$ by assumption. A similar argument verifies (A4) to hold.

Hence, by applying Theorems 1 and 2, there exists an optimal booking-limit rule f_b^*, which can be rewritten as

$$f_t(s_t, r_t) = \begin{cases} 1 & s_t < b_t(r_t) \\ 0 & s_t \geq b_t(r_t) \end{cases}$$

for every $t = 1, \ldots, T$.

For a more intuitive interpretation: If, at time t ($t = T, T-1, \ldots, 1$), there is a request with reward r_t, the airline accepts the request for reservation, if $s_t < b_t(r_t)$, and reject it, otherwise. (Thus the decision whether or not to accept the request depends on the number s_t of reservations already made, the associated reward r_t, and the final time t to departure.)

Moreover, by again applying Theorem 2, it additionally follows that the booking limits are nested, i.e. $b_t(r) \leq b_t(r')$ for all $r < r'$ and all t.

In this (as well as in the corresponding continuous time) setting it is well-known (see, e.g., Lee and Hersh (1993), Talluri and van Ryzin (2004), chapter 2.5) that the optimal policy is of (time- and class-dependent) nested booking-limit type.

Subramanian et. al. (1999) introduce cancellations, no-shows and overbooking. Again, it is assumed that cancellation and no-show probabilities are the same for all customers, are mutually independent across customers and are independent of the time the reservations on hand were accepted. Further they assume that there is a certain probability p_t with which each of the $s_t + a_t$ reservations on hand is cancelled before the next class request arrives in order to show that the optimal policy is of a booking-limit type.

If we chose the terminal costs as in the static model with cancellations and no-shows, (A2) is fulfilled. Let the probabilities $q^{i_t}_{s_t+a_t,s_{t+1}}$ of a reduction from state $s_t + a_t$ to state s_{t+1} form a binomial distribution with parameters $s_t + a_t$ and $(1 - p_t)$. Then (A3) is fulfilled and we again get the optimality of a booking-limit rule immediately from Theorem 1. Together with Theorem 2 the booking limits are (independent of the demand and) monotone in r, i.e. nested with $b_t(r) \leq b_t(r')$ for all $r < r'$ and all t.

6 Examples of a Random Environment

Still, we have not answered what the environment (E_n) could stand for (in addition to a pure time-parameter). Thus, in this section, our objective will be to demonstrate the great versatility of the environmental process. To avoid technical difficulties and cumbersome notation we should often restrict attention to simple situations dealing with one specific topic only. Clearly, most of these features can also be realized simultaneously.

6.1 Exogenous Effects Based on the Evolution of Markov Chain

Assume (E_n) to be an exogenous Markov chain with state space E and transition matrix P^E, which is independent of the arrival process of the customers.

This Markov chain might represent the overall economic cycle or the change in currency exchange rates. Such factors might influence the booking behavior of customers with respect to all, the number of requests at a certain point in time, the booking classes requested, the cancellation probabilities, and, finally, the terminal costs.

Other uncertain effects like the recovery in tourism in regions that suffered from natural catastrophes or terrorist attacks might also be modelled by such an external Markov chain.

6.2 Seat Inventory Control Under Uncertainty

For new routes or for time changes of flights there is often some uncertainty about the transition matrix (p_{ij}) of the external process (I_n). For example, one of the following objects (or combinations of it) may be partially known only:

(i) the transition matrix P^E of the exogenous Markov chain considered above,

(ii) the distribution of the demands D_n, provided that they are i.i.d. random variables,

(iii) the distribution of the fare classes R_n, provided that they are i.i.d. random variables.

To illustrate the idea of an adaptive control of such a system under uncertainty, let the rewards $R_0, R_1, \ldots$ be i.i.d. random variables with distribution $P(R_n = r) = \psi^{\vartheta}(r)$, $r \in R$, which is known up to some unknown parameter $\vartheta \in \Theta$.

Handling the uncertainty about ϑ from the Bayesian point of view, some probability measure ϱ on (the Borel σ-algebra of) Θ is supposed to be given. Based on the prior information ϱ and the observed rewards $r_0, \ldots, r_{n-1}$, the update (posterior information) ϱ_n then gives the actual information about ϑ at stage n.

Finally, by considering the augmented environmental states (e_n, ϱ_n) (in the reduced decision model under risk) there is a (Bayes-) optimal decision rule, which is of a booking-limit type with limits $b(i_n, \varrho_n)$ additionally depending on ϱ_n.

6.3 General Demand Patterns

The construction of the external process enables us to consider dependencies between the actual demand and the demand observed at earlier stages.

For example, let $e_n = (e_{1,n}, \ldots, e_{k,n})$ denote the vector of the total demands $e_{1,n}, \ldots, e_{k,n}$ of the tickets for the fare classes $1, \ldots, k$ up to stage $n-1$. Then, by updating e_n to

$$e_{n+1} = (e_{1,n} + d_n \cdot \delta_{\hat{r}_1, r_n}, \ldots, e_{k,n} + d_n \cdot \delta_{\hat{r}_k, r_n})$$

(with $\delta_{ij} = 1$ for $i = j$ and 0, otherwise), general demand patterns can be modelled easily.

Clearly, this environment has to be augmented by a pure time-parameter in order to fulfill (A1).

References

[1] Aviv Y, Pazgal A (2005) A partially observed Markov decision process for dynamic pricing. Management Science 49(9):1400–1416

[2] Belobaba PP (1987) Airline yield management. Transportation Science 21(2):63–73

[3] Belobaba PP (1989) Application of a probabilistic decision model to airline seat inventory control. Operations Research 37(2):183–197

[4] Brumelle SL, McGill JI (1993) Airline seat allocation with multiple nested fare classes. Operations Research 41(1):127–137

[5] Brumelle SL, McGill JI, Oum TH, Sawaki K, Thretheway MW (1990) Allocation of airline seat between stochastically dependent demands. Transportation Science 24(3):183–192

[6] Brumelle SL, Walczak D (2003) Dynamic airline revenue management with multiple semi-Markov demand. Operations Research 51(1):137–148

[7] Curry RE (1990) Optimal airline seat allocation with fare classes nested by origins and destinations. Transportation Science 24(3):193–204

[8] Helm WE, Waldmann KH (1984) Optimal control of arrivals to multi-server queues in a random environment. Journal of Applied Probability 21(3):602–615

[9] Hinderer K, Waldmann KH (2005) Algorithms for countable state Markov decision models with an absorbing set. SIAM J. Control and Optimization 43(6):2109–2131

[10] Kleywegt AJ, Papastavrou JD (1998) The dynamic and stochastic Knapsack problem. Operations Research 46(1):17–35

[11] Lautenbacher CJ, Stidham SJr (1999) The underlying Markov decision process in the single-leg airline yield management problem. Transportation Science 33(2):136–146

[12] Lee TC, Hersh M (1993) A model for dynamic airline seat inventory control with multiple seat bookings. Transportation Science 27(3):252–265

[13] Liang Y (1999) Solution to the continuous time dynamic yield management model. Transportation Science 33(1):117–123

[14] Littlewood K (1972) Forecasting and control of passengers. 12th AGIFORS Symposium Proceedings, pp. 95–117

[15] Robinson LW (1995) Optimal and approximate control policies for airline booking with sequential nonmonotonic fare classes. Operations Research 43(2):252–263

[16] Schäl M (1975) Conditions for optimality in dynamic programming and for the limit of n-stage optimal policies to be optimal. Z. Wahrscheinlichkeitstheorie verw. Gebiete 32:179–196

[17] Shaked M, Shanthikumar JG (1988) Stochastic convexity and its applications. Advances in applied probabilities 20(2):427–446

[18] van Slyke R, Young Y (2000) Finite horizon stochastic Knapsacks with applications to yield management. Operations Research 48(1):155–172

[19] Stidham SJr (1978) Socially and individually optimal control of arrivals to a GI/GI/1 queue. Management Science 24(15):1598–1610

[20] Subramanian J, Stidham SJr, Lautenbacher CJ (1999) Airline yield management with overbooking, cancellations, and no-shows. Transportation Science 33(2):147–167

[21] Talluri KT, van Ryzin GJ (2004) The Theory and Practice of Revenue Management. Kluwer Academic Publishers, Dordrecht

[22] Walczak D (2001) Dynamic modelling approaches to airline revenue management. Ph.D. thesis at the Centre for Transportation Studies, University of British Columbia

[23] Wollmer RD (1992) An airline management model for a single leg route when lower fare classes book first. Operations Research 40(1):26–37

Immaterielle Input-Output-Systeme

Werner Dinkelbach

BWL, insbesondere Operations Research
Universität des Saarlandes

1 Input-Output-Systeme

„Der Sinn aller betrieblichen Betätigung besteht darin, Güter materieller Art zu produzieren oder Güter immaterieller Art bereitzustellen. Güter materieller Art bezeichnen wir als Sachgüter oder auch als Sachleistungen, Güter immaterieller Art als Dienste oder Dienstleistungen“ (Gutenberg 1951, S. 1).

Ein Vorgang, der zu Veränderungen von materiellen oder immateriellen Gütern führt, wird Transformation (oder Transformator, Throughput, o.ä.) genannt (vgl. Abbildung 1). Die eingesetzten wie auch die ausgebrachten Güter lassen sich durch die Angabe der Merkmalsausprägungen, die im Hinblick auf das zu befriedigende Bedürfnis und die gewählte Transformation wesentlich sind, charakterisieren. Diese Merkmale können qualitativ, quantitativ oder auch sowohl qualitativ als auch quantitativ sein.

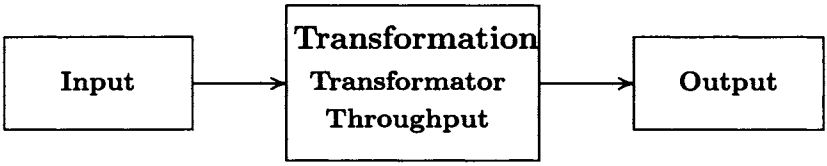


Abbildung 1: Input-Output-System

Ein Input-Output-System kann in erster Annäherung als ein spezielles System aufgefasst werden, das Güter als Input aufnimmt und diese Güter in transformierter Form als Output wieder abgibt, wobei Input und Output durch Beziehungen in unterschiedlicher Weise verknüpft sind. Ein **Input-Output-System** ist ein System aus den drei Komponenten Input, Transformation und Output, bei dem zwischen den Komponenten Input und Transformation einerseits sowie zwischen Transformation und Output andererseits ganz bestimmte Beziehungen bestehen (vgl. Abbildung 1). Statt von Input-Output-Systemen spricht man auch von Input-Output-Modellen als einer formalen Abbildung von realen Phänomenen mit einer analogen dreiteiligen Struktur.

Unter einem **materiellen Gut** wird hier ein physisches Gut verstanden, das einen messbaren ökonomischen Wert hat. Materielle Güter heißen auch Sachgüter. Unter einem **immateriellen Gut** wird hier ein nichtphysisches Gut verstanden, dem ein messbarer ökonomischer Wert zugemessen wird. Immaterielle Güter werden üblicherweise auch als Dienstleistungen bezeichnet. – Immaterielle Güter sind in erster Linie Informationen und menschliche Arbeitsleistungen. Dadurch, dass Informationen, etwa ein Zeugnis, ein Flugticket oder eine Bilanz, auf Papier geschrieben werden, und dadurch, dass eine Krankenpflegerin einen Patienten im Rollstuhl transportiert oder ein Cellist eine Sonate spielt, wird die menschliche Arbeit nicht zu einem materiellen Gut (vgl. Dinkelbach/Rosenberg 2004, S. 4ff). – Im Abschnitt 2 werden ausschließlich immaterielle Input-Output-Systeme vorgestellt, die dann im Abschnitt 3 als Dienstleistungen interpretiert werden.

2 Immaterielle Input-Output-Systeme

Zu den immateriellen Input-Output-Systemen gehören solche Input-Output-Systeme, mit denen vor allem immaterielle Inputgüter in immaterielle Outputgüter transformiert werden. Zwar umfassen hierbei der Input und der Output vielfach auch materielle Güter, die aber für den Output des Systems nicht prägend sind. Alle vier immaterielle Input-Output-Systeme illustrierende Beispiele dieses Abschnitts basieren auf Veröffentlichungen von *Klaus Neumann*. Je zwei der Beispiele sind deterministisch bzw. stochastisch.

2.1 Deterministische immaterielle Input-Output-Systeme

2.1.1 Prognoseverfahren

„Bei den ... Prognoseverfahren handelt es sich um Verfahren der *Zeitreihenanalyse*. Hierbei wird etwa aus der beobachteten Nachfrage in der gegenwärtigen Planungsperiode t und den vorangehenden Perioden $t-1, t-2, \ldots$ auf die Nachfrage in der folgenden $(t+1)$-ten Periode geschlossen ... . Ein sehr einfaches, in der Praxis häufig angewandtes Prognoseverfahren ist die *Methode des gleitenden Durchschnitts*„ (Neumann 1996, S. 10f).

Die Anzahl der immateriellen Ergebnisse (Messungen der letzten Perioden z.B. Verbräuche an materiellen Werkstoffen in einer Maschinenfabrik) können als **Input** eines Input-Output-Systems betrachtet werden. Dieser immaterieller Input wird durch ein numerischen Verfahren in den immate-

riellen Output – hier prognostizierte Verbrauchsmengen eines Werkstoffes in der kommenden Periode – transformiert.

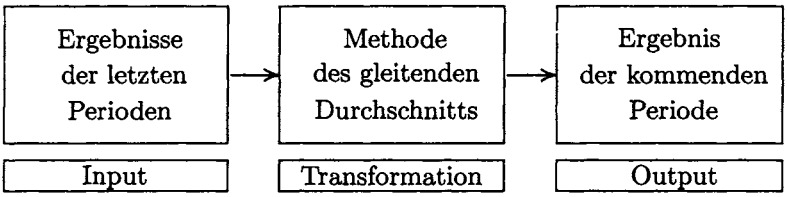


Abbildung 2: Prognoseverfahren

Durch den Einsatz der Methode des gleitenden Durchschnitts als **Transformation**(-sfunktion) errechnet sich der einelementige deterministische Output aus dem mehrelementigen deterministischen Input. Der **Output** ist in diesem Fall das Ergebnis einer elementaren Rechenaufgabe. Sie kann auch als degeneriertes Optimierungsproblem mit einelementiger Alternativenmenge, das sich als deterministisches immaterielles Input-Output-System modellieren lässt, formuliert werden (vgl. Abbildung 2).

2.1.2 Kapitalanlage

„Ein Lager hat die Funktion eines Puffers innerhalb des Güterstromes, der bei einem Produktionsprozeß (oder Distributionsprozeß) vom Einkauf über gegebenenfalls verschiedene Produktionsstufen zum Verbraucher fließt“ (Neumann und Morlock 2002, S. 621).

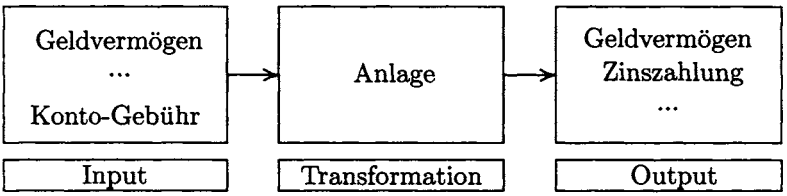


Abbildung 3: Kapitalanlage

Die Molekularbiologie-Studentin Jasmin verfügt über ein Geldvermögen von 20.000 $. Sie beabsichtigt, dieses Vermögen für ein Jahr als Festgeld zu 3% p.a. gebührenfrei bei der A-Bank oder zu 4,5% p.a. und einer Kontogebühr pro Jahr von 1%, die sofort zahlbar ist, bei der B-Bank anzulegen. Ihr Ziel ist es, ihr Vermögen zu maximieren.

Dieses als Kapitalanlage zu bezeichnendes Input-Output-System hat als wesentlichen **Input** die Faktoren einzuzahlendes Geldvermögen und zu zahlende Kontogebühren, wobei in der A-Bank die Gebühr null $ beträgt. Die **Transformation** lässt sich durch eine Wachstumsfunktion, die in Abhängigkeit vom Input und der Zeit definiert ist, problemadäquat abbilden. Sie gibt an, auf welchen Betrag das angelegte (gelagerte) Geldvermögen in einem Jahr anwächst. Der immaterielle **Output** nach Ablauf eines Jahres besteht aus den zwei Komponenten, dem zu erhaltenden Geldvermögen und den zu beanspruchenden Habenzinsen (vgl. Abbildung 3). – Berücksichtigt man noch die Zielsetzungen der Studentin, dann kann auf der Grundlage des skizzierten, immateriellen Input-Output-Systems das Problem als deterministisches Optimierungsmodell formuliert werden.

2.2 Stochastische immaterielle Input-Output-Systeme

2.2.1 Abnahmeprüfung

„Bei der *Abnahmeprüfung* ... steht die Überprüfung der Qualität einzelner Produkte im Vordergrund. Hierbei wird getestet, ob ein (Fertigungs-)Los eines (Vor-, Zwischen- oder End-)Produktes einem geforderten Standard entspricht oder nicht. Hierzu dienen so genannte *Prüfpläne*, die angeben, wie groß bei vorgegebenem Losumfang die Anzahl der dem Los zu entnehmenden Stücke, der *Stichprobenumfang*, sein soll und unter welchen Bedingungen das Los akzeptiert werden kann“ (Neumann 1996, S. 280).

Auf der einen Seite wird eine Abnahmeprüfung bei einer Massenfertigung von weniger wertvollen materiellen Gütern selten die gesamte Produktionsmenge umfassen. Auf der anderen Seite ist eine umfassende Abnahmeprüfung aller produzierten Einheiten bei gefährlichen, insbesondere explosiven materiellen Gütern oft gesetzlich vorgeschrieben.

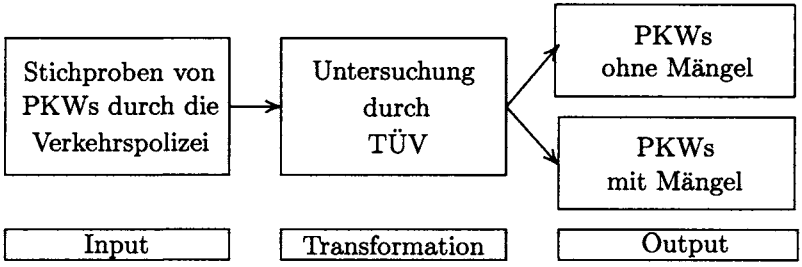


Abbildung 4: Abnahmeprüfung

Auf der vielbefahrenen Bundesstraße von Astadt nach Bedorf haben sich

in den letzten Monaten unerklärlich viele PKW-Unfälle ereignet. Die Verkehrspolizei vermutet auf Grund gewisser Hinweise, dass PKWs in dieser Gegend vielfach nicht vorschriftsmässig gewartet zu sein scheinen. Sie erinnert sich an einen TÜV (Technischen Überwachungs-Verein), der nur unweit von der genannten Bundesstrasse in Caberg liegt. Polizei und TÜV vereinbaren, für zwei Monate wöchentlich an zwei jeweils verschiedenen Tagen eine Stichprobe von 5 PKWs aus dem fließenden Verkehr auszuwählen und auf Sicherheitsmängel zu untersuchen (**stochastischer Input**). Der TÜV prüft ohne Verzögerung die sicherheitsrelevanten Merkmale dieser Fahrzeuge (**Transformation**). Soweit bei einer Überprüfung keine Mängel erkennbar sind, kann ein PKW, so wie er gekommen ist, die Prüfstation wieder verlassen, d.h., es liegt keine Veränderung des Informationsstandes vor (**Input** und **Output** sind identisch). Anderenfalls werden PKWs mit Mängel einer Werkstatt übergeben, die eine Reparatur oder eine Entsorgung in die Wege leitet. Der immaterielle Zustand des Fahrzeugs hat sich verändert. In diesem Fall unterscheiden sich **Input** und **Output**. Der Output des Systems ist stochastisch.

2.2.2 Stochastische Lagerhaltung

„Ein Lager kann als ein Puffer irgendwo innerhalb eines Güterstromes aufgefaßt werden, der vom Produzenten zum Verbraucher fließt. Es dient beispielsweise dazu, Produktions- und Nachfrageschwankungen auszugleichen, es kann einen gewissen Zeitausgleich zwischen dem Eintreffen und Verteilen von Gütern an einem bestimmten Ort schaffen“ (Neumann 1977, S. 202).

Eine bekannte Maschinenfabrik produziert u.a. hochwertige Rasenmäher für private Nutzung. Der Absatz fällt jedes Jahr im Wesentlichen in die Sommerzeit, wobei mit Absatzschwankungen zu rechnen ist. Auf der einen Seite kann die Nachfrage die Fertigung mengenmäßig übersteigen, so dass dem Hersteller einen Erlös entgeht, der im allgemeinen als Fehlkosten (Fehlmengenkosten, c_f) bezeichnet wird. Auf der anderen Seite ist es nicht ausgeschlossen, dass ein Teil der Produktion nicht im gleichen Jahr verkauft werden kann und bis zum Verkauf im nächsten Jahr gelagert werden muss, wofür Lagerkosten (Lagerhaltungskosten, c_l) anfallen.

Wenngleich die Anzahl der Rasenmäher eigentlich nur durch natürliche Zahlen gemessen werden kann, wird in der Literatur – und auch hier – zur vereinfachten Problemlösung die Zahl der produzierten Rasenmäher mit $x \in \mathbb{R}_+$ gemessen. Damit belaufen sich die zufallsabhängigen Lagerhaltungskosten $k_l(x;\gamma)$ mit der vom Zufall abhängigen Nachfragemenge

$\gamma \in \Gamma \subset \mathbb{R}_+$ (vgl. Abbildung 5):

$$k_l(x;\gamma) := \begin{cases} c_l(x-\gamma) & \text{für } x-\gamma > 0 \\ 0\,(x-\gamma) & \text{für } x-\gamma < 0. \end{cases}$$

Die Fehlmengenkosten $k_f(x;\gamma)$ lassen sich analog durch

$$k_f(x;\gamma) := \begin{cases} c_f(\gamma-x) & \text{für } \gamma-x > 0 \\ 0\,(\gamma-x) & \text{für } \gamma-x < 0 \end{cases}$$

berechnen (vgl. Abbildung 5). Die erwarteten zu minimierenden Gesamtkosten lauten damit:

$$E[k_l(x;\gamma)+k_f(x;\gamma)] := c_l \int_0^x (x-\gamma) f(\gamma) d(\gamma) + c_f \int_x^\infty (\gamma - x) f(\gamma) d(\gamma).$$

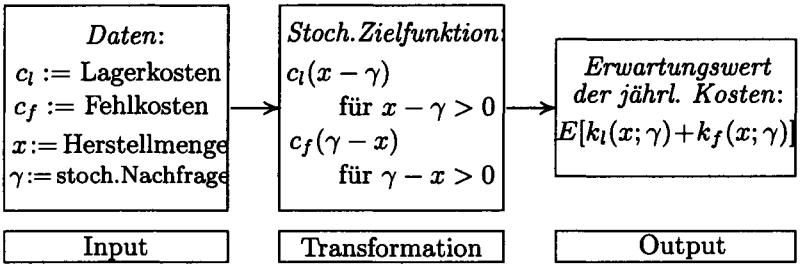


Abbildung 5: Stochastische Lagerhaltung

In der Literatur ist es in diesem Zusammenhang vielfach üblich, mit der Normalverteilung zu arbeiten. Geht man in diesem Beispiel von $\mu = 100$ und $\sigma^2 = 625$ sowie von $c_l = 345$ € und $c_f = 655$ € aus, dann führt die optimale Lösung zu $x \approx 110$ Stück mit einem minimalen Kostenerwartungswert in Höhe von ≈ 9.211 € .

Das dargestellte stochastische Lagerhaltungproblem ist nicht nur im Zusammenhang mit erweiterten Lagerhaltungsmodellen verbreitet, sondern auch als stochastische Entscheidungsmodelle, aber auch als Demonstrationen für Input-Output-Systeme von Interesse. Der **Input** umfaßt als Daten die üblichen Kosten (c_l, c_f), die zu definierende Herstellungsmenge (x) als zugehörige Entscheidungsvariable sowie zur Definition der stochastischen Nachfrage (γ). Die **Transformation**(sfunktion) betrifft die stochastische Zielfunktion, die zusammen mit dem Erwartungswert der jährlichen Kosten den **Output** generiert.

3 Dienstleistungen versus immaterielle Input–Output–Systeme

Was aber sind Dienstleistungen? Einerseits finden sich in der Literatur zahlreiche Versuche von Dienstleistungsdefinitionen. „In den Beispielen ... werden materielle Güter hergestellt bzw. gelagert. ... In den Beispielen ... sind der Input und der Output nicht materielle Güter" (Dinkelbach 1981, Sp. 750). „Um den derzeitigen Stand der produktionstheoretischen Durchdringung des Phänomens ‚Dienstleistung' aufzuzeigen, wird im folgenden die produktionswirtschaftliche Beschreibung von Produktionen als Systeme mit den Elementen Input, Throughput und Output herangezogen" (Corsten/Gössinger 2005, S. 154). „Eine Dienstleistung stellt die Produktion eines Bündels überwiegend immaterieller Güter unter Einsatz interner und externer Faktoren dar" (Schweitzer 2003, S. 62). „Oft sind Sach- und Dienstleistungsprozesse aus technischer Sicht sogar identisch" (Dyckhoff 2003, S. 721). Dem stehen allerdings andererseits einige allgemein diskutierbare Formulierungen gegenüber. „Eine eingehende Analyse zentraler Forschungsarbeiten zu Dienstleistungen zeigt, daß eine allgemeingültige Definition des Begriffs Dienstleistung in der Literatur nicht existiert" (Weiber/Billen 2005, S. 89).

In Fortführung des Beispiels 2.1.1 ist festzuhalten, dass die Verbräuche einer bestimmten Abteilung durch Mitarbeiter oder automatisch über einen gewünschten Zeitraum additiv zu ermitteln sind, um eine für die Zukunft gewünschte Kennzahl berechnen zu können. Es handelt sich um eine einteilige, zweiteilige oder dreiteilige, interne oder externe, aber stets um eine immaterielle **Dienstleistung** zur Planungunterstützung.

Um eine deterministische optimale Kapitalanlage ringt Jasmin im Input-Throughput-Output-System des Beispiels 2.1.2. Sie berechnet die Endwerte nach einem Jahr und bestimmt den maximalen Endwert. Das könnte auch die Aufgabe eines Finanzberaters – eines **Finanzdienstleisters** – sein.

Im Beispiel 2.2.1 stehen Tätigkeiten des TÜV (Technischer Überwachungs-Verein) im Mittelpunkt. In der Inputphase geht es darum, systematisch Stichproben aus dem PKW-Verkehr zu gewinnen. Die ausgewählten PKWs sind auf Sicherheitsmängel zu prüfen. Durch die Prüfung werden die immateriellen Merkmale der PKWs verändert und als Output erhält man in Abhängigkeit vom Prüfungsergebnis PKWs, die unterschiedlichen Kategorien zugeordnet sind. Die vom TÜV durchgeführten Aktivitäten sind immateriell, es sind Dienstleistungen, die zu immateriellen Ergebnissen führen. Der TÜV produziert nichts materielles, er ist ein typisches **Dienstleistungsunternehmen.**

Das stochastische Lagerhaltungsmodell des Beispiels 2.2.2 berührt auf der einen Seite den Hersteller: er muss per anno die erwarteten Gesamtkosten ermitteln und die zugehörige kostenminimale Herstellmenge bestimmen. Auf der anderen Seite melden Käufer hin und wieder zufällig ihre Nachfragewünsche an. Da die stochastische Nachfrage weder dem Hersteller noch einem Dienstleistungsunternehmen bekannt sein dürfte, kann möglicherweise letzteres auf Grund langjähriger Erfahrung gewisse Andeutungen geben. – Wenn der Sachbearbeiter für Lagerhaltungsprobleme das Unternehmen spontan aus familiären Gründen verlassen müsste, bliebe dem Firmenchef lediglich als Ersatz, eine **Dienstleistungsgesellschaft** mit OR-Kompetenz anzuwerben.

Literatur

[1] Corsten H, Gössinger R (2005) Entwurf eines produktionstheoretischen Rahmens für Dienstleistungen. In: Corsten H, Gössinger R, Dienstleistungsökonomie. Duncker & Humblot, Berlin, S. 153–188

[2] Dinkelbach W (1981) Input-Output-Analyse. In: Handwörterbuch des Rechnungswesens. Poeschel, Stuttgart, Sp. 749–761

[3] Dinkelbach W, Rosenberg O (2004) Erfolgs- und umweltorientierte Produktionstheorie, 5. Auflage. Springer, Berlin

[4] Dyckhoff H (2003) Neukonzeption der Produktionstheorie. Zeitschrift für Betriebswirtschaftslehre 73:705–732

[5] Gutenberg E (1951) Grundlagen der Betriebswirtschaftslehre, 1. Auflage. Springer, Berlin

[6] Neumann K (1977) Operations Research Verfahren, Band II. Hanser, München

[7] Neumann K (1996) Produktions- und Operations-Management. Springer, Berlin

[8] Neumann K, Morlock M (2002) Operations Research. Hanser, München

[9] Schweitzer, M (2003) Taktische Planung von Dienstleistungskapazitäten. Duncker & Humblot, Berlin

[10] Weiber R, Billen P (2005) Informationsökonomische Fundierung des Dienstleistungsmarketing. In: Corsten H, Gössinger R, Dienstleistungsökonomie. Duncker & Humblot, Berlin S. 85–127

Audit-Staff Scheduling by Column Generation

Andreas Drexl

Institut für Betriebswirtschaftslehre
Universität Kiel

Johannes Frahm

Vattenfall GmbH
Hamburg

Frank Salewski

Institut für Betriebswirtschaftslehre
Universität Kiel

1 Introduction

When scheduling its audit-staff, the management of an auditing firm encompasses a number of decisions. These may be grouped into several categories which differ markedly in terms of organizational echelon involved, length of the planning horizon and the planning periods, degree of aggregation of the audit tasks, degree of detail of the required information, and decision objective. However, traditional audit-staff scheduling models (Balachandran and Zoltners 1981, Chan and Dodin 1986, Gardner et al. 1990, Dodin and Chan 1991, Drexl 1991, Dodin and Elimam 1997, Dodin et al. 1998, Brucker and Schumacher 1999, Rolland et al. 2005) are single-level models which try to construct a direct assignment of auditors to tasks and periods. To facilitate algorithmic treatment, all these models are more or less gross simplifications of practical planning situations.

These observations led us to conduct a survey among the 200 biggest certified public accountant (CPA) firms in Germany. Based upon its results we formulated an hierarchical model (Salewski and Drexl 1993, Salewski 1995) comprising three levels: The medium-term planning assigns teams of auditors to the engagements; it constructs a schedule by determining the workload per auditor and week over a planning horizon of between three and twelve months. The medium-to-short-term planning disaggregates the results of the first level for one week and all auditors; the outcome is a schedule for each auditor that covers – on the basis of periods of four hours – all engagements in which he is involved in the considered week. The

short-term planning is based upon the results of the second level for one week and one engagement; it assigns the auditors involved in the auditing of that engagement to the corresponding audit tasks and schedules these tasks to periods of one hour. Here, we will focus on the first, that is, the medium-term level.

The paper is organized as follows: In Section 2 we define the problem formally and investigate where it is positioned in the context of audit-staff/project scheduling. In Section 3 the problem is reformulated as a set partitioning problem with an exponential number of columns. The LP-relaxation of this model can be solved to optimality by column generation. Next we show in Section 4 that the columns of the LP-relaxation can be efficiently computed by means of a shortest path model. The description of the test bed is provided in Section 5. Section 6 presents the results of an in-depth computational study. Finally, Section 7 gives a brief summary, along with our conclusions.

2 Problem Setting

The Medium-Term Audit-Staff Scheduling Problem (MASSP) may be characterized by the following assumptions (cp. Salewski et al. 1997 also):

- A firm employs one or more *auditors*, who have to audit one or more *engagements* within a given *planning horizon* of normally 13, 26, or 52 weeks. A period has a duration of one week.

- Each engagement is made up of one or more *phases*, e.g. preliminary, intermediate and final audit. The phases of one engagement must be processed in a strictly linear order. This implies that each phase of an engagement, except for the first one, has exactly one predecessor. Work on some phases may not commence before a specific *release time*, and may have to be completed before a specific *deadline*.

- Each phase has a specific duration in terms of multiples of periods. In order to allow preemption of phase-processing at the end of some periods each phase is decomposed into as many subphases with one period duration as its processing takes in total. E.g. a phase with a processing time of two periods is splitted into two subphases each having a duration of one period (cp. Figure 1).

  Mode-dependent *minimum and maximum time-lags* are given between subsequent phases of one engagement and between subsequent subphases of one phase, respectively (for details see below).

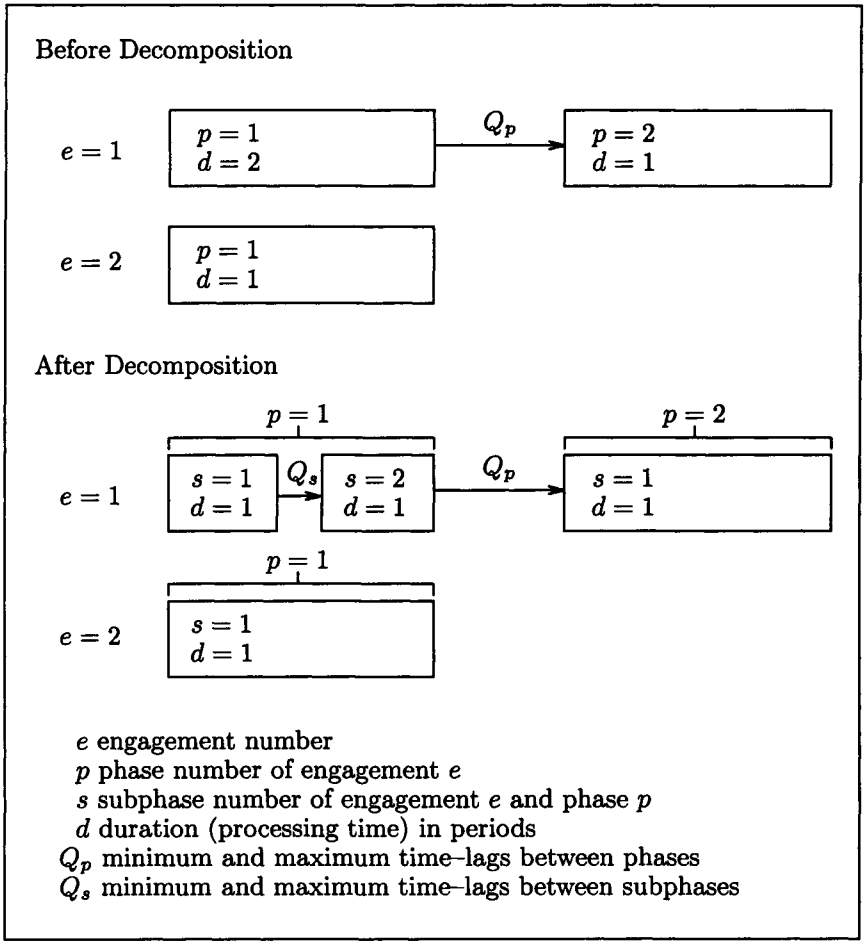


Figure 1: Decomposition of Phases into Subphases

- The *availability* of some auditors may be restricted in certain periods, e.g. due to holidays or vacation. In addition, for some periods a client may want to limit the time during which the auditing takes place (*maximum processing time*), e.g. due to vacation periods or stock-taking activities.

- Often an engagement could be audited by several alternative audit teams (*modes*). Different team compositions will result in different auditor processing times. Usually some modes will be preferable to others: Factors influencing the suitability of an auditor for a specific engagement are e.g. qualification level, industry experience, familiarity with the clients business, and degree of difficulty of the audit tasks. The preferability of a mode as a whole may e.g. be linked to the total processing time needed. Hence, a preference value will be assigned to each mode.

The objective then is to assign the overall best-suited teams to the engagements (*mode assignment with maximization of preferences*), and to determine when the individual subphases are to be executed (subphase scheduling). A detailed example which illustrates why these assumptions are justified with respect to what can be observed in practice can be found in Salewski (1995).

The problem parameters of the MASSP are summarized as follows:

A	:	number of auditors, indexed by a
E	:	number of engagements, indexed by e
v^{em}	:	preference value corresponding to the processing of engagement e in mode m
C_{at}	:	capacity of auditor a in period t
D_{et}	:	maximum processing time of engagement e in period t
δ_{ep}	:	deadline of phase (e,p)
k_{epsma}	:	time auditor a needs to process subphase (e,p,s) in mode m (capacity usage)
λ_{ep}	:	release time of phase (e,p)
M_e	:	number of modes of engagement e, indexed by m
P_e	:	number of phases of engagement e, indexed by p
$q_{epsp's'm}$	:	minimum (finish-to-start) time-lag between subsequent subphases (e,p,s) and (e,p',s') when processing e in mode m

$\hat{q}_{epsp's'm}$: maximum (finish-to-start) time-lag between subsequent subphases (e,p,s) and (e,p',s') when processing e in mode m

S_{ep} : number of subphases of phase (e,p), indexed by s; w.l.o.g. each subphase has a duration of one period

T : number of periods, indexed by t

V_{eps} : set of all immediate predecessors of subphase (e,p,s)

The mode concept is very fundamental in order to establish the relationship between engagements and auditors. For the sake of clarity consider, e.g., engagement $e = 2$ of the example provided in Figure 1. Assume that this engagement can be processed alternatively by auditors 1 or 3 or jointly by auditors 2 and 3. In this situation we define three modes related to the three 'teams' consisting of auditor 1 (team 1), auditor 3 (team 2), and auditors 2 and 3 (team 3), respectively. Formally, the set of auditors $\Lambda(e,m)$ belonging to mode m of engagement e is defined via positive capacity usages, that is $\Lambda(e,m) = \{a \in \{1,\ldots,A\} \mid \exists\, s \in \{1,\ldots,S_{ep}\}, p \in \{1,\ldots,P_e\},$ with $k_{epsma} > 0\}$.

Note, precedence relations expressed via V_{eps} exist only between subphases belonging to the same engagement. This characteristic is called 'isolating' in Salewski et al. (1997).

The problem under consideration is formulated as a binary optimization problem in Salewski et al. (1997). Furthermore, it is shown, that the MASSP is a special case of the more general project scheduling problem with resource and, additionally, mode identity constraints (cp. the comprehensive survey given in Brucker et al. 1999 also). In addition, it is proven that the (feasibility variant of the) MASSP is (strongly $\mathcal{NP}$-complete) strongly $\mathcal{NP}$-hard.

In order to keep this paper self-contained we reproduce the project scheduling-based binary optimization model in Appendix A. There, the indices e,p,s are replaced according to Equation (15) in order to simplify the mathematical formulation. Clearly, this transformation could be done already at the end of this section. Unfortunately, this is disadvantageous because then we would have to attach attributes to jobs in Section 4, especially in Equations (6)-(11). (For example, the attribute e has to be attached to the job number because of the mode identity constraints.) Hence, we stay with the notation introduced previously which explains things in terms of the area of application covered in this work.

Throughout this paper we make use of the illustrative example which is provided in Appendix B.

3 Set Partitioning Model

The basic idea is to iteratively compute sequences for each of the engagements by means of a shortest (in fact a longest) path model. From the set of sequences on hand those are chosen via a set partitioning model which respect the capacity constraints of the auditors (in the LP-relaxation). A description of basic column generation techniques can be found in, e.g., Bradley et al. 1977.

In order to describe the set partitioning model formally we use the following parameters and variables:

$\mathcal{S}^e$	:	set of columns representing sequences for engagement e, index i
$e(i)$	:	engagement e column i is associated with
$m(i)$	:	mode m column i is associated with $(1 \leq m \leq M_{e(i)})$
b_{iat}	:	capacity usage, if engagement $e(i)$ is processed within sequence i by auditor a in period t
c_i	:	objective function coefficient of sequence i related to engagement $e(i)$ and mode $m(i)$, that is, $c_i = v^{e(i),m(i)}$
y_i	:	1, if sequence i is part of the optimal solution (0, otherwise)

In column i it is defined by means of the capacity usage b_{iat} that engagement e is processed in period t. Note that both the engagement $e(i)$ and the mode $m(i)$ are well-known for sequence i. Hence, the cost $c_i = v^{e(i),m(i)}$ are also well-known.

In the following section it will be explained in detail how for each tupel of engagement and mode (e, m), sequences can be computed by solving shortest path problems. Apparently, this essentially renders each (e, m) subgraph, $1 \leq e \leq E, 1 \leq m \leq M_e$, to consist of nodes $\boxed{p, s, t}$ solely (cp. Figure 2 also). In order to simplify the presentation we relabel the nodes $j \leftarrow (p, s, t)$ similarly to what is done in Equation (15) in Appendix A. Then $p(j)$ denotes phase p node j is associated with, $s(j)$ is the subphase s node j is associated with, and $t(j)$ denotes the period t node j is associated with, respectively. This relabeling facilitates to calculate the capacity usages b_{iat} as follows:

$$b_{iat} = \sum_{p=1}^{P_{e(i)}} \sum_{s=1}^{S_{e(i),p}} \begin{cases} k_{e(i),ps,m(i),a}\,, & \text{if } p = p(j), s = s(j) \text{ and } t = t(j) \\ 0\,, & \text{otherwise} \end{cases} \tag{1}$$

Based on these definitions the (restricted) master problem can be stated by Equations (2)-(5) as a set partitioning type model.

The objective (2) is to select a subset of columns at optimum costs. Equations (3) make exactly one sequence i per engagement e to be part of the solution and, hence, exactly one mode $m(i)$, that is a team of auditors, is chosen also for each engagement. Inequalities (4) require to respect the capacity constraints of the auditors in each of the periods of the planning horizon. Finally, restrictions (5) define the decision variables to be binary-valued.

$$\max \sum_{e=1}^{E} \sum_{i \in \mathcal{S}^e} c_i \, y_i \tag{2}$$

$$\text{s.t.} \sum_{i \in \mathcal{S}^e} y_i = 1 \qquad e = 1, \ldots, E \tag{3}$$

$$\sum_{e=1}^{E} \sum_{i \in \mathcal{S}^e} b_{iat} \, y_i \leq C_{at} \qquad a = 1, \ldots, A, \; t = 1, \ldots, T \tag{4}$$

$$y_i \in \{0, 1\} \qquad e = 1, \ldots, E, \; i \in \mathcal{S}^e \tag{5}$$

4 Shortest Path Model

As already mentioned in the previous section for each tupel of engagement and mode (e, m), sequences are computed by solving shortest path problems. The relabeling of nodes $h \leftarrow (p, s, t)$ and $j \leftarrow (p', s', t')$ has to be done for each (e, m) subgraph. Now, let denote $\mathcal{N}^{em}$ the set of nodes of the graph associated with engagement e and mode m and $\mathcal{A}^{em}$ the set of arcs of the graph associated with engagement e and mode m.

Figure 2 illustrates the preliminary shortest path graph for engagement $e = 2$ and mode $m = 2$ of the instance provided in Appendix B. This shortest path graph shows that – for a given tupel (e, m) – two nodes $(p(h), s(h), t(h))$ and $(p(j), s(j), t(j))$ are connected by an arc only if the conditions (7) to (11) are met. Note that nodes can be connected only if h and j belong to the same engagement e ('isolating' precedence structure – cp. (6)) and if they are processed in the same mode.

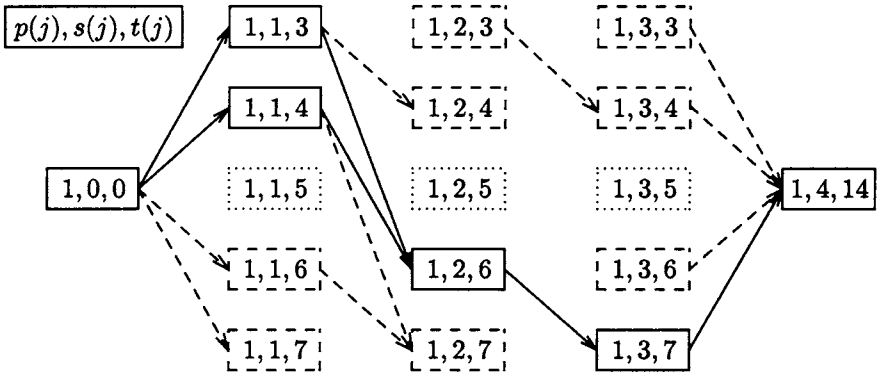


Figure 2: Preliminary Shortest Path Graph – $e = 2$ and $m = 2$

$$(e, p(h), s(h)) \in V_{e,p(j),s(j)} \tag{6}$$

$$t(j) - t(h) \geq q_{e,p(h),s(h),p(j),s(j),m} + 1 \quad h, j \in \mathcal{A}^{em}, h \in V_{e,p(j),s(j)} \tag{7}$$

$$t(j) - t(h) \leq \hat{q}_{e,p(h),s(h),p(j),s(j),m} + 1 \quad h, j \in \mathcal{A}^{em}, h \in V_{e,p(j),s(j)} \tag{8}$$

$$\sum_{a=1}^{A} k_{e,p(j),s(j),m,a} \leq D_{e,t(j)} \quad j \in \mathcal{N}^{em} \tag{9}$$

$$\lambda_{e,p(h)} \leq t(h) \quad \text{and} \quad \lambda_{e,p(j)} \leq t(j) \quad h, j \in \mathcal{N}^{em} \tag{10}$$

$$\delta_{e,p(h)} \geq t(h) \quad \text{and} \quad \delta_{e,p(j)} \geq t(j) \quad h, j \in \mathcal{N}^{em} \tag{11}$$

Let us give some comments on Figures 2 and 3:

- The nodes (1,0,0) and (1,4,14), where we set $4 = S_{e,P_e} + 1 = S_{21} + 1$ and $14 = T + 1$, are dummy source and sink nodes, respectively.

- The nodes (1,1,5), (1,2,5) and (1,3,5) are dotted, because $D_{25} = 0$ does not allow to schedule these subphases in period $t = 5$, and, hence, they can be eliminated. Clearly, incident arcs have not to be considered also.

- There must be no arc connecting nodes (1,1,3) and (1,2,7) because of the maximal time lag of 2 periods.

- The dashed nodes (1,2,3), (1,3,3) and (1,3,4) cannot be reached because of minimal time-lags and, henceforth, are eliminated.

- There is no outgoing arc from the dashed nodes (1,1,7) and (1,2,7) because of minimal time lags in connection with the deadline. Removing node (1,2,7) in turn produces node (1,1,6) to have no successor node and, hence, it can be eliminated also. Similarly, for node (1,2,4). Likewise, the dashed node (1,3,6) has no ingoing arc and can be eliminated too.

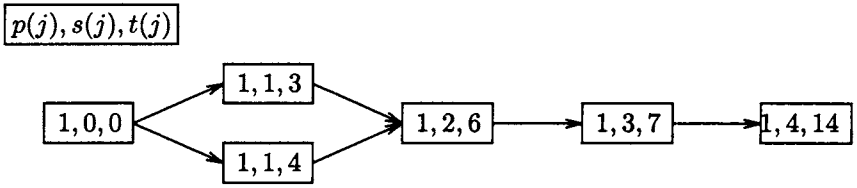


Figure 3: Reduced Shortest Path Graph – $e = 2$ and $m = 2$

Summarizing, Figure 3 provides the reduced shortest path graph where all the dotted and dashed nodes and the incident (dashed) arcs have been eliminated.

Now we are going to explain how the arc weights are calculated. Without loss of generality, consider any engagement-mode-tupel (e, m), i.e. subgraph of the overall shortest graph. Furthermore, consider any pair of related nodes $h \in \mathcal{N}^{em}$ and $j \in \mathcal{N}^{em}$ of the reduced shortest path (sub-) graph and, hence, the arc $(h, j) \in \mathcal{A}^{em}$ connecting both nodes. Now, let denote

d_{hj}^{em} : original weight of arc $(h, j) \in \mathcal{A}^{em}$

g_{hj}^{em} : updated weight of arc $(h, j) \in \mathcal{A}^{em}$

μ_e : dual variable associated with the one sequence per engagement constraint (3), $\mu_e \in \mathbb{R}$

π_{at} : dual variables associated with the capacity constraints (4), $\pi_{at} \geq 0$

x_{hj}^{em} : 1, if arc $(h, j) \in \mathcal{A}^{em}$ is element of the shortest path (0, otherwise)

Then Equation (12) formally defines the original weight d_{hj}^{em} of the arcs. Obviously, all arcs have to be initialized to zero except the arcs emanating from the single source node $q = (1, 0, 0)$. This way the constant preference v^{em} for each engagement-mode-tupel (e, m) is taken into account appro-

priately.

$$d_{hj}^{em} = \begin{cases} v^{em} , & (q,j) \in \mathcal{A}^{em} \\ 0 , & \text{otherwise} \end{cases} \tag{12}$$

Finally, Equation (13) explains how to calculate the weights g_{hj}^{em} of all arcs $(h,j) \in \mathcal{A}^{em}$ taking the dual variables π_{at} into account.

$$g_{hj}^{em} = d_{hj}^{em} - \sum_{a=1}^{A} k_{e,p(h),s(h),m,a}\ \pi_{a,t(h)} \tag{13}$$

Then the objective function of the shortest path model for engagement e can be stated by Equation (14).

$$Z^e = \max \left\{ \sum_{(h,j)\in\mathcal{A}^{em}} g_{hj}^{em}\ x_{hj}^{em} - \mu_e \mid m = 1, \ldots, M_e \right\} \tag{14}$$

Note that the shortest path graph is acyclic with node weights $g_{hj}^{em} \in \mathbb{R}$. Because of the topological structure, the shortest path problems are solvable in linear time.

Apparently, pricing out occurs if $\max\{Z^e \mid 1 \leq e \leq E\} \leq 0$. This is accomplished by computing the shortest path in the overall shortest path graph comprising all the engagements. In our implementation, we compute at most E columns per iteration, one for each engagement e with $Z^e > 0$ (multiple pricing).

A step-by-step description of the overall set partitioning/column generation algorithm shall now be given where we use the following notation:

$\mathcal{S}$	:	current set of columns, i.e. $\mathcal{S} = \{\mathcal{S}^1 \cup \ldots \cup \mathcal{S}^e \cup \ldots \cup \mathcal{S}^E\}$
$SPP(\mathcal{S})$	:	(restricted) master problem/set partitioning model defined for set $\mathcal{S}$
$CG(e)$	:	column generation/shortest path model for engagement e
$Z_{CG(e)}^{(\mathcal{S})*}$	:	optimal objective function value of $CG(e)$ defined for the current set of columns $\mathcal{S}$

Algorithm

1. Initialize $\mathcal{S}^e, e = 1, \ldots, E$.
2. Solve the LP-relaxation of the set partitioning model $SPP(\mathcal{S})$; $e = 0$.
3. $e = e + 1$.

4. Solve $CG(e)$; if $Z^{(\mathcal{S})*}_{CG(e)} > 0$ then add the column to $\mathcal{S}^e$ and $\mathcal{S}$.
5. If $e < E$ then go to Step 3.
6. If at least one $e \in \{1, \ldots, E\}$ with $Z^{(\mathcal{S})*}_{CG(e)} > 0$ has been computed then go to Setp 2.
7. Stop.

In Step 1, the set of columns is initialized as follows: Define the 'first' column of each engagement by setting capacity usages to zero (for all auditors and periods). Likewise, objective function coefficients are set to zero. This means that we start with a (fictitious) 'feasible' solution with a bad objective function value.

Apparently, our algorithm generates at most E columns per iteration (multiple pricing). In Section 6 we will show, that this variant produces slightly more columns than the single pricing (i.e. one column per iteration) counterpart but is faster on the average because less LPs have to be solved.

5 Test Bed

The set of instances which is used for experimental purposes is identical to the one defined in Salewski at al. (1997). In order to keep this paper self-contained we outline in the following the procedure followed for generating a sample of test instances of practical relevance. We also describe the design of an extensive experimental study conducted, along with the definitions of the performance measures used to evaluate the results of the study.

Even in current literature, the systematic generation of test instances does not receive much attention. For the well-researched field of project scheduling, Kolisch at al. (1995) report that "very little research concerned with the systematic generation of benchmark instances has been published. [...] most efforts are only briefly described."

Generally, two possible approaches can be found adopted in literature when having to come up with test instances. First, practical cases. Their strength is their high practical relevance while the obvious drawback is the absence of any systematic structure allowing to infer any general properties. Thus, even if an algorithm performs good on some practice cases, it is not guaranteed that it will continue to do so on other instances as well. Second, artificial instances. Since they are generated randomly according to predefined specifications, their plus lies in the fact that fitting them to certain requirements such as given probability distributions poses no problems. A detailed such procedure for generating project scheduling

instances has been proposed by Kolisch at al. (1995). However, they may reflect situations with little or no resemblance to any problem setting of practical interest. Hence, an algorithm performing well on several such artificial instances may or may not perform satisfactorily in practice.

Therefore, we decided to devise a combination of both approaches, thereby attempting to keep the strengths of both approaches while avoiding their drawbacks. Within the cited survey among the 200 biggest CPA firms in Germany, we asked the respective official in charge of staff planning (if existent) or one of the firm's partners to provide details about length of planning horizon, number of auditors, number and structure of audit engagements, auditor working capacities (working hours per day or week) and possible variations therein (e.g. due to vacation, training), etc. In addition, we carried out interviews with several experts in the field of auditing to clarify our understanding of the peculiarities of the auditing sector. Then, to ensure a systematic and consistent generation of the instances, for each of the parameters of the MASSP a domain and a discrete distribution function on the domain were defined, based upon the survey and the interview results. From these definitions, a test bed of representative instances was generated randomly, using a classification scheme to build instances with specific properties. In this way we tried to construct instances reflecting the specifics of audit-staff scheduling in the industry as closely as possible, yet to employ a systematic design for the generation procedure.

We assumed that only two instance-related factors do have a major influence on the performance of a solution method, viz. the size and the tractability of the instance attempted. Although the size of an instance is determined by the length of the planning horizon, the number of subphases, and the number of modes, statistical analyses of the survey results found all these to depend on the length of the planning horizon. In the sequel, three types of instances will be distinguished with respect to their size: the planning horizon equals 13 weeks for *small* instances (13 weeks with up to 30 auditors and 95 engagements), 26 weeks for *medium-size* instances (with up to 55 auditors and 280 engagements), and 52 weeks for *large* instances (with up to 125 auditors and 880 engagements). In addition, very small instances (13 weeks with up to 6 auditors and 10 engagements) were generated. While these instances are too small to bear practical relevance, they can be solved to optimality with standard MIP-solvers and thus can be used as benchmarks.

The *tractability* of an instance intends to reflect how easy or difficult it is to solve. In our study, the auditor capacities are assumed to be the only factor influencing the tractability of an instance: the higher the au-

ditor capacities are, the easier the corresponding instance is to solve since its solution space becomes larger. Accordingly, the auditor capacities are calculated from the average expected demand, adjusted by a multiplicative factor RS (resource strength). Throughout this work, three types of instances will be distinguished with respect to their tractability: *easy* instances where RS is taken equal to 3.5, *medium* instances where RS equals 2.5, and *hard* instances where RS is 1.5.

Clearly, the performance of an algorithm cannot be evaluated from running it on infeasible instances. It is therefore noteworthy that, in spite of the strong $\mathcal{NP}$-completeness of the associated feasibility problem, it was possible to rig up the design of the (complicated) generation procedure in a way guaranteeing that for each constructed instance there exists at least one provably feasible solution.

Due to the computational effort required to attempt a sample of all sizes, the scope of the experiment was limited to include only small and very small instances. Though no obstacle for using the developed methods on larger instances, this effort prevents the undertaking of a full factorial design experiment covering all instance classes. Additionally, the conjecture that methods performing well on smaller instances do so also on larger ones is widely accepted (cp. Badiru 1988, Alvarez-Valdés and Tamarit 1989). Furthermore, for instances of these sizes lower and upper bounds are available from literature in order to benchmark the results obtained with the set partitioning/column generation approach.

6 Computational Results

The methods described earlier have been imlemented using AMPL (cp. Fourer at al. 1993) and the CPLEX callable library (cp. Bixby and Boyd 1996) on an IBM RS 6000 F40 workstation with 192 MB RAM.

The computational results of our experiments are summarized in Tables 1 and 2. The CPU-times required by our experiments are summarized in Table 3. The symbols have the following meaning:

no.	:	identification number of instances
easy	:	instances whose tractability is easy (likewise medium and hard)
LB	:	best feasible solution (lower bound) known so far (computed with the tabu search procedure of Salewski 1996)
CG	:	objective function value computed by column generation
$\star$	:	column generation solution integral
UB	:	optimal objective function value of the LP-relaxation of the model (16) to (22) (upper bound)
$CPU(CG)$	:	CPU-time in sec required by the CPLEX solver for the column generation process (without AMPL times)
$CPU(LP)$	:	CPU-time in sec required by CPLEX for the solution of the LP-relaxation of the model (16) to (22)

Table 1: Computational Results – Very Small Instances

	easy			medium			hard		
no.	LB	CG	UB	LB	CG	UB	LB	CG	UB
1	43	43*	43	43	43*	43	43	43*	43
2	35	35*	35	35	35*	35	35	35*	35
3	44	44*	44	44	44*	44	43	43.81	43.81
4	44	44*	44	44	44*	44	44	44*	44
5	45	45*	45	45	45*	45	45	45*	45
6	41	41*	41	40	40.9	41	39	39.74	40.82
7	47	47*	47	47	47*	47	47	47	47
8	44	44*	44	44	44*	44	44	44	44
9	40	40*	40	40	40*	40	40	40*	40
10	47	47*	47	47	47*	47	47	47*	47

Table 2: Computational Results – Small Instances

	easy			medium			hard		
no.	*LB*	*CG*	*UB*	*LB*	*CG*	*UB*	*LB*	*CG*	*UB*
1	302	302	304.70	297	300.20	303.60	264	291.36	296.94
2	299	299	304.71	292	297.97	303.67	274	288.78	297.91
3	542	543	543	510	541.62	541.95	433	511.64	524.85
4	180	180	184	173	179.60	184	162	170.63	176.75
5	298	298	299.75	295	298	299.75	267	293.59	296.15
6	434	434	434	420	432.09	433.03	386	418.85	424.34
7	751	751	753	750	751	753	666	742.87	747.16
8	317	317	319.97	309	317	319.97	278	307.50	316.45
9	380	380	380	380	380	380	357	378.05	378.93
10	414	414*	414	413	414	414	363	403.29	409.07

Table 3: CPU-Times – Small Instances

	easy		medium		hard	
no.	*CPU(CG)*	*CPU(LP)*	*CPU(CG)*	*CPU(LP)*	*CPU(CG)*	*CPU(LP)*
1	3.42	2.92	5.30	4.51	10.81	11.35
2	3.12	1.99	4.66	2.32	7.91	8.95
3	9.69	6.85	15.88	15.24	33.81	80.85
4	1.61	0.92	3.05	1.34	3.67	2.49
5	2.72	2.22	5.54	3.54	10.88	9.56
6	4.33	3.63	5.45	8.45	11.38	23.28
7	6.22	10.56	10.81	17.01	62.92	146.95
8	2.49	2.71	5.53	6.33	15.79	18.41
9	4.24	2.81	3.26	3.78	10.75	9.25
10	3.36	3.51	6.94	4.35	19.96	20.30

Table 4: Size of the Last Master – Small Instances

no.	easy	medium	hard
1	123	143	200
2	121	130	198
3	224	279	374
4	80	100	115
5	123	141	215
6	168	195	261
7	285	328	484
8	120	138	217
9	169	165	240
10	165	202	274

Table 5: Size of the Last Master – Small Instances – One Column Per Iteration

no.	easy	medium	hard
1	126	142	186
2	126	139	180
3	214	252	333
4	81	96	107
5	127	141	202
6	173	191	253
7	296	320	428
8	127	139	203
9	169	175	217
10	168	187	262

The result tables can be interpreted as follows:

- The set partitioning/column generation approach produces an integral solution for almost all of the 30 very small instances. These results verify the lower bounds produced with tabu search to be very good also. Moreover, the LP-relaxation of the model (16) to (22) is tight for the very small instances.

- For the small instances only one data set can be solved to optimality by the column generation approach. Eight (one) of the easy (medium) instances are (is) solved to optimality also because the lower and the column generation-based upper bounds coincide.

- In general the upper bounds produced by set partitioning/column generation are far better than the upper bounds of the model (16) to (22). This is due to the following fact: In the LP-relaxation of the model (16)-(22) implicitely the constraints (21) imposed by the maximum processing times D_{et} per engagement e and per period t are relaxed also. On the other hand, these constraints are taken into account by conditions (9) when constructing the shortest path graph, and, hence, cannot be relaxed implicitely when solving the LP-relaxation of the set partitioning model.

- While the LP-relaxation of the model (16) to (22) can be solved in zero sec for the very small instances, the small ones already require a considerable amount of CPU-time. Especially for the hard instances the CPU-times increase drastically while the quality of the upper bounds deteriorates. The CPU-time required by the column generation approach for the very small instances is about half a sec.

If the LP-relaxation of the set partitioning model is not integral then the column generation approach has to be embedded into a branch-and-price framework in order to come up with an optimal integral solution (cp. Barnhart et al. 1998 and Vanderbeck 2000).

Table 4 shows the sizes of the last master problems in terms of the columns generated. Apparently, for the small instances the number of columns generated increases with increasing problem hardness. That is, easy instances need less columns than the medium ones which in turn need less than the hard ones.

For getting the results presented so far, we generated at most E columns per iteration, one for each enagement e with $Z^e > 0$ (multiple pricing). To reveal that this is indeed a good idea, we also show some results when we generated at most one column per iteration which is determined by the

overall shortest path (single pricing). Table 5 shows the figures for the size of the last master problem. Comparing this with Table 4 in general less columns have to be generated. Unfortunately, the run-time upon termination is much shorter when more than just one column is generated per iteration. This is due to the fact that far more LPs have to be solved in the case of single pricing. Hence, multiple pricing is advantageous.

7 Summary and Conclusions

When scheduling its audit-staff, the management of an auditing firm encompasses a number of decisions. These may be grouped into several categories which differ markedly in terms of organizational echelon involved, length of the planning horizon and the planning periods, degree of aggregation of the audit tasks, degree of detail of the required information, and decision objective. However, traditional audit-staff scheduling models are single-level models which try to construct a direct assignment of auditors to tasks and periods. To facilitate algorithmic treatment, all these models are more or less gross simplifications of practical planning situations.

In this paper, we introduce an audit-staff scheduling model which comprises many features being important with respect to audit management in practice. For dealing with this model, a set partitioning/column generation approach is developed. The LP-relaxation of the set partitioning problem is solved by column generation in order to compute tight upper bounds. Frequently, the solution of the continuous relaxation is integral and, hence, an optimal solution is obtained.

References

[1] Alvarez-Valdés R, Tamarit JM (1989) Heuristic algorithms for resource-constrained project scheduling: A review and an empirical analysis. In: Slowinski R, Weglarz J (eds.) Advances in Project Scheduling, Amsterdam, pp. 113–134

[2] Badiru AB (1988) Towards the standardization of performance measures for project scheduling heuristics. IEEE Transactions on Engineering Management 35:82–89

[3] Balachandran BV, Zoltners AA (1981) An interactive audit-staff scheduling decision support system. The Accounting Review 56:801–812

[4] Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: column generation for huge integer programs. Operations Research 46:316–329

[5] Bartusch M, Möhring RH, Radermacher FJ (1988) Scheduling project networks with resource constraints and time windows. Annals of Operations Research 16:201–240

[6] Bixby N, Boyd E (1996) Using the CPLEX Callable Library, CPLEX Optimization Inc, 7710-T Cherry Park, Houston

[7] Bradley SP, Hax AC, Magnanti TL (1977) Applied Mathematical Programming. Addison-Wesley, Reading

[8] Brucker P, Drexl A, Möhring RH, Neumann K, Pesch E (1999) Resource-constrained project scheduling: Notation, classification, models, and methods. European Journal of Operational Research 112:3–41

[9] Brucker P, Schumacher D (1999) A new tabu search procedure for an audit-scheduling problem. Journal of Scheduling 2:157–173

[10] Chan KH, Dodin B (1986) A decision support system for audit-staff scheduling with precedence constraints and due dates. The Accounting Review 61:726–733

[11] Dodin B, Chan KH (1991) Application of production scheduling methods to external and internal audit scheduling. European Journal of Operational Research 52:267–279

[12] Dodin B, Elimam AA (1997) Audit scheduling with overlapping activities and sequence dependent setup costs. European Journal of Operational Research 97:22–33 [Erratum in European Journal of Operational Research 104 (1998) 262–264]

[13] Dodin B, Elimam AA, Rolland E (1998) Tabu search in audit scheduling. European Journal of Operational Research 106:373–392

[14] Drexl A (1991) Scheduling of project networks by job assignment. Management Science 37:1590–1602

[15] Fourer R, Gay DM, Kernighan BW (1993) AMPL – A Modelling Language for Mathematical Programming, The Scientific Press, San Francisco

[16] Gardner JC, Huefner RJ, Lofti V (1990) A multiperiod audit staff planning model using multiple objectives: development and evaluation. Decision Sciences 21:154–170

[17] Kolisch R, Sprecher A, Drexl A (1995) Characterization and generation of a general class of resource-constrained project scheduling problems. Management Science 41:1693–1703

[18] Neumann K, Morlock M (2002) Operations Research, 2nd edition. Hanser, München, Wien

[19] Neumann K, Schwindt C, Zimmermann J (2003) Project Scheduling With Time Windows and Scarce Resources, 2nd edition. Springer, Berlin

[20] Pritsker AAA, Watters WD, Wolfe PM (1969) Multiproject scheduling with limited resources: A zero-one programming approach. Management Science 41:93–108

[21] Rolland E, Patterson RA, Ward K, Dodin B (2005) Scheduling differentially-skilled staff to multiple projects with severe deadlines. European Journal of Operational Research (to appear)

[22] Salewski F (1995) Hierarchische Personaleinsatzplanung in Wirtschaftsprüfungsgesellschaften. Physica, Heidelberg

[23] Salewski F (1996) Tabu search algorithms for project scheduling under resource and mode identity constraints. IFORS 14th Triennial Meeting, Vancouver, July 9

[24] Salewski F, Drexl A (1993) Personaleinsatzplanung in Wirtschaftsprüfungsgesellschaften – Bestandsaufnahme und konzeptioneller Ansatz. Zeitschrift für Betriebswirtschaft 63:1357–1376

[25] Salewski F, Schirmer A, Drexl A (1997) Project scheduling under resource and mode identity constraints: Model, complexity, methods, and application. European Journal of Operational Research 102:88–110

[26] Vanderbeck F (2000) On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. Operations Research 48:111–128

Appendix A: Project Scheduling-Based Mathematical Program

To simplify the mathematical formulation of the problem, we perform some preliminary computations. First, the indices e, p, s are replaced by

$$j = g(e,p,s) = \sum_{e'=1}^{e-1} \sum_{p'=1}^{P_{e'}} S_{e'p'} + \sum_{p'=1}^{p-1} S_{ep'} + s \tag{15}$$

Thus, the parameters V_{eps}, $q_{epsp's'm}$, $\hat{q}_{epsp's'm}$ and k_{epsma} become V_j, $q_{jj'm}$, $\hat{q}_{jj'm}$ and k_{jma}. Then, let denote $f_e = g(e,1,1)$ the first and $l_e = g(e,P_e,S_{eP_e})$ the last subphase of each engagement e. Further, the maximum lags can be transformed into minimum lags (with a corresponding update of V_j) (cf. Bartusch et al. 1988), and from the time-lags, the release times, and the deadlines earliest and latest finish times EF_j and LF_j can be computed, respectively.

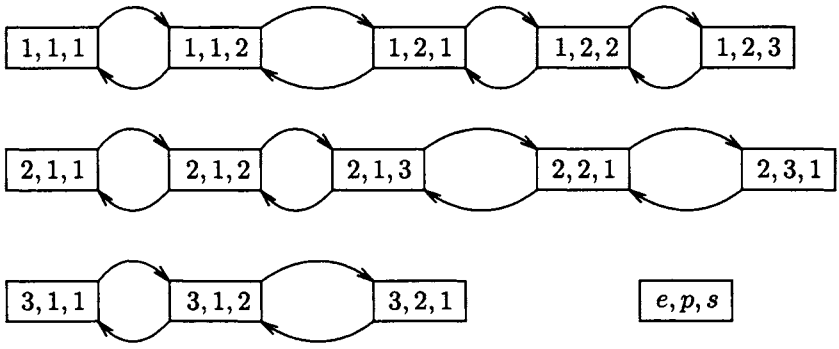


Figure 4: Chain Structure of Time Lags

Note that our problem setting covers not only the standard type of time-lags, namely minimum time-lags, but also the less common maximum time-lags. These can easily be converted into minimum time-lags using the transformation introduced in Bartusch at al. (1988) and Neumann et al. (2003). The presence of minimum and maximum time-lags between subsequent subphases, along with their special sequence, which arises from the above mentioned decomposition process, imply for each engagement a chain structure of the time lags as illustrated in Figure 4, where each node has the format $\boxed{e,p,s}$. Each structure may be seen as being composed of arc-disjoint cycles of length 2, one between first and second subphase, one between second and third one, and so forth up to the last cycle between last but one and last subphase. Finally, recall from Section 2 that precedence

relations exist only between subphases belonging to the same engagement, a characteristic denoted as 'isolating'.

Now, the assignment of subphases to modes and periods can be represented by binary variables $x_{jmt} = 1$, if subphase j is performed in mode m and completed in period t ($x_{jmt} = 0$, otherwise). This allows to formulate a binary program – using the general framework given in Pritsker et al. (1969) – by equations (16) to (22).

$$\max \sum_{e=1}^{E} \sum_{m=1}^{M_e} v^{em} \sum_{t=EF_{f_e}}^{LF_{f_e}} x_{f_e mt} \tag{16}$$

$$\text{s.t.} \sum_{m=1}^{M_e} \sum_{t=EF_{f_e}}^{LF_{f_e}} x_{f_e mt} = 1 \qquad e = 1, \ldots, E \tag{17}$$

$$\sum_{t=EF_{f_e}}^{LF_{f_e}} x_{f_e mt} - \sum_{t=EF_j}^{LF_j} x_{jmt} = 0 \qquad \begin{array}{l} e = 1, \ldots, E \\ j = f_e + 1, \ldots, l_e \\ m = 1, \ldots, M_e \end{array} \tag{18}$$

$$\sum_{m=1}^{M_e} \sum_{t=EF_{j'}}^{LF_{j'}} (t + q_{j'jm})\, x_{j'mt} - \sum_{m=1}^{M_e} \sum_{t=EF_j}^{LF_j} (t-1)\, x_{jmt} \leq 0 \qquad \begin{array}{l} e = 1, \ldots, E \\ j = f_e, \ldots, l_e \\ j' \in V_j \end{array} \tag{19}$$

$$\sum_{e=1}^{E} \sum_{m=1}^{M_e} \sum_{\substack{j=f_e \\ t \in \{EF_j, \ldots, LF_j\}}}^{l_e} k_{jma}\, x_{jmt} \leq C_{at} \qquad \begin{array}{l} a = 1, \ldots, A \\ t = 1, \ldots, T \end{array} \tag{20}$$

$$\sum_{m=1}^{M_e} \sum_{\substack{j=f_e \\ t \in \{EF_j, \ldots, LF_j\}}}^{l_e} k_{jma}\, x_{jmt} \leq D_{et} \qquad \begin{array}{l} e = 1, \ldots, E \\ a = 1, \ldots, A \\ t = 1, \ldots, T \end{array} \tag{21}$$

$$x_{jmt} \in \{0, 1\} \qquad \begin{array}{l} e = 1, \ldots, E \\ j = f_e, \ldots, l_e \\ m = 1, \ldots, M_e \\ t = EF_j, \ldots, LF_j \end{array} \tag{22}$$

The objective function (16) maximizes the total team preference over all engagements. Due to (17) it suffices to include only one subphase of each engagement in (16). The choice of the first one is arbitrary. The subphase completion constraints (17) stipulate that the first subphase of each engagement is completed exactly once in one of its modes. The mode identity constraints (18) guarantee for each engagement that if the first subphase is completed then the other subphases will be completed as well, and in the same mode as the first one. Thus, (17) and (18) combine to ensure that all subphases of each engagement will be processed in the same mode, and that all of them will be completed. The temporal constraints (19)

represent the precedence order on the subphases and enforce respection of the time-lags between them. The auditor capacity constraints (20) assure that for no auditor his per-period workload exceeds his capacity. In this regard, each auditor is treated as a renewable resource. The engagement capacity constraints (21) guarantee that for no engagement and no auditor the maximum per-period processing time is exceeded. In this regard, each engagement is treated as a renewable resource.

Table 6 summarizes the instances and, in addition, states the problem size in terms of the number of binary variables of the model formulation (16) to (22).

Table 6: Instance Characteristics

size	# weeks	# auditors	# engagements	#variables
very small	13	6	10	10,400
small	13	30	95	98,800
medium-size	26	55	280	728,000
large	52	125	880	5,948,800

Appendix B: Illustrative Example

Throughout this paper we make use of the instance provided in Tables 7, 8, 9, 5, 10, and 11, respectively. Note, the index l serves to interrelate the precendence relations with respect to Tables 5 and 10.

Table 7: Basic Parameters

$E = 4, T = 13, A = 6$
$M_1 = 5, M_2 = 3, M_3 = 8, M_4 = 2$
$P_1 = 2, P_2 = P_3 = P_4 = 1$
$S_{11} = 1, S_{12} = 5, S_{21} = 3, S_{31} = S_{41} = 1$
$C_{at} = 96, a = 1, \ldots, 6, t = 1, \ldots, 13$
$D_{et} = 40, e = 1, \ldots, 4, t = 1, \ldots, 13, D_{25} = 0$

Table 8: Preference Values v^{em}

e/m	1	2	3	4	5	6	7	8
1	4	5	9	2	1			
2	10	4	2					
3	7	6	6	5	2	5	2	1
4	5	7						

Table 9: Release Times λ_{ep} and Deadlines δ_{ep}

e/p	1	2	
1	1	7	λ_{ep}
2	3		
3	6		
4	5		
1	2	12	δ_{ep}
2	7		
3	8		
4	6		

Table 10: Minimal/Maximal Time Lags $q_{epsp's'm}/\hat{q}_{epsp's'm}$

l/m	1	2	3	4	5	e
1	5/13	0/13	0/13	1/13	1/13	1
2	0/0	-1/13	-1/13	0/0	0/0	
3	-1/13	-1/13	-1/13	0/0	0/0	
4	-1/13	-1/13	-1/13	0/0	0/0	
5	-1/13	-1/13	-1/13	0/1	0/0	
6	0/1	0/2	0/0			2
7	0/0	0/0	0/0			

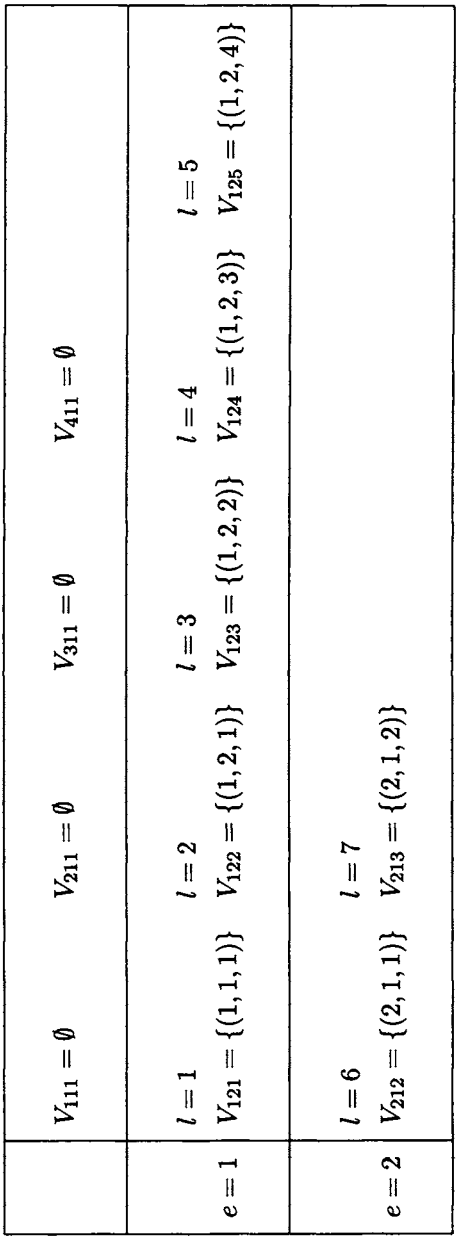


Figure 5: Predecessors V_{eps}

Table 11: Capacity Usages k_{epsma} (Missing Entries Are Zero)

a	1	2	3	4	5	6		
k_{1111a}							$e=1$	$m=1$
⋮	⋮	⋮	⋮	⋮	⋮	⋮		
k_{1251a}								
k_{1112a}	9	4	2	4	2			$m=2$
k_{1212a}		2	10	10	10			
k_{1222a}								
⋮	⋮	⋮	⋮	⋮	⋮	⋮		
k_{1252a}								
k_{1113a}		8	7	4	10	10		$m=3$
k_{1213a}		10	2	4	10	1		
k_{1223a}								
⋮	⋮	⋮	⋮	⋮	⋮	⋮		
k_{1253a}								
k_{1114a}	10							$m=4$
k_{1214a}								
k_{1224a}	4							
k_{1234a}	10							
k_{1244a}	10							
k_{1254a}	2							
k_{1115a}	4							$m=5$
k_{1215a}	8							
k_{1225a}	6							
k_{1235a}	4							
k_{1245a}	4							
k_{1255a}	10							
k_{2111a}							$e=2$	$m=1$
⋮	⋮	⋮	⋮	⋮	⋮	⋮		
k_{2131a}								
k_{2112a}		10		2		6		$m=2$
k_{2122a}		2	4	10		4		
k_{2132a}		6	2	10		6		
k_{2113a}			10	2	10	3		$m=3$
k_{2123a}			10	2	10	10		
k_{2133a}			8	4	8	6		
k_{3111a}							$e=3$	$m=1$
k_{3112a}	6	10		2		2		$m=2$
k_{3113a}		10	6			6		$m=3$
k_{3114a}				2	10	10		$m=4$
k_{3115a}		4			8	6		$m=5$
k_{3116a}	10	10						$m=6$
k_{3117a}			4	2				$m=7$
k_{3118a}	10			10	5	10		$m=8$
k_{4111a}							$e=4$	$m=1$
k_{4112a}		10						$m=2$

An MILP Modelling Approach for Shelf Life Integrated Planning and Scheduling in Scalded Sausage Production

Hans-Otto Günther
Dept. of Production Management
TU Berlin

Paul van Beek
Operations Research and Logistics Group
Wageningen University

Martin Grunow, Matthias Lütke Entrup, and Shuo Zhang
Dept. of Production Management
TU Berlin

1 Introduction

Due to factors such as the high variability of raw materials, intermediate and final products, fluctuating prices, and variable processing times and yields, production planning for sausages is generally a challenging task. One of the most distinctive factors to consider in fresh food production planning is shelf life. Shelf life restrictions directly influence wastage, out-of-stock rates and inventory levels. The shelf life of a product is defined as the time in which the food product will remain safe, be certain to retain the desired sensory, chemical, physical, and microbiological characteristics as well as comply with any label declaration of nutritional data (Kilcast and Subramanian, 2000, according to the London Institute of Food Science and Technology Guidelines, 1993). The possibility to offer a higher shelf life than its competitors constitutes a pivotal competitive advantage for fresh food producers, making the provision of shelf life functions crucial for modern production planning systems.

This research is based on the case study of scalded sausage production. This sub-segment of the processed meat industry has been chosen for two main reasons. On the one hand, scalded sausages constitute the most important product group in terms of production value (€ 2.7 bn. in 1999 in Germany, according to Lebensmittel Zeitung, 2001). On the other hand, production planning is particularly important for scalded sausages as they show the highest criticality with regard to shelf life. The results of the case study can easily be transferred to other product groups of the processed meat industry (fermented sausages, cooked sausages, ham etc.).

2 Case Study: Scalded Sausage Production

2.1 Product Characteristics of Sausages

Sausages are products in which "fresh comminuted meats are modified by various processing methods to yield desired organoleptic and keeping properties" (Savic, 1985). The production process dates back thousands of years to ancient Greeks and Romans, and even earlier. The term "sausage" is derived from the Latin word "salsus" meaning salt or salted (USDA et al., 1999). According to Pearson and Gillett (1996), consumers eat sausages for four reasons. The development of sausages was initially driven by economic factors such as utilizing lower-quality meats from cheaper cuts or edible by-products (Xiong and Mikel, 2001). Convenience is the second reason for the success of sausages. Sausages take only little time in preparation. Thirdly, the great and still increasing variety of sausage products in terms of flavours, textures and shapes makes it possible to serve many different variants. Finally, sausages are also of good nutritional value since most of them are excellent sources for high quality protein, essential minerals, and all B vitamins (Kim, 2001).

Sausages can be categorized into three major segments (Savic, 1985; Neuhäuser, 1996; Pearson and Gillett, 1996):

- Scalded Sausages comprise ready-to-eat products made from comminuted and well-homogenized cured meats, fatty tissue, water and seasonings usually smoked and scalded. Examples are Frankfurters, Bologna (Mortadella), Knackwurst, or Wieners.
- Fermented Sausages are not heat processed and are made from cured or uncured, fermented and often smoked meats. This category can be divided into dry and semi-dry sausages and includes for instance Salami, Pepperoni, Summer Sausages (Cervelat Sausage), or Braunschweiger (Sielaff, 1996).
- Cooked Sausages are ready-to-serve products usually made from previously cooked fresh or exceptionally cured raw materials, subjected to final cooking after stuffing. Oelker (1996) names the three sub segments Liver Sausages, Blood Sausages and Brawn Sausages.

In 1999, over 56% of all processed meat in Germany were sausages (Lebensmittel Zeitung, 2001). Among the sausages, scalded sausages constitute the most important sub-segment (over 50% in 1999). Therefore, this research is particularly concerned with the segment of scalded sausages and specifically with the production of larger diameter, sliced scalded sausages which allows to integrate the capital-intensive slicing and packaging equipment in the case study.

2.2 Supply Chain of Scalded Sausages

The scalded sausages supply chain usually consists of four stages. The grower produces the living animals which are slaughtered and dressed by slaughterhouses. The raw meat is then transported to the sausage producers. After having transformed the raw meat into sausages, the sausage products are distributed and sold via retailers.

The sausage supply chain has been subject to major changes throughout the recent years. With regard to economic developments, a clear trend towards a consolidation can be observed on all stages in the supply chain. At the manufacturer's level, the meat processing industry is still characterized by mid-sized companies (Wünsche, 2002). As economies of scale are substantial (Connor and Wills, 1988) concentration is likely to accelerate (KPMG Corporate Finance UK, 2000; Bourlakis and Weightman, 2004). In particular foreign meat processors are increasingly buying domestic producers. Major concerns for sausage producers are also the overcapacities in the market. Auer (2001) estimates the overcapacities in the German meat industry to be around 30-40% resulting in a tough competition.

The continued consolidation of retailers is perceived as the biggest threat to the manufacturers (KPMG Corporate Finance UK, 2000). In 2000, the five leading retailers in Germany made 63% of the total retail turnover, 18%-points more than ten years ago (Michael et al., 2002). In the UK, the share of the top five retailers has already achieved more than 80% (van Wezel, 2001). The strong retail concentration leads to a significant shift of power in the fresh food supply chain. Retailers will increasingly command the retailer-manufacturer interface (McLaughlin, 2002) or even the whole upstream part of the chain with severe consequences to the manufacturers. Retailers put the manufacturer's prices under pressure; they are now in a position to even dictate the business terms. Another characteristic of the retail environment is the raise of the discount channel and private labels, leading to a further segmentation of the market. While manufacturers can still obtain relatively high margins in the premium segment, the manufacturer's margins for private labels are much lower.

Therefore, differentiation gets increasingly important for manufacturers in order to avoid cost and margin pressure. On the product level, new product introductions increased considerably. The number of product innovations introduced for example in German retail has increased since 1997 by 11.1% annually; however, the failure rate reached almost 66% in 2000 (Michael et al., 2002). Furthermore, this product proliferation leads to production complexity. In contrast to the US, promotional activities are another important differentiating factor in Europe. Particularly in Germany, significant revenue shares are generated by promotions (Treeck and Seishoff, 2002). Most promotions are price discounts; other types include e.g. loyalty rewards, special events, displays,

samples, or contests (Seifert, 2001). Several studies reveal that most promotions are not effective and successful. According to Fairfield (2002), manufacturers estimate that only 35% of all promotions are profitable.

Product freshness is one of the few remaining differentiating factors for manufacturers and retailers. Freshness is one of the most important quality characteristics of a fresh food product for consumers (N.N., 2003). Most food processors and retailers expect that the share of fresh and perishable products in retail will continue to grow (Grievink et al., 2002).

2.3 Production Process of Scalded Sausages

The production process of scalded sausages is characterized by eight major steps, ranging from the input of ingredients to the storage and delivery of the final products.

1. Input of ingredients (raw meat, salt, water/extenders and seasonings).
2. Grinding and mixing of the raw meat.
3. Chopping and emulsifying of the meat particles.
4. Stuffing and tying of the sausage dough into casings and loading onto trolleys for transfer to the smokehouse.
5. Scalding of the intermediate sausages in special scalding chambers.
6. Maturing and intermediate storage.
7. Slicing and packaging.
8. Storage on pallets and delivery to a retail distribution centre or directly to the retail outlets.

3 Literature Review

3.1 Lot Sizing and Scheduling in Make-and-Pack Production

In literature, a production environment which is characterized by a single production stage and a subsequent packaging stage is named "make and pack production" (Günther and Neuhaus, 2004; Méndez and Cerdá, 2002). Major issues of operational production planning in this environment are lot sizing and scheduling, which can be performed in one single or two separate planning steps. As lot sizing usually aims at balancing set-up costs on the one hand and inventory holding costs on the other hand, the set-up costs of each single set-up operation must be known in advance. However, as set-up times and costs in scalded sausage production are sequence-dependent, lot sizing and sequencing must be integrated (Sikora et al., 1996).

The available approaches can be classified according to their representation of time; a discrete and a continuous time representation can be distinguished.

For a *discrete* representation of time, the planning horizon is divided into a certain number of periods that have usually the same length. Examples of problems and related modelling approaches are the Capacitated Lot size Problem (Günther, 1987), the Discrete Lot-sizing and Scheduling Problem (Fleischmann, 1990), the Continuous Setup and Lot-sizing Problem (Karmakar and Schrage, 1985), the Proportional Lot-sizing and Scheduling Problem (Haase, 1994), and the Capacitated Lot-sizing Problem with Sequence-dependent Setup Costs (Haase, 1996). Fleischmann and Meyr (1997) integrate all mentioned models within the General Lot-sizing and Scheduling Problem. All models have in common that set-up times can only be considered if they do not exceed the length of a period. However, Koçlar and Süral (2005) show that set-up times exceeding the length of a period can be incorporated.

Nevertheless, choosing the length of a period becomes a crucial aspect of modelling. Especially because a high number of relatively small periods are required for an exact representation of the production activities (Meyr, 1999), the number of periods can considerably increase the size of the model (Günther and Neuhaus, 2004; Méndez and Cerdá, 2002). Stadtler (2005) emphasizes that particularly sequence-dependent set-up times cannot be represented properly within a model using large time periods. In addition, the approximation or "rounding" of processing and set-up times to fit them into the fixed periods can lead to overproduction, idle times, or infeasibility (Ierapetritou and Floudas, 1998; Günther and Neuhaus, 2004). Due to the high complexity of these models, they are less suitable to practical purposes (Meyr, 1999; Burkard et al., 2002). The application of time periods of different length can be useful to avoid the stated problems. Burkard et al. (2002) introduce the notion of the "Event-Driven Model", in which only such points in time are considered, at which a process is allowed to start.

Alternatively, it is possible to use a *continuous* representation of time, which allows scheduling the start and end of an activity precisely on a continuous time scale. In particular, infeasibilities due to "fitting" set-up times into a discrete time grid can be avoided (Ierapetritou and Floudas, 1998; Günther and Neuhaus, 2004). Sahinidis and Grossmann (1991) propose a model that uses a continuous representation of time and considers explicitly sequence-dependent set-up times. They apply a position-based model and assume constant demand patterns to generate a cyclic schedule. Among the latest publications, the approach of Ierapetritou and Floudas (1998) is to be mentioned that considers sequence-dependent set-up times for both batch and continuous production. Méndez and Cerdá (2002) suggest a similar model formulation that is characterized by a lower number of binary variables and hence is computationally more efficient. Nevertheless, none of the mentioned models supports lot sizing and it is not

possible to integrate demand data of a single final product for every single production day, only for the aggregate demand of the entire planning horizon.

Günther and Neuhaus (2004) present an approach that is based on the principle of block planning and that simultaneously considers lot sizing and scheduling. By integrating several variants of a product type or recipe into a "block", the complexity of the model is significantly reduced. 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 colour to the darker. In the concept of flexible block planning presented by Günther and Neuhaus (2004), the blocks are assigned to a macro-period. According to this concept, the completion of a block must take place before the end of a macro-period. However, as the start of a block can be in the same or a previous period, a production lot cannot be assigned to a specific day, which is critical in order to consider its shelf life.

3.2 Production Planning for Perishable Products

Most models assume an unlimited storage of intermediate and finished products. However, fresh products have a finite shelf life that has to be respected in production planning. Many authors concede the necessity to integrate the shelf life of products in production planning and scheduling (e.g. Kallrath, 2002; Günther and Neuhaus, 2004); nonetheless shelf life has only been considered explicitly in very few models, e.g. Lütke Entrup et al. (2005) for the case of yogurt production. Within the available approaches, two main streams can be distinguished.

On the one hand, a vast body of literature exists on *inventory management* for perishable products. Besides perishable food products, perishable inventory theory covers also the behaviour of radioactive materials, photographic film, prescription drugs, or blood conserves. Nahmias (1982) and Raafat (1991) give a comprehensive literature overview and an analysis of proposed inventory models for perishables. Raafat particularly clarifies the difference between continuously deteriorating goods and products that can be unrestrictedly used before the expiry date. However, sausage products show characteristics of both. On the one hand, the value of sausages decreases over time as customers give a higher value to a fresh product; on the other hand, sausages are almost worthless after the expiry date. Abad (2003) developed a non-linear single-product model for the maximization of the contribution margin while considering the decay of the products in order to determine the best-possible sales price.

On the other hand, with regard to the integration of shelf life into *production planning and scheduling*, most approaches introduce a shelf life constraint into the Economic Lot Scheduling Problem (ELSP), which is concerned with generating a cyclic schedule for several products, based on a single resource and con-

stant demand rate (Elmaghraby, 1978; Cooke et al., 2004). Soman et al. (2004) provide a review of the major contributions.

Almost all mentioned ELSP models include assumptions that are seldom present in an industrial environment. First of all, a constant demand rate is frequently supposed, which is not very realistic for fresh food industries with seasonalities and intense promotional activities. Manna and Chaudhuri (2001) underline, for example, that the demand for deteriorating products may be time-, stock-, or price-dependent (e.g. end-of-day pricing). Moreover, most models consider only one single facility and do not account for sequence-dependent set-up times. Finally, the most important criticism is that ELSP models aim at generating a production cycle which is repeated in certain intervals and which must not exceed the shortest product shelf life. Hence, product freshness is not considered in the objective function, only as a constraint.

4 Profile of Scalded Sausage Production Regarding APS Systems

4.1 Problem Demarcation and Model Overview

The modelling approach suggested relies on a MILP model formulation. It includes binary decision variables for the set up of production lots on a specific day. The model comprises the production steps from scalding to storage and delivery, with special attention being paid to the scalding and slicing and packaging steps (see Figure 1).

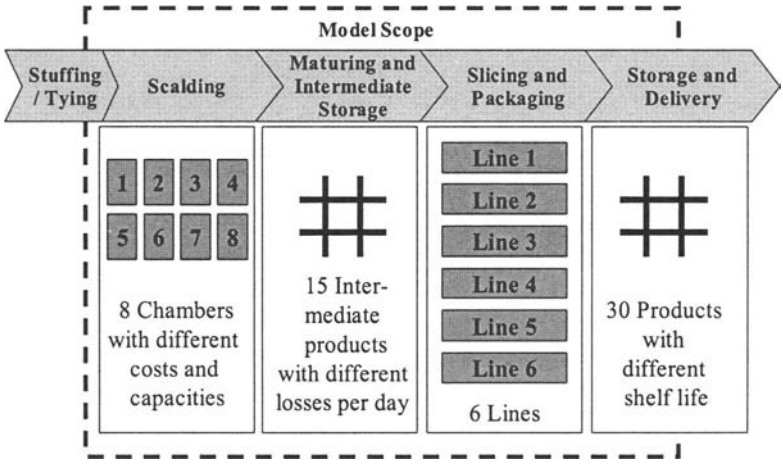


Figure 1: Scope of the production model

All prior production steps from ingredients input to stuffing and tying have been neglected for two reasons. First, the scalding step determines the shelf life of the final sausage product. In order to improve the freshness of the final product, it is sufficient to consider this step and the following processes. With respect to the meat raw materials, it must only be assured that they are processed within their shelf life period, which is different and much shorter than that of the final products. Secondly, scalding as well as slicing and packaging are typical bottlenecks of the entire production process. Therefore, the model will focus on ensuring a high utilization of these units.

In the MILP model eight scalding chambers are considered with different costs per hour and different capacities. After having brought the filled casings into the scalding chamber, they are subject to several processes that include drying, smoking, scalding and cooling. All these activities are fully automated and take place inside the chamber. The time required for this program is fixed and differs by the type of sausage produced. However, if two intermediate sausages are based on the same program, they can be put together into the chamber. Set-up and cleaning times are only of minor importance and are part of the scalding program.

Mainly due to the evaporation of water, important time dependent losses occur in the intermediate storage. The losses must be considered in the model to obtain correct sausage volumes for the packaging step. Furthermore, as most products are sold on a weight-basis, reducing the storage times of the intermediate products helps to increase the volume available for sale. Although it is desirable to keep the stock levels in the intermediate storage as low as possible, a minimum time is imposed to avoid a later sticking of the slices at the packaging stage that would lead to a lower slicing quality.

Considering the high number of products in the packaging step, a block planning approach is chosen in order to guarantee the compactness and computability of the model. For the sequence of the products within a block a "natural order" exists which is mainly based on microbiological concerns (weaker microbiological contamination before the stronger). When changing the production between two products that are not part of the same block, it is always necessary to perform a set-up operation. Only when changing the production between two product variants of the same recipe, the cleaning and sterilizing of the production facilities can be neglected. Hence, not only the sequence of products within a block is fixed but also the sequence of blocks can be fixed within the day. In that case, the different blocks are enumerated according to their position within the day. Products that can be processed within the same block do not need to be based on the same intermediate product, since besides the raw material it is also the diameter of the intermediate product that determines the assignment of a

product to a block. Hence, an additional index is used in order to enumerate the different blocks.

The developed models are based on a continuous representation of time. For the consideration of shelf life, it is necessary to employ a discrete, uniform time grid in addition (see Figure 2; Günther and Neuhaus, 2004). A period refers to a day as shelf life is usually given in days. In most lot-sizing/scheduling models so-called balance constraints are used to link inventory at the end of each period to its starting inventory, the production and demand in that period. To properly handle shelf life considerations, production lots are assigned to specific demand dates. Hence, no inventory balance equations are needed. At the packaging stage, intense cleaning which takes several hours is required once a day. This cleaning, which usually takes place during the night, clearly separates two days so that a conservation of the set-up state over period boundaries is not necessary.

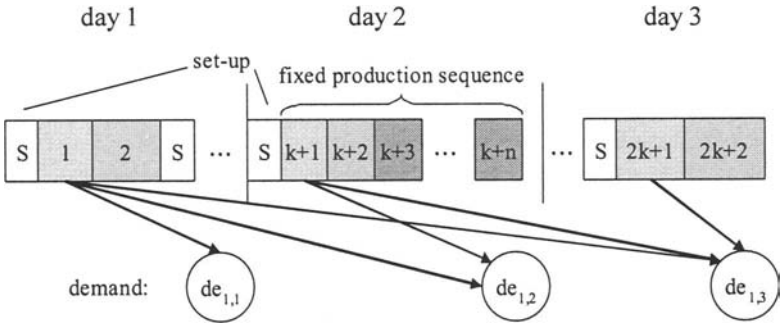


Figure 2: Block planning approach (based on Günther and Neuhaus, 2004)

The objective function aims at maximizing the contribution margin. Shelf life is taken into account by a shelf life dependent pricing component. The cost elements considered in the objective function include the variable cost of the product, the variable costs of the packaging lines and the scalding chambers, and the variable cost of the intermediate products (particularly raw meat).

The planning horizon covers one production week from Monday 00:00 to Friday 24:00 (days 7 to 11). In the packaging department, the Sunday before (day 6) and the Saturday after (day 12) the production week can be used as overtime. The demand of final products is based on forecasts and on already arrived customer orders and comprises the actual production week (days 7-11) and Monday, Tuesday and Wednesday of the following week (days 14-16). Furthermore, two types of stock levels with their corresponding shelf life are taken into account. The stock level of intermediate products at the beginning of the planning period (day 6, Sunday 00:00) contains intermediate products that have been produced on Tuesday, Wednesday, Thursday and Friday of the previous week

(days 1 to 4). The stock level of finished products comprises only products that are based on intermediates of Tuesday, Wednesday and Thursday of the previous week (days 1-3). Due to the required minimum time in the intermediate storage, the intermediate products that have been produced on Friday of the previous week (day 4) cannot be packed on the same day. Hence, at the beginning of the planning week there are no final products in the warehouse that are based on intermediate products that have been produced on Friday (day 4).

In order to be able to start packaging on Monday of the following week (day 14), the corresponding volume of intermediate products has to be produced within the planning horizon in addition. This stock level of intermediate products at the end of the production week is assumed to be the same as the stock level at the beginning of the production week in terms of both volume and age. The stock level of final products at the end of the production week does not need to be determined in advance as it can be derived from the model results.

4.2 Model Formulation

Indices

$i \in I$	intermediate products
$j \in J$	final products
$o \in O$	chambers
$l \in L$	packaging lines
$s, so \in S$	production days (= shelf life days)
$p \in P$	packaging days
$d \in D$	demand days
$k \in K$	positions
$g \in G$	scalding programs
$b \in B$	blocks
$I \in I(g)$	intermediate products based on scalding program g
$j \in JB(b)$	final products based on block b
$j \in JI(i)$	final products based on intermediate product i
$l \in LJ(j)$	packaging lines that can process final product j
$l \in LB(b)$	packaging lines that can process block b

Parameters

ic_i	cost of intermediate product i in € per kg
ou_i	factor indicating the chamber utilization of intermediate product i
$ocap_o$	capacity of chamber o in kg
oc_o	cost of chamber o in € per day
imb	minimum batch size for a scalding program in kg
$scaltime_g$	processing time of scalding program g

$weightloss_i$	loss of weight of intermediate product i per day in the intermediate warehouse in % of the initial weight
$estart_k$	earliest possible starting time of a block at position k
is_{is}	inventory of intermediate product i produced on day s in kg
id_{is}	demand of intermediate product i to be produced on day s (to start packaging at the beginning of the following week)
$iscap$	capacity of inventory for intermediate products
fp	final position
$varc_j$	variable cost of final product j in € pro kg
rev_j	revenue for selling one kg of final product j in €
sl_j	maximum shelf life of final product j in days
cr_j	minimum shelf life of product j required by the customer (as a fraction of maximum shelf life, applied to multiply the shelf life of product j)
lu_j	factor indicating the packaging line utilization of final product j
cap_l	capacity of packaging line l in kg per day
c_l	cost of packaging line l in € per day
st_b	set-up time for block b in days
ct	daily cleaning time for packaging lines
ben_j	maximum additional benefit when meeting the maximum shelf life of final product j, in € per kg
q_j	maturation time for final product j in days
mb	minimum batch size of a block in kg
de_{jd}	demand of final product j of demand day d in kg
$fixdem$	demand day up to which the demand must be fully satisfied
s_{js}	inventory of final product j produced on day s in kg
os_l	overtime supplement for weekend production on packaging line l
$ploss_j$	slicing loss of final product j in % of sliced volume
rq_j	share of slicing losses of final product j in % that can be reintegrated into the production process
ldp	start of the last packaging day within the planning week (Saturday)
fdp	start of the first packaging day within the planning week (Sunday)

Decision Variables

T_{gko}	=1, if scalding program g is set up at position k in chamber o (0, otherwise)
S_{pbl}	=1, if block b is set up on packaging day p on packaging line l (0, otherwise)
V_{iko}	volume of intermediate product i at position k in chamber o in kg
W_{is}	volume of intermediate product i produced on day s in kg
$START_{ko}$	start of production at position k in chamber o

X_{jpsl}	volume of final product j produced on production day s and packed on packaging day p on packaging line l in kg
Z_{jpsd}	volume of final product j produced on production day s and packed on packaging day p in kg that is used to meet the demand of demand day d
Y_{jsd}	inventory of final product j produced on production day s in kg that is used to meet the demand of demand day d
L_{pbl}	duration of block b on packaging day p on packaging line l
END_{pbl}	end time of block b on packaging day p on packaging line l
EST_{pl}	start time of packaging line l on packaging day p
LFT_{pl}	end time of packaging line l on packaging day p
SUO_l	overtime at the beginning of the planning week (Sunday) on packaging line l
SAO_l	overtime at the end of the planning week (Saturday) on packaging line l

Objective Function

$$\begin{aligned}
&\max \sum_{j\in J}\sum_{s\in S}\sum_{d\in D}\Big(\sum_{p\in P} Z_{jpsd} + Y_{jsd}\Big)\cdot\Big(ben_j \cdot \frac{\big((1-cr_j)\cdot sl_j\big)-(d-s)}{(1-cr_j)\cdot sl_j} + rev_j\Big) \\
&+ \sum_{p\in P}\sum_{s\in Sl\in L}\sum_{i\in I}\sum_{j\in JI(i)} X_{jpsl}\cdot ploss_j \cdot rq_j \cdot ic_i \\
&- \sum_{i\in I}\sum_{k\in K}\sum_{o\in O} V_{iko}\cdot ic_i - \sum_{g\in G}\sum_{k\in K}\sum_{o\in O} T_{gko}\cdot oc_o \cdot scaltime_g \\
&- \sum_{j\in J}\sum_{p\in P}\sum_{s\in Sl\in L} X_{jpsl}\cdot(1-ploss_j)\cdot varc_j \\
&- \sum_{l\in L}\sum_{p\in P}\big(LFT_{pl}-EST_{pl}\big)\cdot c_l - \sum_{l\in L}\big(SAO_l + SUO_l\big)\cdot os_l
\end{aligned} \tag{1}$$

The objective function aims at maximizing the contribution margin. Regarding revenues, it considers the regular revenues of the sold products and a shelf life dependent pricing component. It is supposed that the manufacturer yields a financial benefit if the products have a longer residual shelf life when being delivered. The shelf life dependent benefit increases linearly between the minimum customer requirement on shelf life (cr_j*sl_j) and the maximum possible shelf life (sl_j) since the benefits for the retailer increase with every additional day of residual shelf life (see Figure 3).

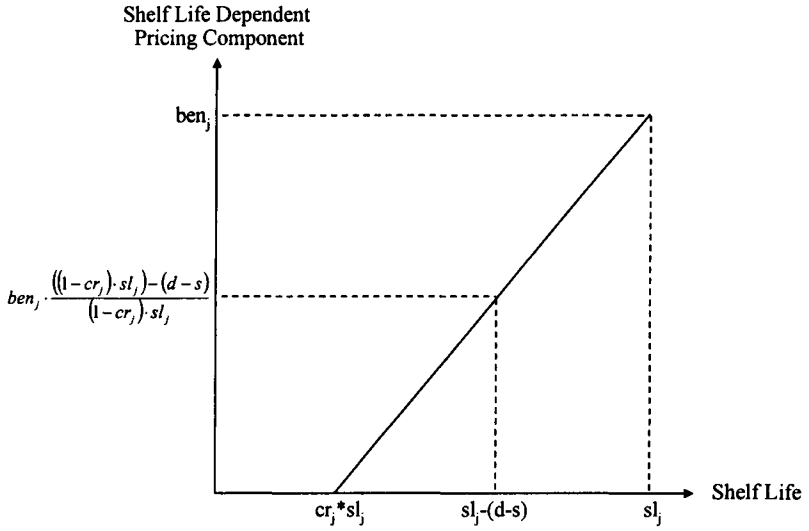


Figure 3: Shelf life and shelf life dependent pricing component

As an example, suppose that product j has a total shelf life of 30 days (sl_j = 30) and that the customers require a minimum residual shelf life when being delivered of 66% of the total shelf life (cr_j = 66%). Suppose further that the shelf life of the product starts on day 7 (s = 7), the product is delivered to the retailer on day 11 (d = 11) and the maximum benefit for meeting the maximum shelf life of product j, ben_j is € 0.30 per kg. In this case, the manufacturer yields a financial benefit of € 0.18 per kg of product j (60% of the maximum benefit).

Moreover, as slicing losses can partly be reintegrated into the process, they must be respected in addition ($X_{jpsl} * ploss_j * rq_j * ic_i$). The considered cost elements of the smoking and scalding step include the variable costs for the raw meat and for the ingredients of the intermediate products (ic_i) as well as the processing costs of the scalding chambers (oc_o). In packaging, the variable costs of packaging are taken into consideration. These costs do not occur for the slicing losses ($ploss_j$). Furthermore, the costs of utilizing the packaging lines in regular and overtime mode are considered. Fixed cost elements have been neglected, as there are not relevant on a weekly planning level. Constraints to be respected are the following.

Set-up and capacity of chambers

$$\sum_{i \in I(g)} (V_{iko} / ou_i) \leq T_{gko} \cdot ocap_o \qquad k \in K;\ g \in G;\ o \in O \qquad (2)$$

$$\sum_{g \in G} T_{gko} \leq 1 \qquad k \in K;\ o \in O \qquad (3)$$

According to (2), intermediate product i can only be produced in chamber o if the chamber is set up ($T_{gko} = 1$) for the corresponding scalding program g. In addition, the volume of all intermediate products i that are based on the same scalding program g ($i \in I(g)$) and that are produced at the same position k in chamber o must not exceed the available capacity of the chamber ($ocap_o$). Since smaller diameter sausages cannot fill the chamber to the same extent as large-calibre sausages, the fill grade of the chamber by different types of intermediate products is taken into account by the factor ou_i. Moreover, a position k of scalding chamber o can only be used to start one scalding program (3).

Sequencing scalding

$$START_{ko} \geq START_{k-1,o} + scaltime_g \cdot T_{g,k-1,o} \qquad k \in K:\ k>1;\ o \in O;\ g \in G \qquad (4)$$

This constraint ensures that scalding program g may not start at position k before the end of its predecessor k-1 in order to avoid overlapping of scalding programs in a chamber.

Day bounds scalding

$$estart_k \leq START_{ko} < estart_k + 1 \qquad k \in K;\ o \in O \qquad (5)$$

$$START_{ko} + scaltime_g \cdot T_{gko} \leq estart_k + 1 \qquad k \in K:\ k=fp;\ o \in O \qquad (6)$$

For the consideration of shelf life, each position k is assigned to a specific production day by the parameter $estart_k$. Accordingly, the start of a scalding program at position k must take place on the production day the position is assigned to (5). Since intermediate products can only be produced from Monday to Friday, the last position of the planning horizon ($k = fp$) must be finished at Friday midnight ($= estart_{fp} + 1$) at the latest (6).

Production volume scalding

$$W_{is} \leq \sum_{o \in O} \sum_{k \in K:\, estart_k = s} V_{iko} \qquad s \in S;\ i \in I \qquad (7)$$

The produced volume of intermediate product i on production day s (W_{is}) must not exceed the sum of all volumes of that intermediate product i that have been produced at any position that is assigned to this production day s ($estart_k = s$) and in any scalding chamber o.

Warehouse capacity for intermediate products

$$\sum_{i\in I}\sum_{s\in S:so\le s}(W_{is}+is_{is})/ou_i-\sum_{p\in P:p\le s}\sum_{s\in Sl}\sum_{l\in Li}\sum_{j\in JI(i)}\frac{X_{jpsl}}{\left[\left(1-weightloss_i\cdot(p-s)\right]\cdot ou_i\right)}$$

$$\le iscap \qquad s\in S \qquad (8)$$

The warehouse capacity (*iscap*) for intermediate products must not be exceeded on any production day s (8). The inventory level on production day s is determined by first summing up the inventories of the previous week (is_{is}) and the produced volumes of intermediate products of the previous days of the current planning week ($so \le s$). Then, the volume of intermediate products that has already been used for packaging is deducted (X_{jpsl}, for all packaging days $p \le s$). The capacity utilization of different intermediate products i is respected by the factor ou_i. The weight loss of intermediate products in the warehouse ($weightloss_i$) is also taken into account. Although (8) is only valid at the end of each day, it can nevertheless adequately represent the course of a day as all inventory inputs and outputs are relatively equally distributed over the day.

Minimum batch size scalding

$$\sum_{i\in I(g)}V_{iko}\ge imb\cdot T_{gko} \qquad k\in K;\ o\in O \qquad (9)$$

A minimum filling level of the chambers is ensured by (9). If a scalding program g is set up at position k in chamber o ($T_{gko} > 0$), the volumes of the intermediate products being part of the production batch must meet or exceed the minimum batch size (*imb*).

Weight losses in intermediate storage and meeting packaging demand

$$(W_{is}+is_{is})\ge\sum_{j\in JI(i)}\sum_{l\in L}\sum_{p\in P}\frac{X_{jpsl}}{\left[1-weightloss_i\cdot(p-s)\right]+id_{is}} \qquad s\in S;\ i\in I \qquad (10)$$

This constraint constitutes the link between the scalding and the packaging part. For all intermediate products i of production day s, the produced quantities (W_{is}) and the inventory from the previous week (is_{is}) must cover the demand of the packaging department (X_{jpsl}) and the demand of intermediate products for the start of packaging at the beginning of the following week (id_{is}). The weight loss of the intermediate products in the warehouse must also be considered in order to provide the accurate quantities of intermediate products for the packaging department. The weight loss that is mainly due to water evaporation is given in % per day in the warehouse ($weightloss_i$) and multiplied by the days passed between scalding and packaging (p-s). Although in reality, the weight loss follows a more exponential function, a linear modelling approach for the weight losses has been chosen for two reasons. On the one hand, the solvability of the

model is increased by choosing a linear approach. On the other hand, scalded sausages stay usually only a couple of days in the warehouse. The losses during these first days can relatively well be reflected by a linear function as the water loss rate remains relatively stable during these first days. However, in case of fermented sausages (e.g. salami) with longer maturing times (up to weeks and months), a more detailed modelling approach would be required.

Set-up packaging

$$\sum_{j \in JB(b)} \sum_{s \in S} X_{jpsl} / lu_j \leq S_{pbl} \cdot cap_l \qquad p \in P;\ b \in B;\ l \in LB(b) \quad (11)$$

The final product j can only be packed on packaging day p on packaging line l if the corresponding block b of the final product $j \in JB(b)$ is set up on that line and on that day.

Output quantities packaging

$$S_{pbl} \cdot st_b + \sum_{j \in JB(b)} \sum_{s \in S} \frac{X_{jpsl} \cdot (1 - ploss_j)}{cap_l \cdot lu_j} \leq L_{pbl} \qquad p \in P;\ b \in B;\ l \in LB(b) \quad (12)$$

The duration of block b on packaging line l on packaging day p (L_{pbl}) includes the set-up time of the block b (st_b) and the actual production times of all final products $j \in JB(b)$ that are produced within the block b. As for intermediate products in the scalding chambers, the final products also differ with regard to their degree of utilization of the capacity of the packaging lines. This fact is considered by the factor lu_j. If packaging line l is not set up for the block b ($S_{pbl} = 0$), the output quantities of all final products of this block become zero ($X_{jpsl} = 0$) according to (11). Consequently, the duration of this block (L_{pbl}) becomes zero as well so that the start time and the end time of this block are the same. Since the capacities of the packaging lines (cap_l) are given in terms of output-kg, the slicing loss ($ploss_j$) must be subtracted from the quantities given into the process (X_{jpsl}). The slicing loss is partly re-integrated in the production process which is respected in the objective function (1).

Sequencing packaging

$$END_{pbl} \geq L_{pbl} + END_{p,b-1,l} \qquad p \in P;\ b \in B:\ b > 1;\ l \in LB(b) \quad (13)$$

This constraint guarantees that an overlapping of blocks on a packaging line l cannot take place since block b may not start before the end of its predecessor b-1. It is supposed that not only the sequence of final products within a block is fixed, but also the sequence of blocks within a day.

Day bounds packaging

$$END_{pbl} \leq p+1 \qquad b \in B;\ p \in P;\ l \in LB(b) \quad (14)$$

$$END_{pbl} - L_{pbl} \geq p + ct \qquad b \in B;\ p \in P;\ l \in LB(b) \quad (15)$$

These constraints assign each block to a specific packaging day p. A block b on packaging line l on packaging day p must be finished before the end of production day p. Furthermore, the beginning of this block (END_{pbl} - L_{pbl}) must take place after the beginning of the corresponding packaging day p. In addition, a general cleaning time (ct) must be respected for all lines, which takes place every day at midnight. Hence, the feasible interval for the float variables indicating the end of a block is set by the interval derived from the integer day numbers. For instance, according to (14) and (15), any block b packed on day p=8, must be completed by 9 and started after 8. Thus, END_{pbl} will assume a value between 8.00 + L_{pbl} + ct and 9.00.

Stock balance final products

$$\sum_{l \in L} X_{jpsl} \cdot (1 - ploss_j) \geq \sum_{d \in D} Z_{jpsd} \qquad j \in J;\ s \in S;\ p \in P \quad (16)$$

$$s_{js} \geq \sum_{d \in D} Y_{jsd} \qquad j \in J;\ s \in S \quad (17)$$

For the coverage of the demand (22), final products from stock (s_{js}), produced in the previous week, or final products that have been produced within the actual planning week (X_{jpsl}) can be taken into account. Due to the fact that the slicing of the intermediate products takes place within the packaging step, the resulting slicing losses ($ploss_j$) must be subtracted in order to provide accurate quantities. The slicing losses can partly be reintegrated into the production of intermediate products (1).

Utilization of packaging lines

$$LFT_{pl} \geq END_{pbl} \qquad b \in B;\ p \in P;\ l \in LB(b) \quad (18)$$

$$EST_{pl} \leq END_{pbl} - L_{pbl} \qquad b \in B;\ p \in P;\ l \in LB(b) \quad (19)$$

$$SUO_l \geq LFT_{pl} - EST_{pl} \qquad l \in L;\ p = fdp \quad (20)$$

$$SAO_l \geq LFT_{pl} - EST_{pl} \qquad l \in L;\ p = ldp \quad (21)$$

The variables LFT_{pl} (18) and EST_{pl} (19) determine the finishing time and the start time of packaging line l on packaging day p. The difference (LFT_{pl} - EST_{pl}) describes the actual run time of the packaging line l on that packaging day p which is used in the objective function (1) in order to determine the cost of the line utilization. The variables SUO_l and SAO_l in (20) and (21) determine the

overtime required on packaging line l at the beginning (Sunday) and at the end (Saturday) of the planning period.

Meeting demand

$$de_{jd} = \sum_{p \in P} \sum_{\substack{s \in S: \\ (d-s) \le (1-cr_j) \cdot sl_j}} Z_{jpsd} + \sum_{\substack{s \in S: \\ (d-s) \le (1-cr_j) \cdot sl_j}} Y_{jsd} \qquad j \in J;\ d \in D:\ d \le fixdem \quad (22)$$

$$de_{jd} \ge \sum_{p \in P} \sum_{\substack{s \in S: \\ (d-s) \le (1-cr_j) \cdot sl_j}} Z_{jpsd} + \sum_{\substack{s \in S: \\ (d-s) \le (1-cr_j) \cdot sl_j}} Y_{jsd} \qquad j \in J;\ d \in D:\ d > fixdem \quad (23)$$

The demand of final product j of demand day d can be covered by volumes produced within planning week (Z_{jpsd}) or from stock (Y_{jsd}). The customer requirements regarding the minimum residual shelf life (cr_j) have to be respected for both products from stock and products produced within the planning week. For all days up to day *fixdem*, full satisfaction of demand is required, reflecting the very competitive retail environment. *fixdem* is usually set to Saturday of the planning week. For the remaining days (especially the demand of the following week), the satisfaction of the demand is optional. In urgent cases, these products can be sliced and packed on Sunday of the following planning week.

Minimum batch size packaging

$$\sum_{j \in JB(b)} \sum_{s \in S} X_{jpsl} \cdot (1 - ploss_j) \ge mb \cdot S_{pbl} \qquad b \in B;\ p \in P;\ l \in LB(b) \quad (24)$$

Similar to the scalding chambers, a minimum batch size for block b for every packaging day p and on every packaging line l is required in order to justify the set-up times. A block consists of all production volumes of all final products of this block ($j \in JB(b)$).

Maturation time for intermediate products

$$X_{jpsl} \le 0 \qquad j \in J;\ p \in P;\ s \in S:\ s+q_j \ge p;\ l \in LJ(j) \quad (25)$$

A maturation time for intermediate products in the warehouse (q_j) is considered. Products cannot be packed if the maturation time has not passed. Although the maturation time concerns intermediate products, the time is modelled for each product j in order to allow prolonging or shortening the maturation times for specific customers.

Variable domains

$$T_{gko} \in \{0;1\} \qquad k \in K;\ g \in G;\ o \in O \quad (26)$$

$$V_{iko} \ge 0 \qquad k \in K;\ i \in I;\ o \in O \quad (27)$$

$$W_{is} \ge 0 \qquad i \in I;\ s \in S \quad (28)$$

$$fdp+1 \le START_{xo} < ldp \qquad k \in K;\ o \in O \qquad (29)$$

$$S_{pbl} \in \{0;1\} \qquad p \in P;\ b \in B;\ l \in LB(b) \qquad (30)$$

$$X_{jpsl} \ge 0 \qquad j \in J;\ p \in P;\ s \in S:\ s < p;\ l \in LJ(j) \qquad (31)$$

$$Z_{jpsd} \ge 0 \qquad j \in J;\ p \in P;\ s \in S:\ s<p;\ d \in D:\ s<d \& p<d \qquad (32)$$

$$Y_{jsd} \ge 0 \qquad j \in J;\ s \in S;\ d \in D:\ s<d \qquad (33)$$

$$L_{pbl} \ge 0 \qquad p \in P;\ b \in B;\ l \in LB(b) \qquad (34)$$

$$p \le END_{pbl} \le p+1 \qquad p \in P;\ b \in B;\ l \in LB(b) \qquad (35)$$

$$p \le EST_{lp} \le p+1 \qquad p \in P;\ l \in L \qquad (36)$$

$$p \le LFT_{lp} \le p+1 \qquad p \in P;\ l \in L \qquad (37)$$

$$SAO_l, SUO_l \ge 0 \qquad l \in L \qquad (38)$$

Two variables (T_{gko} and S_{bpl}) are binary; all other variables are continuous. The variables END_{pbl}, EST_{lp} and LFT_{lp} can only take values of the corresponding packaging day p. As the intermediate products must be produced before being sliced and packed ($s < p$) and the final products must be sliced and packed before being delivered ($p < d$), this sequence is guaranteed by the definition of the corresponding variables.

5 Numerical Results

The purpose of the numerical investigation is to assess the suitability of the model for specific planning problems in industry. The performance of the model is assessed along the dimensions Objective Value (OV), MIP-Gap, and computation time. OV is the value of the objective function at the moment at which the optimization run is stopped. The MIP-Gap is the difference in percent between the actual OV and a theoretical upper bound for the optimal OV which is obtained from a Linear Programming (LP) relaxation of the problem.

In order to assess the applicability of the proposed model for a real-life planning problem, first the number of positions for the scalding chambers must be determined. A low number of positions will substantially reduce the size of the model, especially with regard to the number of binary variables of the type T_{gko}. On the other side, a too low number of positions will also reduce the available capacity. Considering the duration of the scalding programs in the data set (4, 6 and 8 hours), 6 positions per day can be considered as an upper bound, as the shortest scalding program (4 hours) can be run 6 times per day at maximum. If the number of available positions per day is reduced to four, the capacity can still be utilized by 100% by 6- and 8-hour scalding programs; however, if only

4-hour scalding programs are required, the capacity can only be utilized for 16 hours per day, resulting in a capacity shortage of 33%. Some numerical analysis revealed that with 5 positions per day and chamber, high OVs and low MIP-gaps can be obtained within relatively short computational times which cannot be topped by a 6-positions-per-day model after one hour. Therefore, all following analysis has been performed using 5 positions per production day. For the numerical validation of the model, a data set is applied, which represents a mid- to large-size production environment. The resulting model contains 698 binary and 20,787 continuous decision variables.

In spite of the high number of binary variables, the model shows a relatively good solvability. Since the performance of the model regarding solvability, computational times and MIP-gaps can vary depending on the problem instance, the demand has been varied in order to assess the stability of the model (see Table 1). In all cases, a result has been obtained after less than 5 minutes. Nonetheless, in none of the three demand scenarios the first obtained solution cannot be improved even after several hours of CPU time. Yet, MIP-gaps from 2% to 3% are still satisfactory considering the two-stage planning problem.

Table 1: Model performance

Demand [% of original demand]	*t[s] for first solution*	*Objective value [T€]*	*MIP-gap*
80%	45 sec	313.4	2.8%
100%	85 sec	390.8	3.0%
120%	280 sec	454.7	2.0%

Further analysis has been performed regarding the impact of the shelf life dependent pricing component on the freshness of the products and the production costs per kg of the final product. In Figure 4, a factor of 1 represents shelf life dependent benefits as used in the data set (ca. 3.5% of the total revenues) – a factor of 0.1 means 10% of the initial value, a factor of 10 means 10 times the initial value. The freshness of the products - measured as the weighted average of the remaining shelf life on the total shelf life of the products - increases if the shelf life factor is increased, however, only to a limited extent (from 79% to ca. 81.5%). This is mainly due to the fact that products with a low shelf life dependent pricing component are now produced earlier and the products with a higher shelf life dependent pricing component are postponed, resulting in an only slight increase of the average shelf life. In addition, scalding and packaging must now be performed with lower batch sizes. Therefore, the higher shelf life benefits at a factor of 5 or 10 are levelled off against higher scalding and packaging costs,

which include not only a higher number of set-ups, but also the use of more expensive machines and of overtime capacity. Hence, the average production costs per kg of final product increased from € 1.27 at a shelf life factor of 0.1 to € 1.35 at a shelf life factor of 10.

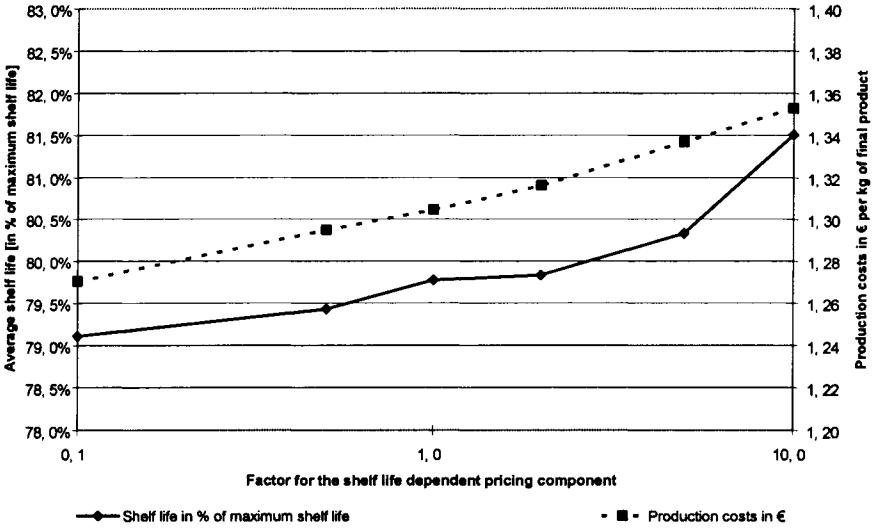


Figure 4: Variation of shelf life dependent pricing component (MIP gap: 2%)

6 Conclusions

In this paper, an MILP modelling approach has been presented that integrates shelf life into the weekly planning of the production of scalded sausages. The model covers the steps of scalding and packaging as well as the intermediate warehouse. To model certain features of fresh food business, a shelf life dependent pricing component has been integrated. Hence, the freshness of the products can now be influenced explicitly in the weekly production planning. The model has shown to be able to deliver scheduling results with acceptable MIP-gaps in a relatively short time. However, as the run-time of the model and the MIP-gap depend on the specific data set, the number of positions to use and also the appropriate run-times must first be evaluated if major parameters of the planning problem change (e.g. number and capacity of chambers and packaging lines or number and structure of intermediates, blocks and products).

If very large model instances (due to, for example, a high number of intermediate or final products or a high number of different scalding programs) must be resolved, a decomposition approach can be promising. However, in that case the model has to be reformulated in order to obtain a two stage planning procedure. At the beginning, the packaging lines that can process the same range of final products are optimized individually based on a first model formulation. The results of this first model represent the required volumes of intermediate products per day. A second model must then summarize all required quantities of intermediate products and must schedule the scalding chambers accordingly. The described procedure will help to reduce computational times. Nonetheless the overall solution quality will probably decrease since no overall optimization is carried out; local optima are obtained instead. Moreover, as the planner in industry will probably not be able to perform the decomposition on its own, the two new models must be developed by OR experts and delivered to the planner, along with clear guidelines on how to use the models.

However, for the success of shelf-life integrated planning for real life applications, the value of the shelf-life dependent pricing component needs to be determined. The implementation into existing supply chain management concepts may prove to be difficult. Developing suitable incentives based on customer satisfaction is therefore necessary. Yet, even without taking an integrated perspective onto the supply chain, this concept offers a suitable tool for a sausage manufacturer to consider freshness as a part of production planning. For markets characterized by intense competition, this provides an additional quality-oriented feature, which can constitute a pivotal competitive advantage. In addition to the processed meat industry, the applicability of these models for similar problems arising in the production of other fresh foods (e.g. fresh meat, dairy, fish, fruits, vegetables, or bakery goods) has to be examined.

References

[1] Abad PL (2003) Optimal pricing and lot-sizing under conditions of perishability, finite production and partial backordering and lost sales. European Journal of Operational Research 144:677–685

[2] Auer J (2001) BSE-Krise verschärft Strukturprobleme. Lebensmittel Zeitung, 09.03.2001:68

[3] Bourlakis MA, Weightman PWH (2004) Introduction to the UK food supply chain. In: Bourlakis MA, Weightman PWH (eds.) Food Supply Chain Management. Blackwell, Oxford et al., pp. 1–10

[4] Burkard RE, Fortuna T, Hurkens CAJ (2002) Makespan minimization for chemical batch processes using non-uniform time grids. Computers and Chemical Engineering 26:1321–1332

[5] Connor JM, Wills RL (1988) Marketing and market structure of the U.S. food processing industries. In: McCorkle CO Jr. (ed.) Economics of Food Processing in the United States. Academic Press, San Diego et al., pp. 117–166

[6] Cooke DL, Rohleder TR, Silver EA (2004) Finding effective schedules for the economic lot scheduling problem: A simple mixed integer programming approach. International Journal of Production Research 42:21–36

[7] Elmaghraby SE (1978) The economic lot scheduling problem: Review and extensions. Management Science 24:587–598

[8] Fairfield DK (2002) Fünf Fehlannahmen über CPFR. In: Seifert D (ed.) Collaborative Planning, Forecasting and Replenishment – Ein neues Konzept für state-of-the-art Supply Chain Management. Galileo, Bonn, pp. 205–224

[9] Fleischmann B (1990): The discrete lot-sizing and scheduling problem. European Journal of Operational Research 44:337–348

[10] Fleischmann B, Meyr H (1997) The general lot sizing and scheduling problem. European Journal of Operational Research 44:337–348

[11] Grievink JW, Josten L, Valk C (2002) State of the art in food: The changing face of the worldwide food industry. Elsevier Business Information, Arnhem

[12] Günther HO (1987) Planning lot sizes and capacity requirements in a single stage production system. European Journal of Operational Research 31:223–231

[13] Günther HO, Neuhaus U. (2004) Advanced planning and scheduling in the consumer goods industry: Realizing block planning concepts for make-and-pack production using MILP modelling and SAP's APO software. In: van Wassenhove LN, de Meyer A, Yücesan E, Günes ED, Muyldermans L (eds.) Operations management as a change agent, Vol. 2, INSEAD, pp. 757–766

[14] Haase K (1994) Lotsizing and Scheduling for Production Planning. Springer, Berlin et al.

[15] Haase, K. (1996): Capacitated lot-sizing with sequence dependent setup costs, OR Spektrum 18:51–59

[16] Ierapetritou MG, Floudas CA (1998) Effective continuous-time formulation for short-term scheduling - 1. Multipurpose batch processes - 2. Continuous and semicontinuous processes. Industrial and Engineering Chemistry Research 37:4341–4374

[17] Kallrath J. (2002) Planning and scheduling in the process industry. OR Spectrum 24:219–250
[18] Karmarkar US, Schrage L (1985) The deterministic dynamic product cycling problem. Operations Research 33:326–345
[19] Kilcast D, Subramaniam P (2000) The stability and shelf-life of food. Woodhead, Cambridge
[20] Kim YS (2001) Meat production. In: Hui YH, Nip WK, Rogers RW, Young OA (eds.) Meat Science and Applications. Marcel Dekker, New York et al., pp. 563–579
[21] Koçlar A, Süral H (2005) A note on "The general lot sizing and scheduling problem". OR Spectrum, 27:145–146
[22] KPMG Corporate Finance UK (2000) Europe's recipe for success – innovate and consolidate. Industry report of KPMG Corporate Finance UK
[23] Lebensmittel Zeitung (2001) LZ Report 2001/2002, Verlagsgruppe Deutscher Fachverlag, Frankfurt am Main
[24] Lütke Entrup M, Günther HO, van Beek P, Grunow M, Seiler T (2005) Mixed integer linear programming approaches to shelf life integrated planning and scheduling in yogurt production. International Journal of Production Research (to appear)
[25] Manna SK, Chaudhuri KS (2001) An economic order quantity model for deteriorating items with time-dependent deterioration rate, demand rate, unit production cost and shortages. International Journal of System Science 32:1003–1009
[26] McLaughlin E (2002) The food industry in 2005, a forecast. In Grievink JW, Josten L, Valk C (eds.) State of the Art in Food: The changing face of the worldwide food industry. Elsevier Business Information, Arnhem, pp. 598–608
[27] Méndez CA, Cerdá J (2002) An MILP-based approach to the short-term scheduling of make-and-pack continuous production plants. OR Spectrum 24:403–429
[28] Meyr H (1999) Simultane Losgrößen- und Reihenfolgeplanung für kontinuierliche Produktionslinien. Gabler, Wiesbaden
[29] Michael J, Funder J, Huppertz S (2002) KonsumGüter 2010: Handel-Macht-Marken – Eine Expertenbefragung und Trendstudie über die Entwicklung des deutschsprachigen Lebensmitteleinzelhandels und der Konsumgüterindustrie bis zum Jahr 2010. Industry Report of Accenture, Frankfurt am Main
[30] N.N. (2003) Globus plant Disposition im Stundentakt. Lebensmittel Zeitung, 28.11.2003:28
[31] Nahmias S (1982) Perishable inventory theory: A review. Operations Research 30:680–708

[32] Neuhäuser S (1996) Fleischwaren. In: Heiss R (ed.): Lebensmitteltechnologie – Biotechnologische, chemische, mechanische und thermische Verfahren der Lebensmittelverarbeitung. 5th edition, Springer, Berlin et al., pp. 51–61
[33] Oelker P (1996) Kochwurstherstellung. In Sielaff H (ed.) Fleischtechnologie. B. Behr's Verlag, Hamburg, pp. 383–396
[34] Pearson AM, Gillett TA (1996) Processed Meats. 3rd edition, Chapman & Hall, New York et al.
[35] Raafat F (1991) Survey of literature on continuously deteriorating inventory models. Journal of the Operational Research Society 42:27–37
[36] Sahinidis NV, Grossmann IE (1991) MINLP Model for cyclic multiproduct scheduling on continuous parallel lines. Computers and Chemical Engineering 15:85–103
[37] Savic IV (1985) Small-scale sausage production. FAO Animal Production and Health Paper 52, Food and Agriculture Organization of the United Nations, Rome
[38] Seifert D (2001) Efficient Consumer Response – Strategische Erfolgsfaktoren für die Wertschöpfungspartnerschaft von Industrie und Handel. Rainer Hampp, München et al.
[39] Sielaff H (1996) Fleischtechnologie. B. Behr's Verlag, Hamburg
[40] Sikora R, Chhajed D, Shaw MJ (1996) Integrating the lot-sizing and sequencing decisions for scheduling a capacitated flow line. Computers and industrial Engineering 30:659–679
[41] Soman CA, van Donk DP, Gaalman GJC (2004) A basic period approach to the economic lot scheduling problem with shelf life considerations. International Journal of Production Research 42:1677–1689
[42] Stadtler H (2005) Production planning and scheduling. In Stadtler H, Kilger C (eds.) Supply Chain Management and Advanced Planning, 3rd edition, Springer, Berlin et al., pp. 197–214
[43] Treeck S, Seishoff M (2002) CPFR in Deutschland, Österreich und der Schweiz In Seifert D (ed.) Collaborative Planning, Forecasting and Replenishment – Ein neues Konzept für state-of-the-art Supply Chain Management. Galileo, Bonn, pp. 265–277
[44] USDA, FSIS, AFDO (1999) Safe Practices for Sausage Production. U.S. Department of Agriculture (USDA), Food Safety and Inspection Service (FSIS) and The Association of Food and Drug Officials (AFDO), Washington D. C.
[45] van Wezel WMC (2001) Tasks, Hierarchies, and Flexibility – Planning in Food Processing Industries. Labyrint, Capelle a/d IJssel

[46] Wünsche K (2002) Die Nahrungsmittelindustrie – eine stabile Branche mit hohem Wachstum. In: Ernst & Young, Deutsche Investitions- und Entwicklungsgesellschaft. F.A.Z.-Institut, manager magazin (eds.): Nahrungsmittelindustrie – Trends und Perspektiven im Food Processing, Köln, pp. 9–15

[47] Xiong YL, Mikel WB (2001) Meat and meat products. In: Hui YH, Nip WK, Rogers RW, Young OA (eds.) Meat Science and Applications. Marcel Dekker, New York et al., pp. 351–369

Optimale Anpassung im Gutenberg-Produktionsmodell: Eine analytische Ermittlung der Kostenfunktion aus den Karush-Kuhn-Tucker-Bedingungen

Heinz Isermann, Joachim Houtman und Eric Sucky
Seminar für Logistik und Verkehr
Johann Wolfgang Goethe-Universität Frankfurt am Main

1 Einleitung

Im Gutenberg-Produktionsmodell werden drei Anpassungsformen an alternative Beschäftigungen (Produktquantitäten) unterschieden: die zeitliche, die intensitätsmäßige und die quantitative Anpassung (vgl. Gutenberg, 1983). Zeitliche Anpassung bedeutet, dass bei konstanter Anzahl der eingesetzten Potenzialfaktoren eines Typs (Aggregate) und konstanter Intensität (Produktionsgeschwindigkeit) die Einsatzzeit der Aggregate an die zu erzeugende Produktquantität angepasst wird. Intensitätsmäßige Anpassung liegt vor, wenn bei fester Anzahl der eingesetzten Aggregate gleichen Typs und gegebener Einsatzzeit die Intensität der Potenzialfaktoren in Abhängigkeit der Produktquantität variiert wird. Von quantitativer Anpassung wird schließlich gesprochen, wenn bei vorgegebener Intensität und Einsatzzeit die Anzahl der eingesetzten Aggregate variiert wird (vgl. z. B. Fandel, 1996). Es handelt sich dabei um kurzfristige Anpassungsformen, da die Anpassung an alternative Produktquantitäten auf der Basis der zur Verfügung stehenden Potenzialfaktoren erfolgt, d. h. diese Anpassungsformen beziehen sich ausschließlich auf im Betrieb vorhandene Aggregate und sind daher sofort realisierbar (vgl. Bloech et al., 2001). Die zeitliche, intensitätsmäßige und quantitative Anpassung bestimmen die Leistungsabgaben der vorhandenen Potenzialfaktoren und damit sowohl die resultierenden Faktorverbräuche als auch die damit verbunden Kosten. In der betrieblichen Praxis werden diese Anpassungsformen i. d. R. kombiniert. Es stellt sich somit das Problem, die drei Anpassungsformen derart zu gestalten, dass die gewünschte (vorgegebene) Produktquantität effizient erzeugt wird.

Das betrachtete Auswahlproblem besteht in der Ermittlung der kostenminimalen Anpassungsform für alle realisierbaren Produktquantitäten. Stellvertretend für die Vielzahl der Beiträge, welche die jeweils optimale Anpassungsform ableiten, seien an dieser Stelle die grundlegenden Arbeiten von Albach (1962a), Jacob (1962), Pack (1963, 1966), Dellmann und Nastansky (1969), Lücke (1969), Karrenberg und Scheer (1970), Adam (1972), Schüler (1973), (1975)

und Botta (1974) genannt. Eine weiterführende Analyse der kostenminimalen Kombination der Anpassungsformen unter besonderer Berücksichtigung variabler Faktorpreise findet sich z. B. bei Lambrecht (1978). Eine umweltorientierte Darstellung der Produktionsfunktion von Gutenberg unter Beachtung des Anfalls unerwünschter Nebengüter (z. B. Abfälle, Schadstoffe) entwickeln Dinkelbach und Rosenberg (2000). Empirische Überprüfungen der Produktionsfunktion von Gutenberg finden sich z. B. bei Hall (1959) und Pressmar (1971). Die Bestimmung der Produktions- und Kostenfunktion für ein konkretes Dieselaggregat zur Stromerzeugung präsentieren Houtman und Sucky (2001). Einen Überblick der empirischen Arbeiten zur Gutenberg-Produktionsfunktion gibt Fischer (1980).

Im vorliegenden Beitrag wird das wohlbekannte betriebswirtschaftliche Problem der Bestimmung der optimalen Anpassungsform für alle mit einem Aggregat alternativ realisierbaren Produktquantitäten erneut aufgegriffen. Wird ein einzelnes Aggregat betrachtet, kann sowohl dessen Einsatzzeit (zeitliche Anpassung) bei konstanter Intensität, mit der das Aggregat betrieben wird, als auch die Intensität (intensitätsmäßige Anpassung) bei gegebener Einsatzzeit, variiert werden. Die optimale Anpassungsform für alle mit einem Aggregat realisierbaren Produktquantitäten bildet dann die Basis zur Ermittlung der optimalen Kombination aus zeitlicher, intensitätsmäßiger und quantitativer Anpassung im Fall mehrerer, funktionsgleicher Potenzialfaktoren (vgl. z. B. Jakob, 1962).

Werden sowohl die Einsatzzeit eines Aggregats als auch die Intensität, mit der das Aggregat betrieben wird, als Entscheidungsvariablen interpretiert (vgl Dinkelbach und Rosenberg, 2000), kann das Problem der Auswahl der kostenminimalen Anpassungsform für eine gegebene Produktquantität als nichtlineares Optimierungsproblem formuliert werden. Für nichtlineare Optimierungsprobleme stellen die erstmals von Karush (1939) und später von Kuhn und Tucker (1951) abgeleiteten Karush-Kuhn-Tucker-Bedingungen notwendige Optimalitätskriterien dar. Während die Karush-Kuhn-Tucker-Bedingungen oftmals zur Optimalitätsprüfung einer gegebenen Lösung herangezogen werden, wird in diesem Beitrag ein anderer Weg beschritten. Im vorliegenden Beitrag wird gezeigt, dass die Auswahl der kostenminimalen Anpassungsform unmittelbar unter Verwendung der Karush-Kuhn-Tucker-Bedingungen analytisch hergeleitet werden kann, sodass auf die in der relevanten Literatur verwendeten Ersatzmodelle unter Heranziehung der Linearen Programmierung oder der Dynamischen Optimierung verzichtet werden kann.

Bei Einsatz eines Aggregats ist die kostenminimale Anpassungsform für eine gegebene Produktquantität durch eine bestimmte Kombination aus Einsatzzeit und Intensität gekennzeichnet, die im Weiteren als Minimalkostenkombination bezeichnet wird.

Der Begriff der Minimalkostenkombination ist eng verbunden mit so genannten substitutionalen Produktionsfunktionen. Die Substitutionsmöglichkeiten der

Faktoreinsätze bilden hierbei die Grundlage für den Begriff der substitutionalen Produktionsfunktion. Die Entscheidungsvariablen im Rahmen der Minimalkostenkombination sind dann die zur Erzeugung einer bestimmte Produktquantität zu wählenden Faktoreinsätze.

Im Rahmen dieses Beitrags wird somit ein weiter gefasster Begriff der Minimalkostenkombination verwendet. Es wird verdeutlicht, dass bei konsequenter Interpretation der Einsatzzeit und der Intensität als Entscheidungsvariablen eine den substitutionalen Produktionsfunktionen analoge Ableitung der Minimalkostenkombination als kostenminimale Kombination aus Einsatzzeit und Intensität zur Erzeugung einer bestimmten Produktquantität, gelingt. Zur Verwendung des Begriffs der Minimalkostenkombination im Rahmen des Gutenberg-Produktionsmodells vgl. Jakob (1962). Aus den ermittelten Minimalkostenkombinationen eines Aggregats für alle im Planungszeitraum realisierbaren Produktquantitäten wird schließlich die Kostenfunktion bestimmt.

2 Das Produktionsmodell von Gutenberg

2.1 Minimalkostenkombination und Kostenfunktion

Eine Kostenfunktion stellt die funktionale Abhängigkeit zwischen den Kosteneinflussgrößen und den Produktionskosten im Planungszeitraum dar. Diese Kosteneinflussgrößen sind u.a. Faktorqualität, Faktorpreise, Beschäftigung, Betriebsgröße, Fertigungsprogramm und die Gestaltung des Fertigungsablaufs (vgl. Gutenberg, 1983). Insbesondere interessiert der „Einfluß von Beschäftigungsschwankungen auf die Produktionskosten" (Gutenberg, 1983), d. h. die Angabe einer Kostenfunktion K(x), welche die gesamten minimalen Kosten im Planungszeitraum in Abhängigkeit von der Beschäftigung x, gemessen in Produkteinheiten [PE] des zu erzeugenden Produkts, angibt. Das kostentheoretische Auswahlproblem besteht in der Ermittlung der Minimalkostenkombination: Sind bis auf die Quantitäten $r_1,r_2,...,r_n$ der einzusetzenden Faktoren, gemessen in Einheiten des Faktors i [FE_i], sämtliche Kosteneinflussgrößen konstant und sind positive Faktorpreise $q_1,q_2,...,q_n$, gemessen in Geldeinheiten je Faktoreinheit [GE/FE_i], gegeben, so sind für eine bestimmte Produktquantität x [PE] die Faktoreinsatzquantitäten so zu wählen, dass die Kosten minimal werden. Eine parametrische Variation der Produktquantität x im Rahmen der Minimalkostenkombination liefert die Kostenfunktion K(x) (vgl. Dinkelbach und Rosenberg, 2000).

2.2 Das Gutenberg-Produktionsmodell

Im Folgenden wird ein Aggregat betrachtet, an dem durch den Einsatz von n Verbrauchsfaktoren die Quantitäten x [PE] eines Produkts erzeugt werden. Die

Ausbringungsquantität x hängt ab von der Intensität $\lambda = \frac{x}{t}$ (Produktionsgeschwindigkeit), gemessen in Produkteinheiten x je Zeiteinheit t [PE/ZE], mit der das Aggregat betrieben wird, der Einsatzzeit t [ZE] des Aggregats sowie den technischen Eigenschaften $\mathbf{z}(t)$, d. h. $x = x(\lambda, t, \mathbf{z}(t))$. Der Zustandsvektor $\mathbf{z}(t)$ (z-Situation) charakterisiert die technische Leistungsfähigkeit des Aggregats zum Zeitpunkt t und ist eine Kosteneinflussgröße. Im Rahmen dieses Beitrags wird von einer konstanten z-Situation ausgegangen. Eine Untersuchung der Auswirkungen von sich im Zeitverlauf verändernden technischen Eigenschafen eines Aggregats, z.B. aufgrund technischen Verschleiß, auf Investitionsentscheidungen führt Albach (1962b) durch. Analysen der Auswirkungen der z-Situation auf kurz- und langfristige Produktions- und Investitionsentscheidungen haben Luhmer (1975), Stepan (1981) sowie Kistner und Luhmer (1988) durchgeführt.

Wird die Intensität $\hat{\lambda}$, mit der das Aggregat betrieben wird, in Arbeitseinheiten b [AE] je Zeiteinheit t [ZE] gemessen, d. h. $\hat{\lambda} = \frac{b}{t}$ [AE/ZE], so ergibt sich die Leistungsabgabe b [AE] des Aggregats im Planungszeitraum mit $b = \hat{\lambda} \cdot t$. Zwischen der technischen Leistung b [AE] des Aggregates und der ökonomischen Leistung, gemessen in Produkteinheiten x [PE], wird von einem proportionalen Zusammenhang $c \cdot x = b$ ausgegangen, wobei c [AE/PE] die notwendige Anzahl an Arbeitseinheiten je Produkteinheit angibt. Mit $b = \hat{\lambda} \cdot t$ und $c \cdot x = b$ ergibt sich die proportionale Beziehung $x = \frac{1}{c} \cdot \hat{\lambda} \cdot t$ (vgl. Fandel, 1996). In diesem Beitrag wird von c=1 ausgegangen, sodass die proportionale Beziehung $x = \lambda \cdot t$ gilt, mit $\lambda \in [\lambda^{min}, \lambda^{max}]$, $(0 < \lambda^{min} < \lambda^{max})$ und $t \in [0, t^{max}]$, $(t^{max} > 0)$.

Das Gutenberg-Produktionsmodell gestattet die Darstellung der Faktorfunktionen $r_i = a_i \cdot x = a_i \cdot \lambda \cdot t$ (i=1,2,...,n) mit a_i als Produktionskoeffizient des i-ten Faktors, gemessen in Einheiten des Faktors i je Produkteinheit [FE_i/PE]. Die Ausprägungen der Produktionskoeffizienten hängen ab von der Intensität λ, mit der das Aggregat betrieben wird. Es ergeben sich die Funktionen $a_i(\lambda)$ (i=1,2,...,n), die als technische Verbrauchsfunktionen (vgl. Gutenberg, 1983) oder Durchschnittsverbrauchsfunktionen bezeichnet werden. Mit den technischen Verbrauchsfunktionen $a_i = a_i(\lambda)$ (i=1,2,...n) und der Beziehung $x = \lambda \cdot t$ ergeben sich die Gesamtverbrauchsfunktionen der n Faktoren zu:

$$r_i(\lambda, t) = a_i(\lambda) \cdot (\lambda \cdot t) \qquad i=1,2,...,n \tag{1}$$

Dieses System von Faktorfunktionen (1) wird Gutenberg-Produktionsmodell genannt. Eine allgemeine Darstellung des Gutenberg-Produktionsmodells als Technologie geben Dinkelbach und Rosenberg (2000).

2.3 *Das kostentheoretische Auswahlproblem im Gutenberg-Produktionsmodell*

Die Faktoreinsatzquantitäten je Produkteinheit, dargestellt durch die Produktionskoeffizienten $a_i(\lambda)$ (i=1,2,...,n) sind nicht konstant, sondern hängen von der gewählten Intensität λ ab (vgl. Dinkelbach und Rosenberg, 2000) Die variablen Gesamtkosten können daher nicht in direkter Abhängigkeit von der Produktquantität x angegeben werden. Unter der Voraussetzung konstanter, positiver Faktorpreise $q_1,q_2,...,q_n$ lauten die Kosten des Faktorverbrauchs in Abhängigkeit von der gewählten Intensität λ und der Einsatzzeit t des Aggregats:

$$K(\lambda,t)=\sum_{i=1}^{n} q_i \cdot r_i(\lambda,t)=\left(\sum_{i=1}^{n} q_i \cdot a_i(\lambda)\right)\cdot(\lambda\cdot t)=k(\lambda)\cdot(\lambda\cdot t). \qquad (2)$$

Werden Verbrauchsfunktionen wie in (2) mit konstanten Faktorpreisen bewertet, so ergibt sich lediglich eine Verschiebung der Lage der Verbrauchsfunktionen. Der typische Verlauf der Verbrauchsfunktionen bleibt jedoch erhalten (vgl. Lücke, 1969). Die Funktion $k(\lambda)=\sum_{i=1}^{n} q_i \cdot a_i(\lambda)$ heißt Stückkostenfunktion. In diesem Beitrag werden streng konvexe Stückkostenfunktion $k(\lambda)$ im Definitionsbereich $[\lambda^{min},\lambda^{max}]$ angenommen. Positive Faktorpreise vorausgesetzt ergeben sich streng konvexe Stückkostenfunktionen aus konvexen Verbrauchsfunktionen $a_i(\lambda)$ mit mindestens einer streng konvexen Verbrauchsfunktion. Konkave Verbrauchsfunktionen stellen in praktischen Anwendungen die Ausnahme dar.

Unter den hier getroffenen Annahmen existiert genau ein λ^*, für das die Stückkostenfunktion $k(\lambda)$ ihr Minimum annimmt (vgl. Kistner, 1993). Das Minimum kann dabei in einem Randpunkt liegen, d. h. in λ^{min} bzw. λ^{max}. Abbildung 1 zeigt beispielhaft den streng konvexen Verlauf der Stückkostenfunktion $k(\lambda)=1/20\cdot\lambda^2-2\cdot\lambda+24$ im Definitionsbereich $10\le\lambda\le 30$, mit $\lambda^*=20$.

Die Aufgabenstellung der Minimalkostenkombination im Rahmen des Gutenberg-Produktionsmodells besteht zum einen in der Ermittlung kostenminimaler Kombinationen aus Intensität λ und Einsatzzeit t (Minimalkostenkombinationen) für alternative Produktquantitäten x und zum anderen in der Angabe der Kostenfunktion. Es ist das folgende nichtlineare Optimierungsproblem zu lösen:

$$\min K(\lambda, t) = k(\lambda) \cdot (\lambda \cdot t) \tag{3}$$

unter den Nebenbedingungen

$$x = \lambda \cdot t \tag{4}$$

$$(\lambda, t) \in B := [\lambda^{min}, \lambda^{max}] \times [0, t^{max}] \tag{5}$$

mit den Entscheidungsvariablen λ und t.

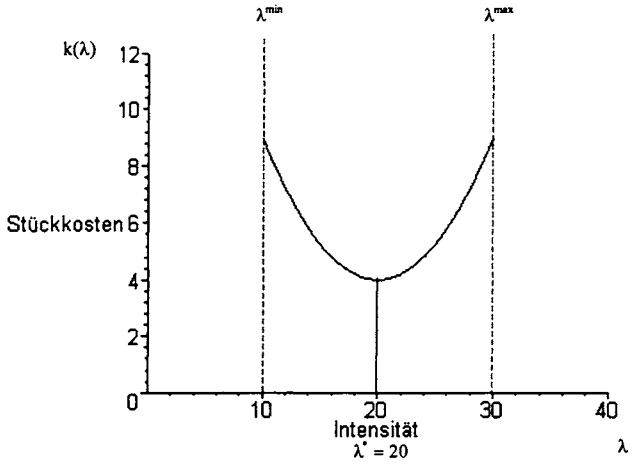


Abbildung 1: Verlauf der Stückkostenfunktion

Das nichtlineare Optimierungsproblem (3) – (5) kann in einem (λ,t)-Koordinatensystem illustriert werden (vgl. Abbildung 2). Mit den Grenzen für die Entscheidungsvariablen λ und t ergibt sich der Bereich B der zulässigen (λ,t)-Kombinationen. Eine Darstellung der zulässigen (λ,t)-Kombinationen in einem (λ,t)-Koordinatensystem findet sich bereits bei Pack (1963). Zur Darstellung der zulässigen (λ,t)-Kombinationen vgl. auch Dinkelbach und Rosenberg (2000) sowie Bloech et al. (2001). Die Produktisoquanten für gegebene Quantitäten x ergeben sich mit $x = \lambda \cdot t$ im Bereich der zulässigen (λ,t)-Kombinationen (vgl. Dinkelbach und Rosenberg, 2000).

Das Entscheidungsproblem besteht in der Bestimmung der kostenminimalen (λ,t)-Kombinationen aus den zulässigen (λ,t)-Kombinationen für jede realisierbare Produktquantität x und somit in der Ermittlung der Kostenfunktion K(x). Die folgende Abbildung 2 zeigt die konvexe Menge B der zulässigen (λ,t)-Kombinationen für $10 \le \lambda \le 30$, $0 \le t \le 10$ sowie die Produktisoquante zur Produktquantität x=150.

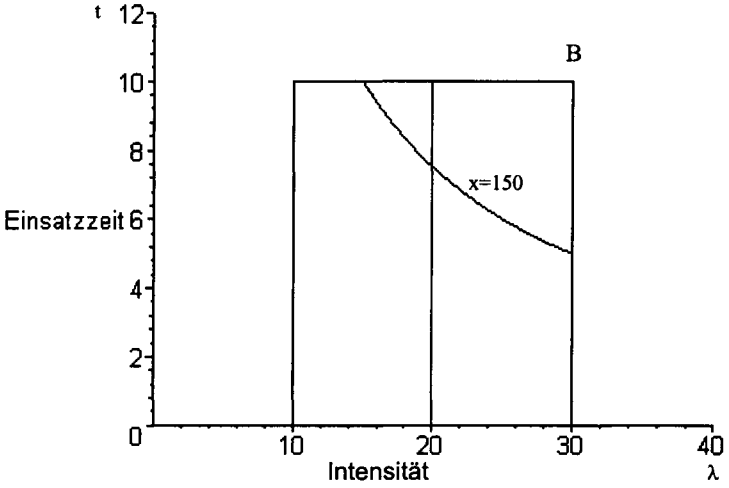


Abbildung 2: Zulässige (λ,t)-Kombinationen und Produktisoquante

3 Analytische Ermittlung der Kostenfunktion aus den Karush-Kuhn-Tucker-Bedingungen

Das nichtlineare Optimierungsproblem lautet:

$$\min K(\lambda, t) = k(\lambda) \cdot (\lambda \cdot t) \tag{6}$$

unter den Nebenbedingungen

$$x - \lambda \cdot t = 0 \tag{7}$$

$$t \geq 0 \tag{8}$$

$$t^{max} - t \geq 0 \tag{9}$$

$$\lambda - \lambda^{min} \geq 0 \tag{10}$$

$$\lambda^{max} - \lambda \geq 0\,. \tag{11}$$

Die analytische Lösung des nichtlinearen Optimierungsproblems (6) – (11) erfolgt mittels der Karush-Kuhn-Tucker-Bedingungen. Diese Bedingungen wurden unabhängig voneinander durch Karush im Jahre 1939 sowie Kuhn und Tucker im Jahre 1951 abgeleitet. Jedoch wurde die Arbeit von Karush nie veröffentlicht und dadurch lange Zeit ignoriert (vgl. Kuhn, 1991). Zur Herleitung der Karush-Kuhn-Tucker-Bedingungen vgl. Ellinger et al. (1998), Ohse (1998), Zimmermann (1999), Domschke und Drexl (1995) sowie Stepan und Fisher (1996). Zu den Unterschieden in den Ansätzen vgl. Kuhn (1991).

Das nicht-restringierte Ersatzproblem lautet:

$$\begin{aligned}\min L(\lambda,t,\mu_1,\mu_2,\mu_3,\mu_4,\mu_5) = k(\lambda)\cdot(\lambda\cdot t)+\mu_1\cdot(x-\lambda\cdot t)+\mu_2\cdot t \\ +\mu_3\cdot\left(t-t^{max}\right)+\mu_4\cdot\left(\lambda-\lambda^{min}\right)+\mu_5\cdot\left(\lambda^{max}-\lambda\right)\end{aligned} \tag{12}$$

Die Funktion L ist im Allgemeinen nicht konvex in λ und linear in t (vgl. Abbildung 5). Die resultierenden Karush-Kuhn-Tucker-Bedingungen sind somit notwendige Bedingungen für ein Minimum:

$$\frac{\partial L}{\partial \lambda} = k'(\lambda)\cdot\lambda\cdot t + k(\lambda)\cdot t - \mu_1\cdot t + \mu_4 - \mu_5 \geq 0 \tag{13}$$

$$\lambda\cdot\frac{\partial L}{\partial \lambda} = k'(\lambda)\cdot\lambda^2\cdot t + k(\lambda)\cdot\lambda\cdot t - \mu_1\cdot\lambda\cdot t + \mu_4\cdot\lambda - \mu_5\cdot\lambda = 0 \tag{14}$$

$$\frac{\partial L}{\partial t} = k(\lambda)\cdot\lambda - \mu_1\cdot\lambda + \mu_2 - \mu_3 \geq 0 \tag{15}$$

$$t\cdot\frac{\partial L}{\partial t} = k(\lambda)\cdot\lambda\cdot t - \mu_1\cdot\lambda\cdot t + \mu_2\cdot t - \mu_3\cdot t = 0 \tag{16}$$

$$x-\lambda\cdot t = 0 \quad (17) \qquad \mu_1 \geq 0 \quad (18) \qquad \mu_1\cdot(x-\lambda\cdot t) = 0 \quad (19)$$

$$t \geq 0 \quad (20) \qquad \mu_2 \geq 0 \quad (21) \qquad \mu_2\cdot t = 0 \quad (22)$$

$$t^{max} - t \geq 0 \quad (23) \qquad \mu_3 \geq 0 \quad (24) \qquad \mu_3\cdot(t^{max}-t) = 0 \quad (25)$$

$$\lambda-\lambda^{min} \geq 0 \quad (26) \qquad \mu_4 \geq 0 \quad (27) \qquad \mu_4\cdot(\lambda-\lambda^{min}) = 0 \quad (28)$$

$$\lambda^{max}-\lambda \geq 0 \quad (29) \qquad \mu_5 \geq 0 \quad (30) \qquad \mu_5\cdot(\lambda^{max}-\lambda) = 0 \quad (31)$$

Aus den Karush-Kuhn-Tucker-Bedingungen (13) bis (31) wird unter Beachtung der Zielfunktion (8) die Kostenfunktion K(x) abgeleitet. Diese Herleitung erfolgt zunächst unter der Annahme, dass für die kostenminimale Intensität λ^* gilt $\lambda^* \in]\lambda^{min},\lambda^{max}[$, mit $k'(\lambda^*)=0$. Die ermittelten Ergebnisse lassen sich unmittelbar auf $\lambda^* = \lambda^{min}$ und $\lambda^* = \lambda^{max}$ übertragen. Für positive Produktquantitäten $x > 0$ muss die Einsatzzeit gemäß (17) $t > 0$ sein, sodass generell $\mu_2 = 0$ gilt. Mit $\lambda \geq \lambda^{min} > 0$ folgt aus (14) unter Beachtung von $\mu_2 = 0$:

$$k'(\lambda)\cdot\lambda^2\cdot t + k(\lambda)\cdot\lambda\cdot t - \mu_1\cdot\lambda\cdot t + \mu_4\cdot\lambda - \mu_5\cdot\lambda = 0 \tag{32}$$

$$\Leftrightarrow \lambda\cdot(k'(\lambda)\cdot\lambda\cdot t + k(\lambda)\cdot t - \mu_1\cdot t + \mu_4 - \mu_5) = 0 \tag{33}$$

$$\Leftrightarrow k'(\lambda)\cdot\lambda\cdot t + k(\lambda)\cdot t - \mu_1\cdot t + \mu_4 - \mu_5 = 0\,. \tag{34}$$

Mit $t > 0$ und $\mu_2 = 0$ folgt aus (16):

$$k(\lambda)\cdot\lambda\cdot t-\mu_1\cdot\lambda\cdot t-\mu_3\cdot t=0 \tag{35}$$

$$\Leftrightarrow\ t\cdot(k(\lambda)\cdot\lambda-\mu_1\cdot\lambda-\mu_3)=0 \tag{36}$$

$$\Leftrightarrow\ k(\lambda)\cdot\lambda-\mu_1\cdot\lambda-\mu_3=0\,. \tag{37}$$

Da die Zielfunktion (6) nicht streng konvex ist, können Kostenminima im Inneren $\overset{\circ}{B}$ von B oder am Rand Rd(B) von B ($B:=[\lambda^{min},\lambda^{max}]\times[0,t^{max}]$) auftreten. Hierbei ist $x=\lambda\cdot t$ zu beachten: Die geforderte Produktquantität x muss durch mindestens eine zulässige (λ,t)-Kombinationen erfüllt sein. Es erfolgt eine disjunkte Zerlegung der Menge der realisierbaren Produktquantitäten (vgl. Abbildung 3) in:

$$[0,x^{max}]=\{0\}\cup\,]0,x_1[\,\cup\{x_1\}\cup\,]x_1,x_2[\,\cup\{x_2\}\cup\,]x_2,x^{max}[\,\cup\{x^{max}\},$$

$$\text{mit } x_1=\lambda^{min}\cdot t^{max},\ x_2=\lambda^{*}\cdot t^{max} \text{ und } x^{max}=\lambda^{max}\cdot t^{max}. \tag{38}$$

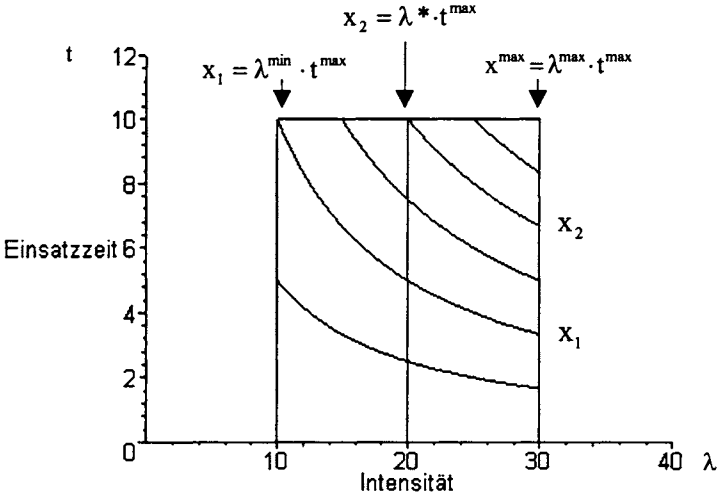


Abbildung 3: Produktisoquanten für alternative Produktquantitäten

3.1 Optimale (λ,t)-Kombinationen für Produktquantitäten $x\in]x_1,x_2[$

Zu gegebenen x bestimmt sich $t=\frac{x}{\lambda}$, mit $(\lambda,t)\in B$. Sei $x\in]x_1,x_2[$ gegeben, so ergeben sich die inneren Lösungen für $x\in]x_1,x_2[$ durch $x=\lambda\cdot t$, mit $(\lambda,t)\in\overset{\circ}{B}=B\setminus Rd(B)$ und die Randlösungen für $x\in]x_1,x_2[$ durch $x=\lambda\cdot t$,

mit $\lambda = \lambda^{max}$, $t = \frac{x}{\lambda^{max}}$ bzw. $t = t^{max}$, $\lambda = \frac{x}{t^{max}}$ (vgl. Abbildung 3). Zur Prüfung der Karush-Kuhn-Tucker-Bedingungen werden drei Fälle unterschieden:

$x = \lambda \cdot t$	Intensität λ	Einsatzzeit t	Daraus folgt für die Lagrange-Multiplikatoren (vgl. (19) – (33))
a) $x \in Rd(B)$	$\lambda^{min} < \lambda < \lambda^* < \lambda^{max}$	$t = t^{max}$	$\mu_4 = \mu_5 = 0$
b) $x \in Rd(B)$	$\lambda = \lambda^{max}$	$0 < t < t^{max}$	$\mu_3 = \mu_4 = 0$
c) $x \in \overset{\circ}{B}$	$\lambda^{min} < \lambda < \lambda^{max}$	$0 < t < t^{max}$	$\mu_3 = \mu_4 = \mu_5 = 0$

Ad a)
Mit $\mu_4 = \mu_5 = 0$ folgt aus (34):

$$t^{max} \cdot (k'(\lambda) \cdot \lambda + k(\lambda) - \mu_1) = 0 \tag{39}$$

$$\Rightarrow k'(\lambda) \cdot \lambda + k(\lambda) = \mu_1 . \tag{40}$$

Einsetzen in (37) ergibt:

$$k(\lambda) \cdot \lambda - \lambda \cdot (k'(\lambda) \cdot \lambda + k(\lambda)) - \mu_3 = 0 \tag{41}$$

$$\Rightarrow -k'(\lambda) \cdot \lambda^2 = \mu_3 . \tag{42}$$

Aufgrund $0 < \lambda < \lambda^*$ ist $\lambda^2 > 0$ und $k'(\lambda) < 0$, so dass sich $\mu_3 > 0$ ergibt. Für positive Produktquantitäten $x \in]x_1, x_2[$ ist eine optimale Lösung mit $\lambda = \frac{x}{t^{max}}$ und $t = t^{max}$ vorläufig nicht auszuschließen. Wird diese (λ,t)-Kombination in die Zielfunktion (6) eingesetzt ergibt sich:

$$K\left(\frac{x}{t^{max}}, t^{max}\right) = k\left(\frac{x}{t^{max}}\right) \cdot \left(\frac{x}{t^{max}} \cdot t^{max}\right) = k\left(\frac{x}{t^{max}}\right) \cdot x .$$

Ad b)
$\mu_3 = \mu_4 = 0$ liefert mit (37):

$$k(\lambda^{max}) \cdot \lambda^{max} - \mu_1 \cdot \lambda^{max} = 0 \tag{43}$$

$$\Rightarrow k(\lambda^{max}) = \mu_1 . \tag{44}$$

Einsetzen in (34) ergibt:

$$k'(\lambda^{max}) \cdot \lambda^{max} \cdot t + k(\lambda^{max}) \cdot t - k(\lambda^{max}) \cdot t - \mu_5 = 0 \tag{45}$$

$$\Rightarrow k'(\lambda^{max}) \cdot \lambda^{max} \cdot t = \mu_5 . \tag{46}$$

Aufgrund von $0 < \lambda^* < \lambda^{max}$ ist $k'(\lambda^{max}) > 0$. Mit $t > 0$ folgt daher $\mu_5 > 0$. Für positive Produktquantitäten $x \in]x_1, x_2[$ ist eine optimale Lösung mit $\lambda = \lambda^{max}$ und $t = \frac{x}{\lambda^{max}}$ vorläufig nicht auszuschließen. Wird diese (λ,t)-Kombination in die Zielfunktion (6) eingesetzt ergibt sich:

$$K\left(\lambda^{max}, \frac{x}{\lambda^{max}}\right) = k\left(\lambda^{max}\right) \cdot \left(\lambda^{max} \cdot \frac{x}{\lambda^{max}}\right) = k\left(\lambda^{max}\right) \cdot x .$$

Ad c)

$\mu_3 = \mu_4 = \mu_5 = 0$ und (37) ergeben:

$$k(\lambda) \cdot \lambda - \mu_1 \cdot \lambda = 0 \tag{47}$$

$$\Rightarrow k(\lambda) = \mu_1 . \tag{48}$$

Einsetzen in (34) ergibt:

$$k'(\lambda) \cdot \lambda \cdot t + k(\lambda) \cdot t - k(\lambda) \cdot t = 0 \tag{49}$$

$$\Rightarrow k'(\lambda) = 0 \tag{50}$$

$$\Rightarrow \lambda = \lambda^* . \tag{51}$$

Aufgrund von $k'(\lambda^*) = 0$ ist für positive Produktquantitäten $x \in]x_1, x_2[$ eine optimale Lösung mit $\lambda = \lambda^*$ und $t = \frac{x}{\lambda^*}$ vorläufig nicht auszuschließen. Wird diese (λ,t)-Kombination in die Zielfunktion (6) eingesetzt ergibt sich:

$$K\left(\lambda^*, \frac{x}{\lambda^*}\right) = k\left(\lambda^*\right) \cdot \left(\lambda^* \cdot \frac{x}{\lambda^*}\right) = k\left(\lambda^*\right) \cdot x .$$

Aus $k(\lambda) > k(\lambda^*)$ für $\lambda \neq \lambda^*$ folgt, dass für positive Produktquantitäten $x \in]x_1, x_2[$ nur die Kombination (λ^*, t) mit $t = \frac{x}{\lambda^*}$ optimal ist. Die Kostenfunktion im Bereich $x \in]x_1, x_2[$ lautet daher:

$$K(x) = k(\lambda^*) \cdot x \qquad \text{für } x_1 < x < x_2 \tag{52}$$

3.2 Optimale (λ,t)-Kombinationen für Produktquantitäten $x \in]0, x_1[$

Den nächsten zu untersuchenden Bereich positiver Produktquantitäten stellt das Intervall $]0, x_1[$ dar. Es gilt weiterhin $\mu_2 = 0$. Für positive Produktquantitäten $0 < x < x_1 = \lambda^{min} \cdot t^{max}$ ist $t < t^{max}$ und damit auch $\mu_3 = 0$. Aus (37) folgt:

$$k(\lambda) = \mu_1 . \tag{53}$$

Einsetzen in (34) ergibt:

$$k'(\lambda)\cdot\lambda\cdot t = -\mu_4 + \mu_5 . \tag{54}$$

Analog zu dem ersten untersuchten Bereich positiver Produktquantitäten können auch hier zur Prüfung der Karush-Kuhn-Tucker-Bedingungen wiederum drei Fälle unterschieden werden.

$x = \lambda \cdot t$	Intensität λ	Einsatzzeit t	Daraus folgt für die Lagrange-Multiplikatoren (vgl. (19) – (33))
a) $x \in Rd(B)$	$\lambda^{min} = \lambda < \lambda^* < \lambda^{max}$	$0 < t < t^{max}$	$\mu_3 = \mu_5 = 0$
b) $x \in Rd(B)$	$\lambda^{min} < \lambda^* < \lambda = \lambda^{max}$	$0 < t < t^{max}$	$\mu_3 = \mu_4 = 0$
c) $x \in \overset{\circ}{B}$	$\lambda^{min} < \lambda < \lambda^{max}$	$0 < t < t^{max}$	$\mu_3 = \mu_4 = \mu_5 = 0$

Ad a)

Mit $\mu_3 = \mu_5 = 0$ und (54) ergibt sich:

$$k'(\lambda^{min})\cdot\lambda^{min}\cdot t = -\mu_4 . \tag{55}$$

Aufgrund von $k'(\lambda^{min}) < 0$, $\lambda^{min} > 0$ sowie $t > 0$ ist $\mu_4 > 0$. Eine optimale Lösung ist durch die Kombination (λ^{min}, t), mit $t = \frac{x}{\lambda^{min}}$ vorläufig nicht auszuschließen. Wird diese (λ,t)-Kombination in die Zielfunktion (6) eingesetzt, ergibt sich: $K\left(\lambda^{min}, t = \frac{x}{\lambda^{min}}\right) = k\left(\lambda^{min}\right)\cdot x$.

Ad b)

Mit $\mu_3 = \mu_4 = 0$ und (54) ergibt sich:

$$k'(\lambda^{max})\cdot\lambda^{max}\cdot t = \mu_5 . \tag{56}$$

Aufgrund von $k'(\lambda^{max}) > 0$, $\lambda^{max} > 0$ sowie $t > 0$ ist $\mu_5 > 0$. Eine optimale Lösung ist durch (λ^{max}, t), mit $t = \frac{x}{\lambda^{max}}$ vorläufig nicht auszuschließen. Einsetzen in die Zielfunktion (6) ergibt: $K\left(\lambda^{max}, t = \frac{x}{\lambda^{max}}\right) = k\left(\lambda^{max}\right)\cdot x$.

Ad c)

Mit $\mu_3 = \mu_4 = \mu_5 = 0$ und (54) ergibt sich:

$$k'(\lambda)\cdot\lambda\cdot t = 0 . \tag{57}$$

Aufgrund von $k'(\lambda^*) = 0$ ist für positive Produktquantitäten $0 < x < x_1 = \lambda^{min} \cdot t^{max}$ eine optimale Lösung durch die Kombination (λ^*, t), mit $t = \frac{x}{\lambda^*}$ vorläufig nicht auszuschließen. Einsetzen in die Zielfunktion (6) ergibt: $K\left(\lambda^*, \frac{x}{\lambda^*}\right) = k(\lambda^*) \cdot x$.

Aus $k(\lambda) > k(\lambda^*)$ für $\lambda \neq \lambda^*$ folgt, dass für positive Produktquantitäten $x \in]0, x_1[$ nur die Kombination (λ^*, t), mit $t = \frac{x}{\lambda^*}$ optimal sein kann. Die Kostenfunktion in diesem Bereich lautet daher:

$$K(x) = k(\lambda^*) \cdot x \qquad \text{für } 0 < x < x_1 \tag{58}$$

3.3 Optimale (λ,t)-Kombinationen für Produktquantitäten $x = x_1$ *und* $x = 0$

Für die Produktquantität $x = x_1 = \lambda^{min} \cdot t^{max}$ gilt weiterhin $\mu_2 = 0$. Es können wiederum drei Fälle unterschieden werden:

$x = \lambda \cdot t$	Intensität λ	Einsatzzeit t	Daraus folgt für die Lagrange-Multiplikatoren (vgl. (19) – (33))
a) $x \in Rd(B)$	$\lambda^{min} = \lambda < \lambda^* < \lambda^{max}$	$t = t^{max}$	$\mu_5 = 0$
b) $x \in Rd(B)$	$\lambda^{min} < \lambda^* < \lambda = \lambda^{max}$	$0 < t < t^{max}$	$\mu_3 = \mu_4 = 0$
c) $x \in \overset{\circ}{B}$	$\lambda^{min} < \lambda < \lambda^{max}$	$0 < t < t^{max}$	$\mu_3 = \mu_4 = \mu_5 = 0$

Eine Analyse dieser (λ,t)-Kombinationen für die Produktquantität $x = x_1 = \lambda^{min} \cdot t^{max}$ ergibt analog zu den Intervallen $x \in]0, x_1[$ und $x \in]x_1, x_2[$ eine optimale Lösung $\lambda = \lambda^*$:

$$K(x) = k(\lambda^*) \cdot x \qquad \text{für } x = x_1. \tag{59}$$

Für x=0 muss t=0 sein. Das Aggregat kommt nicht zum Einsatz. Gemäß (5) sind die Kosten gleich Null für beliebige Intensitäten. Insgesamt ergibt sich für den Bereich $x \in [0, x_2[$ mit (52), (58) und (59) die Kostenfunktion:

$$K(x) = k(\lambda^*) \cdot x \qquad \text{für } 0 \leq x < \lambda^* \cdot t^{max} \tag{60}$$

3.4 *Optimale (λ,t)-Kombinationen für Produktquantitäten* $x \in [x_2, x^{max}]$

Den letzten zu untersuchenden Bereich möglicher Produktquantitäten x stellt das Intervall $x \in [x_2, x^{max}]$ dar, mit $x_2 = \lambda^* \cdot t^{max}$ und $x^{max} = \lambda^{max} \cdot t^{max}$. Werden zunächst die Produktquantitäten $x \in]x_2, x^{max}[$ betrachtet, so ergibt sich für die Intensität: $\lambda^* < \lambda < \lambda^{max}$. Weiterhin gilt $t > 0$ und daher $\mu_2 = \mu_4 = \mu_5 = 0$. Aus (34) folgt:

$$k'(\lambda) \cdot \lambda + k(\lambda) = \mu_1 . \tag{61}$$

$x = \lambda \cdot t$	Intensität λ	Einsatzzeit t	Daraus folgt für die Lagrange-Multiplikatoren (vgl. (19) – (33))
a) $x \in \overset{\circ}{B}$	$\lambda^* < \lambda < \lambda^{max}$	$0 < t < t^{max}$	$\mu_3 = \mu_4 = \mu_5 = 0$
b) $x \in Rd(B)$	$\lambda^* < \lambda < \lambda^{max}$	$t = t^{max}$	$\mu_4 = \mu_5 = 0$

Es können die obigen zwei Fälle unterschieden werden (vgl. Abbildung 3).

Ad a)

Mit $\mu_3 = \mu_4 = \mu_5 = 0$ und (37) ergibt sich:

$$k(\lambda) \cdot \lambda - \mu_1 \cdot \lambda = 0 \tag{62}$$

$$\Rightarrow k(\lambda) = \mu_1 . \tag{63}$$

Einsetzen in (61) liefert:

$$k'(\lambda) \cdot \lambda + k(\lambda) = k(\lambda) \tag{64}$$

$$\Rightarrow k'(\lambda) = 0 \tag{65}$$

$$\Rightarrow \lambda = \lambda^* . \tag{66}$$

Zu einer gegebenen Quantität $\bar{x} \in]x_2, x^{max}[$ ($x_2 = \lambda^* \cdot t^{max}$, $x^{max} = \lambda^{max} \cdot t^{max}$) folgt mit $\bar{x} = \bar{\lambda} \cdot \bar{t}$ und $\bar{\lambda} = \lambda^*$, dass $\bar{t} > t^{max}$ ist, d.h. $(\bar{\lambda}, \bar{t})$ ist nicht zulässig ($(\bar{\lambda}, \bar{t}) \notin B$). Dann ist b) die einzige zulässige und damit kostenminimale Lösung für $\bar{x} = \bar{\lambda} \cdot t^{max}$, die aber nicht die Karush-Kuhn-Tucker-Bedingungen erfüllt.

Ad b)

Im Fall b) liefert Einsetzen von (61) in (37):

$$k(\lambda) \cdot \lambda - \lambda \cdot (k'(\lambda) \cdot \lambda + k(\lambda)) - \mu_3 = 0 \tag{67}$$

$$\Rightarrow -k'(\lambda) \cdot \lambda^2 = \mu_3 \tag{68}$$

$$\Rightarrow \mu_3 < 0 . \tag{69}$$

Für $t = t^{max}$, folgt aus (68): $k'(\lambda) = -\frac{\mu_3}{\lambda^2} > 0 = k'(\lambda^*)$ und damit $\lambda > \lambda^*$. Der Ausdruck $k'(\lambda) = -\frac{\mu_3}{\lambda^2}$ stellt die Grenzkosten der Erhöhung der Intensität λ bei maximaler Einsatzzeit t^{max} dar. Der Lagrange-Multiplikator μ_1 in (61) ist Ausdruck der Grenzkosten einer weiteren Einheit x bei $t = t^{max}$. Für $x \in]x_2, x^{max}[$ erfüllen nur die (λ,t)-Kombination (λ, t^{max}), mit $\lambda = \frac{x}{t^{max}}$ die Karush-Kuhn-Tucker-Bedingungen und sind damit auch kostenminimal. Es ergibt sich die Kostenfunktion:

$$K\left(\frac{x}{t^{max}}, t^{max}\right) = k\left(\frac{x}{t^{max}}\right) \cdot x \quad \text{für } x_2 < x < x^{max}. \tag{70}$$

Abschließend werden die Produktquantitäten $x_2 = \lambda^* \cdot t^{max}$ und $x^{max} = \lambda^{max} \cdot t^{max}$ untersucht. Für $x_2 = \lambda^* \cdot t^{max}$ können zwei Fälle unterschieden werden (vgl. Abbildung 3):

$x = \lambda \cdot t$	Intensität λ	Einsatzzeit t	Daraus folgt für die Lagrange-Multiplikatoren (vgl. (19) – (33))
a) $x \in Rd(B)$	$\lambda = \lambda^* < \lambda^{max}$	$t = t^{max}$	$\mu_4 = \mu_5 = 0$
b) $x \in \overset{\circ}{B}$	$\lambda > \lambda^*$	$t < t^{max}$	$\mu_3 = \mu_5 = 0$

Einsetzen von a) und b) in die Zielfunktion (6) liefert für $\lambda > \lambda^*$, $t < t^{max}$ die Ungleichung: $k(\lambda) \cdot x_2 > k(\lambda^*) \cdot x_2$, d. h. für $x = x_2 = \lambda^* \cdot t^{max}$ ist nur die (λ,t)-Kombination (λ^*, t^{max}) kostenminimal. Daraus folgt:

$$K(x) = k(\lambda^*) \cdot x \qquad \text{für } x = x_2. \tag{71}$$

Für die Produktquantität $x = x^{max}$ schließlich, ist (λ^{max}, t^{max}) die einzige zulässige und damit kostenminimale (λ,t)-Kombination. Insgesamt ergibt sich:

$$K(x) = k\left(\frac{x}{t^{max}}\right) \cdot x \quad \text{für } x \in [x_2, x^{max}], \quad \text{wobei } x_2 = \lambda^* \cdot t^{max}. \tag{72}$$

Die hier vorgenommene Zerlegung von $[0, x^{max}]$ zeigt, dass sich aus den Karush-Kuhn-Tucker-Bedingungen die minimalen Kosten K(x) für alle

$x \in [0, x^{max}]$ eindeutig bestimmen lassen. Anzumerken ist, dass im Falle $\lambda^* = \lambda^{min}$ bzw. $\lambda^* = \lambda^{max}$ die oben durchgeführten Überlegungen zu den gleichen Ergebnissen führen. Der Fall $\lambda^* \notin [\lambda^{min}, \lambda^{max}]$ ist lediglich von theoretischem Interesse, für praktische Anwendungen dagegen irrelevant. In diesem Fall sind die Karush-Kuhn-Tucker-Bedingungen für kein $x \in [0, x^{max}]$ erfüllt, und es ist entweder λ^{min} $(\lambda^* < \lambda^{min})$ oder λ^{max} $(\lambda^* > \lambda^{max})$ kostenminimal aufgrund der vorausgesetzten strengen Konvexität von $k(\lambda)$.

4 Bestimmung der Kostenfunktion

Die folgende Abbildung 4 zeigt für alle realisierbaren Produktquantitäten x die im Rahmen der durchgeführten Analyse gewonnenen kostenminimalen (λ,t)-Kombinationen sowie die sich daraus ergebende Kostenfunktion K(x).

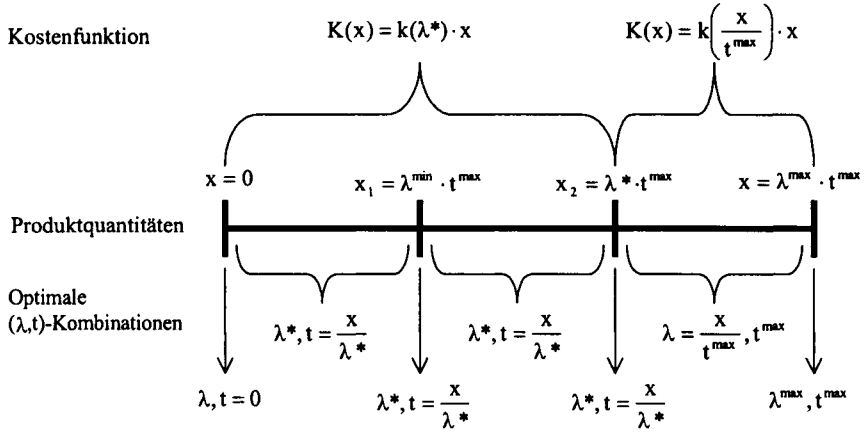


Abbildung 4: Optimale (λ,t)-Kombinationen für alternative Produktquantitäten

Für die Ausbringungsquantitäten $x \in [0, x_2[$ gilt $K(x) = k(\lambda^*) \cdot x$ mit $x = \lambda^* \cdot t$ und $t < t^{max}$. Im Bereich $x \in [x_2, x^{max}]$ gilt $K(x) = k\left(\frac{x}{t^{max}}\right) \cdot x$ mit $t = t^{max}$ und $\lambda \geq \lambda^*$. Gleichwertig ist $K(x) = k(\lambda^*) \cdot x$ mit $x = \lambda^* \cdot t$, $t \leq t^{max}$

und $K(x) = k\left(\frac{x}{t^{max}}\right) \cdot x$ für $t = t^{max}$ und $\lambda > \lambda^*$. Das gestattet die Darstellung der Kostenfunktion K(x) in der in der Literatur üblichen Form, mit:

$$K(x) = \begin{cases} k(\lambda^*) \cdot x & \text{für} \quad 0 \le x \le \lambda^* \cdot t^{max} \\ k\left(\frac{x}{t^{max}}\right) \cdot x & \text{für} \quad \lambda^* \cdot t^{max} < x \le \lambda^{max} \cdot t^{max} \end{cases} \tag{73}$$

Die Analyse der Karush-Kuhn-Tucker-Bedingungen zeigt, dass im Bereich $x \in [0, x_2]$ die Intensität λ^* kostenminimal ist. Für Produktquantitäten $x \in [0, x_2]$ kann eine kostenminimale Steigerung der Produktquantität x nur durch eine Steigerung der Einsatzzeit t gemäß $t = \frac{x}{\lambda^*}$ erreicht werden. Das stellt den bekannten Fall der zeitlichen Anpassung dar (vgl. Gutenberg, 1983). Im Bereich $x \in]x_2, x^{max}]$ werden die Kosten minimal, wenn bei maximaler Einsatzzeit t^{max} die Intensität λ gemäß $\lambda = \frac{x}{t^{max}}$ an alternative Produktquantitäten x angepasst wird (intensitätsmäßige Anpassung).

Für sämtliche Produktquantitäten, die im Planungszeitraum mit der kostenminimalen Intensität erzeugt werden können, erfolgt eine zeitliche Anpassung derart, dass die kostenminimale Intensität gewählt wird und die Einsatzzeit des betrachteten Aggregats an die zu erzeugende Produktquantität angepasst wird. Für Produktquantitäten, die auch bei maximaler Einsatzzeit des Aggregats nicht mit der kostenminimalen Intensität erzeugt werden können, erfolgt eine intensitätsmäßige Anpassung. Es wird die maximale Einsatzzeit des Aggregats gewählt und die Intensität des Aggregats an die zu erzeugende Produktquantität angepasst (vgl. z. B. Adam (1993), Bloech et al. (2001), Schneeweiß (1999) sowie Fandel (1996).

5 Ein Beispiel

Die zuvor ermittelten Ergebnisse werden anhand eines Beispiels illustriert. Gegeben sei ein Aggregat mit der Stückkostenfunktion $k(\lambda) = \frac{1}{20} \cdot \lambda^2 - 2 \cdot \lambda + 24$ (vgl. Abbildung 1). Die Intensität λ und die Einsatzzeit t des Aggregates können in den Intervallen $\lambda \in [10,30]$ und $t \in [0,10]$ stetig variiert werden. Das nichtlineare Optimierungsproblem lautet:

$$\min K(\lambda,t)=\left(\frac{1}{20}\cdot\lambda^2-2\cdot\lambda+24\right)\cdot(\lambda\cdot t) \tag{74}$$

unter den Nebenbedingungen

$$x=\lambda\cdot t \tag{75}$$

$$0\le t\le 10 \tag{76}$$

$$10\le\lambda\le 30\,. \tag{77}$$

Die stückkostenminimale Intensität λ^* ergibt sich mit:

$$k'(\lambda)=\frac{1}{10}\cdot\lambda-2=0\;\Rightarrow\;\lambda^*=20\;\Rightarrow\;k(\lambda^*)=4 \tag{78}$$

Für alternative Produktquantitäten $x\in[0,x^{max}]$, mit $x=\lambda\cdot t$ ergeben sich die optimalen (λ,t)-Kombinationen für (λ^*,t), $\lambda^*=20$, $0\le t\le 10$, d. h. $0\le x\le 200$ (zeitliche Anpassung) und (λ,t^{max}), $20<\lambda\le 30$, $t^{max}=10$, d.h. $200<x\le 300$ (intensitätsmäßige Anpassung). Die optimalen (λ,t)-Kombinationen (Minimalkostenkombinationen) bilden den Minimalkostenpfad, der in Abbildung *5* fett unterlegt ist (Eine zweidimensionale Darstellung dieses Minimalkostenpfades findet sich bereits bei Pack, 1963).

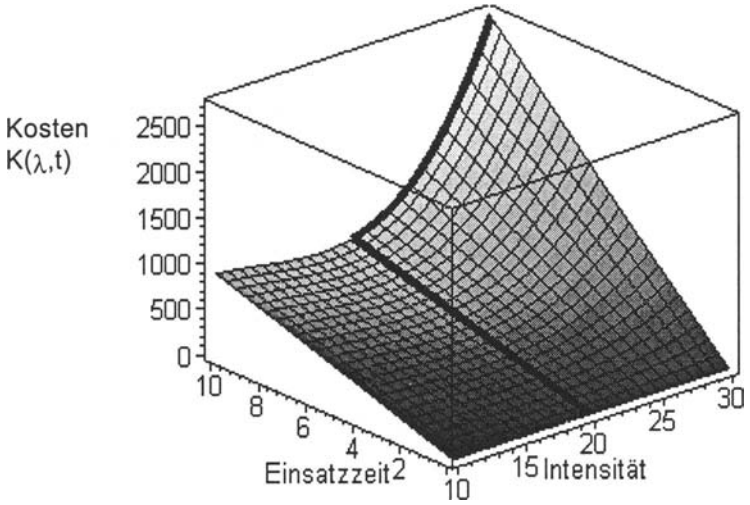


Abbildung 5: Minimalkostenpfad

Die Stückkostenfunktion k(x) und die Kostenfunktion K(x) ergeben sich mit (73) zu:

$$k(x) = \begin{cases} k(20) = 4 & \text{für} \quad 0 \le x \le 200 \\ k\left(\frac{x}{10}\right) = \frac{1}{2000} \cdot x^2 - \frac{1}{5} \cdot x + 24 & \text{für} \quad 200 < x \le 300 \end{cases} \tag{79}$$

$$K(x) = \begin{cases} k(20) \cdot x = 4 \cdot x & \text{für} \quad 0 \le x \le 200 \\ k\left(\frac{x}{10}\right) \cdot x = \frac{1}{2000} \cdot x^3 - \frac{1}{5} \cdot x^2 + 24 \cdot x & \text{für} \quad 200 < x \le 300 \end{cases} \tag{80}$$

Die Verläufe der Stückkostenfunktion (79) sowie der Kostenfunktion (80) sind in Abbildung 6 dargestellt.

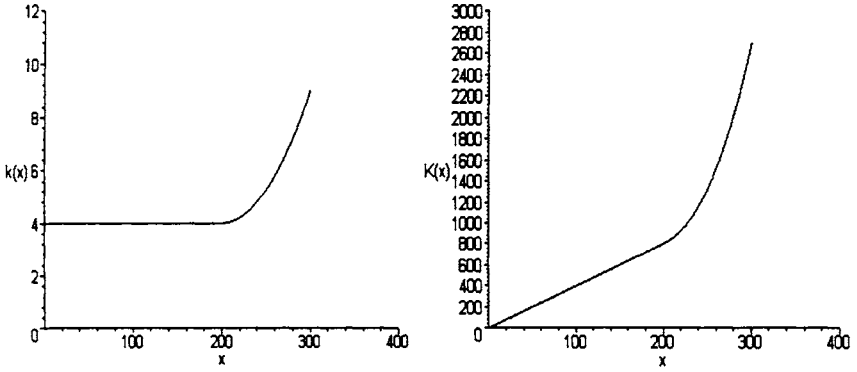


Abbildung 6: Verläufe der Stückkostenfunktion und Kostenfunktion

6 Schlussbetrachtung

Im vorliegenden Beitrag wurde das wohlbekannte Problem der Bestimmung der optimalen Kombination aus zeitlicher und intensitätsmäßiger Anpassung an einem Aggregat erneut aufgegriffen. Im Gutenberg-Produktionsmodell stellen die Intensität λ mit der ein Aggregat betrieben wird und die Einsatzzeit t des Aggregats die Entscheidungsvariablen dar. Mittels der proportionalen Beziehung $x = \lambda \cdot t$ besteht ein substitutionaler Zusammenhang der Entscheidungsvariablen λ und t, wobei die Höhe der Faktoreinsätze und die Produktausbringung durch die Wahl einer zulässigen (λ,t)-Kombination determiniert werden. Zur Bestimmung der kostenminimalen (λ,t)-Kombinationen für alle realisierbaren

Produktquantitäten wurde ein nichtlineares Minimierungsproblem mit den Entscheidungsvariablen λ und t formuliert. Auf Basis der Karush-Kuhn-Tucker-Bedingungen konnte die Minimalkostenkombination im Gutenberg-Produktionsmodell an einem Aggregat mittels zeitlicher und intensitätsmäßiger Anpassung analytisch hergeleitet werden. Aus den Karush-Kuhn-Tucker-Bedingungen lassen sich für jede Produktquantität $x \in [0, x^{max}]$ eindeutig die minimalen Kosten und somit die Kostenfunktion K(x) bestimmen.

Literatur

[1] Adam D (1972) Quantitative und intensitätsmäßige Anpassung mit Intensitäts-Splitting bei mehreren funktionsgleichen, kostenverschiedenen Aggregaten. Zeitschrift für Betriebswirtschaft 42(6):381–400

[2] Adam D (1993) Produktionsmanagement, 7. Auflage. Wiesbaden

[3] Albach H (1962a) Produktionsplanung auf der Grundlage technischer Verbrauchsfunktionen. Arbeitsgemeinschaft für Forschung des Landes Nordrhein-Westfalen, Heft 105. Köln und Opladen, S. 45–96

[4] Albach H (1962b) Zur Verbindung von Produktionstheorie und Investitionstheorie. In: Koch H (Hrsg.): Zur Theorie der Unternehmung, Festschrift zum 65. Geburtstag von Erich Gutenberg. Wiesbaden, S. 137–203

[5] Bloech J, Bogaschewsky R, Götze U, Roland F (2001) Einführung in die Produktion, 4. Auflage. Heidelberg

[6] Botta V (1974) Zur Bestimmung von im Zeitablauf optimalen Leistungsschaltungen. Zeitschrift für Betriebswirtschaft 44(2):89–110

[7] Dellmann R, Nastansky L (1969) Kostenminimale Produktionsplanung bei rein intensitätsmäßiger Anpassung mit differenzierten Intensitätsgraden. Zeitschrift für Betriebswirtschaft 39:239–261

[8] Dinkelbach W, Rosenberg O (2000) Erfolgs- und umweltorientierte Produktionstheorie, 3. Auflage. Berlin u.a.

[9] Domschke W, Drexl A (1995) Einführung in das Operations Research, 3. Auflage. Berlin u.a.

[10] Ellinger Th, Beuermann G, Leisten R (1998) Operations Research – Eine Einführung, 4. Auflage. Berlin u.a.

[11] Fandel G (1996) Produktion I, Produktions- und Kostentheorie, 5. Auflage. Berlin u.a.

[12] Fischer KH (1980) Empirische Anwendungen der Produktionstheorie. Zeitschrift für Betriebswirtschaft 50(3):314–335

[13] Gutenberg E (1983) Grundlagen der Betriebswirtschaftslehre, Erster Band, Die Produktion, 24. Auflage. Berlin u.a.

[14] Hall R (1959) Das Rechnen mit Einflussgrößen im Stahlwerk. Köln und Opladen
[15] Houtman J, Sucky E (2001) Minimalkostenkombination im Gutenberg-Produktionsmodell. WiSt, Wirtschaftswissenschaftliches Studium 30(11): 588–592
[16] Jacob H (1962): Produktionsplanung und Kostentheorie. In: Koch H (Hrsg.) Zur Theorie der Unternehmung, Festschrift zum 65. Geburtstag von Erich Gutenberg. Wiesbaden, S. 205–269
[17] Karrenberg R, Scheer A-W (1970) Ableitung des kostenminimalen Einsatzes von Aggregaten zur Vorbereitung der Optimierung simultaner Planungssysteme. Zeitschrift für Betriebswirtschaft 40(10):689–706
[18] Karush W (1939) Minima of Functions of Several Variables with Inequalities as Side Conditions. MS Thesis, Department of Mathematics, University of Chicago
[19] Kistner K-P (1993) Produktions- und Kostentheorie, 2. Auflage. Heidelberg
[20] Kistner K-P, Luhmer A (1988) Ein dynamisches Modell des Betriebsmitteleinsatzes. Zeitschrift für Betriebswirtschaft 58(1):63–83
[21] Kuhn HW (1991) Nonlinear Programming: A Historical Note. In: Lenstra JK, Rinnooy Kan AHG, Schrijver A (Hrsg.): History of Mathematical Programming – A Collection of Personal Reminiscences. Amsterdam, S. 82–96
[22] Kuhn HW, Tucker AW (1951) Nonlinear Programming. In: Neymann J (Hrsg.): Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, S. 481–492
[23] Lambrecht H-W (1978) Die Optimierung intensitätsmäßiger Anpassungsprozesse. Schriften zur wirtschaftswissenschaftlichen Forschung, Band 128. Meisenheim/Glan
[24] Lücke W (1969) Produktions- und Kostentheorie, Würzburg und Wien
[25] Luhmer A (1975) Maschinelle Produktionsprozesse – Ein Ansatz dynamischer Produktions- und Kostentheorie. Opladen
[26] Ohse D (1998) Quantitative Methoden in der Betriebswirtschaftslehre. München
[27] Pack L (1963) Die Bestimmung der optimalen Leistungsintensität. Zeitschrift für die gesamte Staatswissenschaft 119:1–57
[28] Pack L (1966) Die Ermittlung der kostenminimalen Anpassungsprozeßkombination – ein Anwendungsbeispiel der Dynamischen Programmierung. Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung 18: 466–476
[29] Pressmar DB (1971) Kosten- und Leistungsrechnung im Industriebetrieb. Wiesbaden

[30] Schneeweiß Ch (1999) Einführung in die Produktionswirtschaft, 7. Auflage. Berlin u.a.
[31] Schüler W (1973) Prozeß- und Verfahrensauswahl im einstufigen Einproduktunternehmen. Zeitschrift für Betriebswirtschaft 43(6):435–458
[32] Schüler W (1975) Kostenoptimaler Anlageneinsatz bei mehrstufiger Mehrproduktfertigung. Zeitschrift für Betriebswirtschaft 45(6):392–406
[33] Stepan A (1981) Produktionsfaktor Maschine. Würzburg und Wien
[34] Stepan A, Fischer EO (1996) Betriebswirtschaftliche Optimierung – Einführung in die quantitative Betriebswirtschaftslehre, 5. Auflage. München, Wien
[35] Zimmermann W (1999) Operations Research, Quantitative Methoden zur Entscheidungsvorbereitung, 9. Auflage. München u.a.

Just–in–Time Production of Large Assemblies Using Project Scheduling Models and Methods

Rainer Kolisch

Lehrstuhl für Technische Dienstleistungen und Operations Management
Technische Universität München

1 Introduction

Since the advent of just–in–time driven production planning and control at the Toyota manufacturing plants, the just–in–time paradigm has considered wide–spread consideration within production and operations management (cf., e.g., Schniederjans [22] and Cheng and Podolski [5]). While it was first employed for the high–volume–production of goods only, later there has been considerable research in the area of low–volume, make–to–order manufacturing (cf., e.g., Baker and Scudder [2], Neumann et al. [18], and Rachamadugu [21]). Agrawal et al. [1] considered a practical scheduling problem at Westinghouse ESG, where a number of customer–specific products have to be assembled subject to technological precedence and capacity constraints. The authors developed a MIP–formulation and – in the face of the $\mathcal{NP}$-hardness of the problem – a 'lead time evaluation and scheduling algorithm' with acronym LETSA.

In what follows we will show that the problem as considered by Agrawal et al. [1] – in line with many other well known scheduling problems – can be modeled as classical resource–constrained project scheduling problem (RCPSP). The remainder of the paper is organized as follows: In Section 2 we introduce the assembly scheduling problem and the heuristic proposed by Agrawal et al. [1]. Section 3 provides the resource–constrained project scheduling problem and outlines the serial scheduling algorithm. In Section 4 we show how the assembly scheduling problem can be modeled and solved as RCPSP. Finally, Section 5 outlines the impact of this result.

2 The Assembly Scheduling Problem

The assembly scheduling problem (ASP) can be depicted as follows (we use, with some minor modifications, the original notation proposed by Agrawal et al. [1]): There are $e = 1, \ldots, n_f$ customer–specific products. Each product e has to be assembled until its due date D_e. The assembly–structure of each product e is depicted by its bill of material (BOM). Figure 1 shows the

BOM of two products. Product $e = 1$ with due date $D_1 = 14$ comprises operations $O_1, \ldots, O_6$, product $e = 2$ with due date $D_2 = 10$ comprises operation O_7. Each rectangle depicts a make part and each circle depicts an operation. A make part is manufactured by a sequence of operations. Overall, there are n operations. Each product e has one final–assembly operation $f(e)$ which does not have any downstream operations. All other assembly operations O_i of product e have exactly one downstream operation $d(i)$. This gives for each product an assembly structure of the operations. In the assembly shop there are m different work–centers. In each work–center $W_\mathcal{K}$, $\mathcal{K} = 1, \ldots, m$, there are $f_\mathcal{K}$ functional identical machines. $I_\mathcal{K}$ is the set of operations which have to be processed by one of the machines in work–center $W_\mathcal{K}$. The processing of operation O_i takes t_i periods time. Once started, an operation cannot be preempted. When processed, operation O_i occupies one of the functional identical machines of the work–center where it has to be manufactured. Table 1 and Figure 1 give a two–product example which has been derived by adding product $e = 2$ to the example originally given in Agrawal et al. [1].

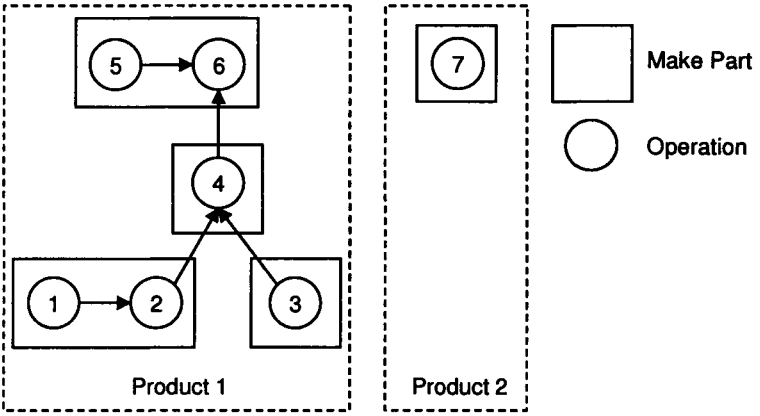


Figure 1: BOM of two products

In order to model the ASP, Agrawal et al. [1] introduce the following decision variables:

$$\delta_{i,j} = \begin{cases} 1, & \text{if operation } O_j \text{ precedes operation } O_i \\ 0, & \text{otherwise} \end{cases}$$

Table 1: Data of the assembly scheduling problem

i	1	2	3	4	5	6	7		
t_i	2	3	3	1	5	2	6		
$I_1 = \{$		2,	3,	4,	5			$\}$	$f_1 = 1$
$I_2 = \{$	1,					6,	7	$\}$	$f_2 = 1$

$$\Delta_{i,k} = \begin{cases} 1, & \text{if operation } O_i \text{ is performed on the } k\text{-th functional} \\ & \text{identical machine of 'its' work–center} \\ 0, & \text{otherwise} \end{cases}$$

$S_i \geq 0$ start time of operation O_i.

$C_i \geq 0$ completion time of operation O_i.

Given a large number M, a MILP for the ASP develops as follows:

$$\begin{aligned}
\text{Max } Z &= \min\{S_i \mid i = 1, \ldots, n\} && (1)\\
\text{s.t.}& \\
S_{d(i)} &\geq C_i \quad (i = 1, \ldots, n) && (2)\\
C_i &= S_i + t_i \quad (i = 1, \ldots, n) && (3)\\
C_{f(e)} &\leq D_e \quad (e = 1, \ldots, n_f) && (4)\\
\delta_{i,j} + \delta_{j,i} &= 1 \quad (i, j \in I_{\mathcal{K}},\ i \neq j,\ \mathcal{K} = 1, \ldots, m) && (5)\\
S_i - C_j &\geq M \cdot (\delta_{i,j} + \Delta_{i,k} + \Delta_{j,k} - 3) && (6)\\
& \qquad (i, j \in I_{\mathcal{K}},\ i \neq j,\ \mathcal{K} = 1, \ldots, m,\ k = 1, \ldots, f_{\mathcal{K}}) \\
\sum_{s=1}^{f_{\mathcal{K}}} \Delta_{i,s} &= 1 \quad (\mathcal{K} = 1, \ldots, m,\ i \in I_{\mathcal{K}}) && (7)\\
S_i &\geq 0 \quad (i = 1, \ldots, n) && (8)\\
C_i &\geq 0 \quad (i = 1, \ldots, n) && (9)\\
\delta_{i,j} &\in \{0,1\} \quad (i, j \in I_{\mathcal{K}},\ i \neq j,\ \mathcal{K} = 1, \ldots, m) && (10)\\
\Delta_{i,k} &\in \{0,1\} \quad (\mathcal{K} = 1, \ldots, m,\ i \in I_{\mathcal{K}},\ k = 1, \ldots, f_{\mathcal{K}}) && (11)
\end{aligned}$$

(1) maximizes the start time of the first–starting operation. Together with constraints (4), which secure that each product is delivered until its due date, this enforces schedules which start operations as late as possible with the aim to obtain production plans with low tied–up capital. This

is the classical pull–strategy of just–in–time manufacturing planning. The constraints (2) enforce the technological precedence relations. (3) links the start and finish time for each operation. Constraints as enforced by scarce capacities of the machines are given by (5) – (7). (5) selects for each pair of operations (O_i, O_j) which have to be processed at the same work–center the orientation $O_i \rightarrow O_j$ or $O_j \rightarrow O_i$. (7) assigns each operation to be processed at a work–center one of its machines. Finally, (6) enforces the orientation between the two operations O_i and O_j if they are processed on the same machine.

Agrawal et al. [1] prove that the ASP is an $\mathcal{NP}$-hard optimization problem. In order to solve large real–world instances, they propose a 'lead time evaluation and scheduling algorithm' (LETSA). Let F denote the set of precedence–feasible operations, i.e. operations that can be scheduled in the current iteration since all their downstream operations have already been scheduled. Then, LETSA can be outlined as follows.

Initialization Set $F = \{O_{f(e)} \mid e = 1, \ldots, n_f\}$

While $F \neq \emptyset$ do

1. Calculate for each path from an operation in F to an operation which is not downstream of any other operation the path–length, i.e. the sum of the durations of all operations on this path.

2. Determine the path with the longest duration (critical path) and its associated operation O_c in F.

3. Set the tentative completion time C_c of operation O_c equal the start time of the downstream operation, or, in case O_c is a final–assembly operation, equal the due date of the associated product.

4. Calculate for each machine $k = 1, \ldots, f_{\mathcal{K}}$ of the work–center where operation O_c has to be processed the latest resource–feasible start–time $S_c(k)$ such that $S_c(k) \leq C_c - t_c$ holds.

5. Select the start time $S_c = \max\{S_c(k) \mid k = 1, \ldots, f_{\mathcal{K}}\}$, set $\Delta_{c,k} = 1$ for the associated machine k, set the final completion time $C_c = S_c + t_c$. Update F by removing operation O_c and adding the operations which are upstream–operations of O_c.

The computational effort of LETSA is as follows. Obviously, steps (2), (3), and (5) are linear. In step (1), due to the fact that the underlying graph is a rooted tree, we have at most n paths which have to be considered.

Calculating the length of one path can be done in n which gives an effort of $\mathcal{O}(n^2)$. Step (4) can be performed for each machine k ($k = 1, \ldots, f_{\mathcal{K}}$) with linear effort of n. Assuming that $n > f_{\mathcal{K}}$ holds, we have $\mathcal{O}(n^2)$ for step (4). Performing the while–loop n times results in an overall computational effort of $\mathcal{O}(n^3)$. Note that this effort can be reduced to $\mathcal{O}(n^2)$ when the computation of the path lengths in step (1) is only performed once and afterwards the path lengths are only updated.

Solving the example problem with LETSA we obtain the schedule given in Figure 2 with an objective function value of 14.

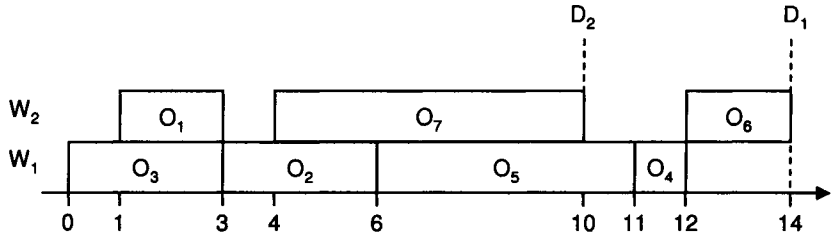


Figure 2: Schedule generated with LETSA

3 The Resource–Constrained Project Scheduling Problem

The resource-constrained project scheduling problem (RCPSP) can be given as follows. A single project consists of $j = 1, \ldots, J$ jobs which are interrelated by precedence and resource constraints. Acyclic precedence constraints impose precedence relations with minimal time lags between pairs of jobs such that job j has a set of immediate predecessors $\mathcal{P}_j$. The minimal time lag between the completion time of each predecessor $h \in \mathcal{P}_j$ and the start of job j is $l_{h,j}$. There are R different types of resources where resource type $r = 1, \ldots, R$ has a capacity of K_r units at any point in time. Job j uses $k_{j,r}$ units of resource type r while being processed. The duration of job j is d_j. W.l.o.g. we assume that job $j = 1$ is the unique dummy start job and that job $j = J$ is the unique dummy end job of the network, i.e. we have $d_1 = d_J = 0$ and $k_{1,r} = k_{J,r} = 0$ for all $r = 1, \ldots, R$. The parameters d_j, $k_{j,r}$, $l_{h,j}$, and K_r are assumed to be continuous, non–negative, and deterministic. The objective of the RCPSP is to find precedence– and resource–feasible completion times for all activities such that the makespan of the project is minimized.

Let F_j denote the finish time of job j. A vector of finish times $(F_1, \ldots, F_J)$ is called a schedule S. Let $\mathcal{A}(t) = \{1, \ldots, J \mid F_j - d_j \leq t < F_j\}$

be the set of jobs which are processed at time instant t. We now can provide the following conceptual decision model (12) – (15) (cf. Christofides et al. [6]).

$$\text{Min} \quad F_J \tag{12}$$

$$\text{s.t.}$$

$$F_h + l_{h,j} \leq F_j - d_j \quad (j = 2, \ldots, J,\ h \in \mathcal{P}_j) \tag{13}$$

$$\sum_{j \in \mathcal{A}(t)} k_{j,r} \leq K_r \quad (r = 1, \ldots, R,\ t \geq 0) \tag{14}$$

$$F_j \geq d_j \quad (j = 1, \ldots, J) \tag{15}$$

The objective function (12) minimizes the finish time of the project end activity and thus the makespan of the project. Constraints (13) enforce the precedence constraints between activities and constraints (14) assure for each resource type r and for each time instant t that the resource demand of the activities which are currently processed does not exceed the available capacity. Finally, (15) define the decision variables. (12) – (15) is a conceptual model since the sets $\mathcal{A}(t)$ are a function of the decision variables F_j. Hence, the model cannot be solved with mixed integer programming techniques. In order to solve the RCPSP with MIP–solvers, one has to employ the $0-1$ formulation of Pritsker et al. [20]. Note that this formulation has two drawbacks. First, the number of binary variables is $J \cdot T$ where T denotes an upper bound of the projects makespan, e.g. $T = \sum_{j=1}^{J} d_j$. Hence, with increasing problem size, the number of binary variables grows quadratically. Second, the model assumes a period, e.g. an hour, a shift, or a day, as minimum time bucket where all processing times are discrete multiples of this standard period. This is not the case for the model (12) – (15) where the durations of activities can be any continuous and non–negative value.

It has been shown by Błażewicz et al. [3] that the RCPSP as a generalization of the classical job shop scheduling problem belongs to the class of $\mathcal{NP}$–hard optimization problems. Therefore, heuristic solution procedures are indispensable when solving large problem instances as they usually appear in practical cases. A building block for most heuristics for solving the RCPSP is the serial scheduling scheme which has been originally proposed by Kelley [12]. The method consists of $J-2$ stages, in each of which one non–dummy job is selected and scheduled at the earliest precedence– and resource–feasible completion time. Associated with stage g are two disjoint job sets. The schedules set $\mathcal{S}_g$ comprises the jobs which have already been scheduled, the eligible set $\mathcal{D}_g$ comprises all jobs which

are eligible for scheduling. Note that the conjunction of $\mathcal{S}_g$ and $\mathcal{D}_g$ does not give the set of all jobs because, generally, there are so–called ineligible jobs, i.e., jobs which have not been scheduled and can not be scheduled at stage g because not all of their predecessors have been scheduled. Let $\tilde{K}_r(t) = K_r - \sum_{j \in \mathcal{A}(t)} k_{j,r}$ be the remaining capacity of resource type r at time instant t and let $\mathcal{F}_g = \{F_j \mid j \in \mathcal{S}_g\}$ be the set of all finish times. Let further be $\mathcal{D}_g = \{j \in \{1, \ldots, J\} \setminus \mathcal{S}_g \mid \mathcal{P}_j \subseteq \mathcal{S}_g\}$ the set of eligible activities. We can now give the following description of the serial scheduling scheme (cf. Kolisch and Hartmann [15]):

Serial Scheduling Scheme

Initialization: $F_1 = 0, \mathcal{S}_1 = \{1\}$
For $g = 2$ to $J - 1$ do

1. Calculate $\mathcal{D}_g, \mathcal{F}_g, \tilde{K}_r(t) \quad (r = 1, \ldots, R; t \in \mathcal{F}_g)$
2. Select one j from $\mathcal{D}_g$
3. $EF_j = \max_{h \in \mathcal{P}_j}(F_h + l_{h,j}) + d_j$
4. $F_j = \min \{t \in [EF_j - d_j, LF_j - d_j] \cap \mathcal{F}_g \mid k_{j,r} \leq \tilde{K}_r(\tau), r = 1, \ldots, R, \tau \in [t, t + d_j[\cap \mathcal{F}_g \Big\} + d_j$
5. $\mathcal{S}_g = \mathcal{S}_{g-1} \cup \{j\}$

$F_J = \max_{h \in \mathcal{P}_J}\{F_h\}$

The initialization assigns the dummy source job $j = 0$ a completion time of 0 and puts it into the partial schedule. At the beginning of each step g, the decision set $\mathcal{D}_g$, the set of finish times $\mathcal{F}_g$, and the remaining capacities $\tilde{K}_r(t)$ at the finish times $t \in \mathcal{F}_g$ are calculated. Afterwards, one job j is selected from the decision set. The finish time of j is calculated by first determining the earliest precedence–feasible finish time EF_j and then calculating the earliest (precedence and–) resource–feasible finish time F_j within $[EF_j, LF_j]$. LF_j denotes the latest finish time as calculated by backward recursion (cf. Elmaghraby [10]) from an upper bound of the projects makespan T.

The time complexity of the serial scheduling scheme as given above is $\mathcal{O}(n^2 \cdot R)$ where n denotes the number of non–dummy jobs, i.e., $n = J - 2$ (cf. Pinson et al. [19]). This results from the fact that we have n stages and within each stage we have an effort of no more than n for steps (1) – (5). When solving a transformed ASP (cf. Section 4), at most one resource type

is used by each job, and hence the time complexity is reduced to $\mathcal{O}(n^2)$. The serial scheduling scheme is a well–known methodology with the following properties. It always generates feasible schedules which are optimal in the case of ample capacity. Also, it has been shown in Kolisch [13] that for scheduling problems with regular performance measure (for a definition of the latter cf. to Sprecher et al. [24]) such as makespan minimization, an optimal solution can always be constructed with this method. The latter is important for metaheuristic approaches which employ the serial scheduling scheme as building block (cf. Kolisch and Hartmann [15]).

4 Modeling and Solving the ASP with the RCPSP

Due to its general applicability the RCPSP has been a platform to model and solve a variety of special scheduling problems. Błażewicz et al. [3] show how the job shop problem can be modeled as RCPSP, Drexl [9] depicts the flow shop problem as RCPSP, Sprecher [23] models the open shop problem and the single product assembly line balancing problem as RCPSP, respectively, whereas Neumann et al. [17] use a generalization of the RCPSP to model batch scheduling problems in the process industry. We will extend this line of research by employing RCPSP–models and methods to solve the assembly scheduling problem (ASP) presented in Section 2.

We begin by showing that the ASP (1) – (11) can be transformed polynomially into an RCSPS (12) – (15). The basic idea is to reverse the assembly structure of each final product and to combine all structures into a single super–project where due dates of final assemblies are modeled by minimal time lags between the dummy start job of the project and the job representing the final assembly operation of each final product, respectively. More formally, a polynomial transformation of the ASP to the RCPSP can be given as follows:

Transformation ASP $\rightarrow$ RCPSP

1. **Initialization**
   Set the number of jobs: $J = n + 2$
   Set the number of resources: $R = m$
   Set the capacity of resource r: $K_r = f_{\mathcal{K}=r}\ (r = 1, \ldots, R)$
   Set the dummy–source: $d_1 = 0$, $k_{1,r} = 0\ (r = 1, \ldots, R)$, $\mathcal{P}_1 = \emptyset$
   Set the dummy sink: $d_J = 0$, $k_{J,r} = 0\ (r = 1, \ldots, R)$,
   $\mathcal{P}_J = \{h \mid h \neq d(i), i = 1, \ldots, n\}, l_{h,J} = 0\ (h \in \mathcal{P}_J)$.

2. **Transformation of the operations**
   For $i = 1, \ldots, n$ do: $j(i) = J - i$, $\mathcal{P}_{j(i)} = d(i)$, $l_{h,j(i)} = 0\ (h \in \mathcal{P}_{j(i)})$

$$k_{j(i),r} = \begin{cases} 1, & \text{if } i \in I_{(\mathcal{K}=r)} \\ 0, & \text{otherwise} \end{cases}$$

3. **Set due date–based minimal time lags from the dummy–source to the final assembly jobs**
   For $e = 1, \ldots, n_f$ do:
   $\mathcal{P}_{j(f(e))} = \{1\}$, $l_{1,j(f(e))} = \max\{D_e \mid e = 1, \ldots, n_f\} - D_e$.

If we apply the transformation steps (1) – (3) to the problem as depicted in Figure 1 and Table 1 we obtain the RCPSP as given in Figure 3 and Table 2 where only non–zero minimal time lags are given. Applying the serial scheduling scheme together with the LST priority rule we obtain the schedule provided in Figure 4 with an objective function value of 14. The schedule represents the same solution as generated with LETSA.

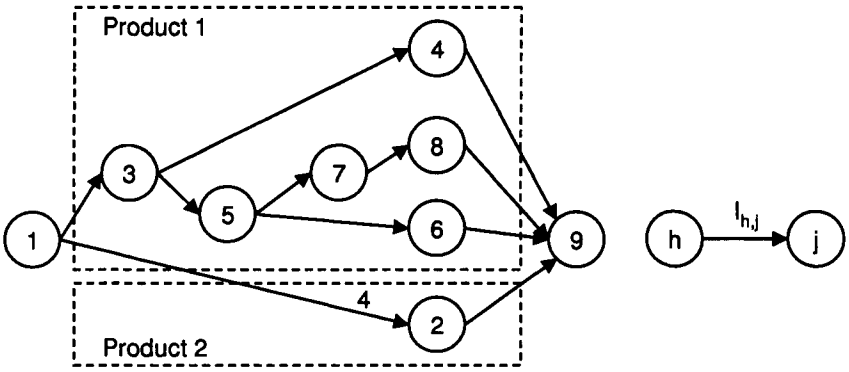


Figure 3: Precedence network of the example problem

Table 2: RCPSP–data

i		7	6	5	4	3	2	1		
$j(i)$	1	2	3	4	5	6	7	8	9	
$d_{j(i)}$	0	6	2	5	1	3	3	2	0	
$k_{j(i),1}$	0	0	0	1	1	1	1	0	0	$K_1 = 1$
$k_{j(i),2}$	0	1	1	0	0	0	0	1	0	$K_2 = 1$

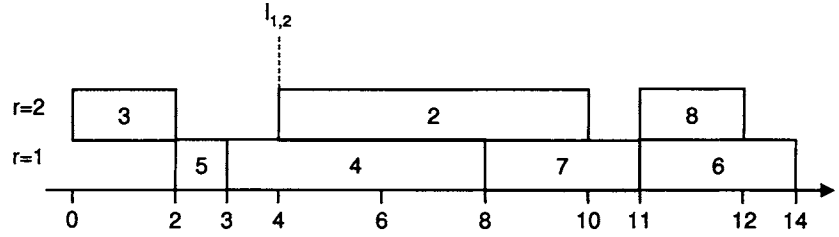


Figure 4: Schedule generated by the serial scheduling algorithm

Theorem. The serial scheduling scheme applied with the LST priority rule (henceforth coined 'serial scheduling algorithm') generates a schedule for the RCPSP which is equivalent to the schedule generated by LETSA for the ASP.

Proof. The serial scheduling algorithm starts selected jobs as early as possible while LETSA starts selected operations as late as possible. Hence, it suffices to show that the sequence of jobs selected by the serial scheduling algorithm is the same as the associated operation–sequence selected by LETSA. This will be the case if the following holds: For iteration $g = 2$ of the serial scheduling algorithm the set of jobs $\mathcal{D}$ equals the set of associate operations F within the first iteration of LETSA (*Equivalence of selection sets*). For each iteration $2, \ldots, J-1$ of the serial scheduling algorithm the selected job j equals the associate operation O_c selected by LETSA in iteration $1, \ldots, n$ (*Equivalence of selected entity*).

Equivalence of selection sets. In iteration $g = 2$ of the serial scheduling algorithm the dummy–source has already been scheduled and we have $\mathcal{D} = \{j \mid \mathcal{P}_j = \{1\}\}$. The jobs in $\mathcal{D}$ represent exactly the final–assembly operations being in the set F at the first iteration of LETSA. This can be seen by looking at the transformation step (3).

Equivalence of selected entity. W.l.o.g. we assume that the data is such that we do not need a tie–breaker for the two heuristics. First, we consider LETSA. Obviously, we can add the artificial operation O_0 to LETSA. O_0 has all the operations as downstream operations which originally do not have an upstream operation. This way we have to calculate for each operation $O_i \in F$ only the longest path $P_{i,0}$ from O_i to O_0. The operation which will be selected is O_c with $P_{c,0} = \max\{P_{i,0} \mid i \in F\}$. The serial scheduling algorithm selects the job j for which $LS_j = \min\{LS_i \mid i \in \mathcal{D}\}$ holds. Denoting with $P_{i,J}$ the longest path from job i to the sink J we can

write instead $T - P_{j,J} = \min\{T - P_{i,J} \mid i \in \mathcal{D}\}$. Since T is constant for all $i \in \mathcal{D}$, we can further write $-P_{j,J} = \min\{-P_{i,J} \mid i \in \mathcal{D}\}$ which equals $P_{j,J} = \max\{P_{i,J} \mid i \in \mathcal{D}\}$.

Resource Assignment. A major difference between the ASP and the RCPSP which has not been addressed so far is the fact that the ASP assigns operations explicitly to machines whereas the RCPSP does resource allocation in an aggregated way. I.e. for each resource type which comprises a number of functional identical machines it is assured that for each time instant the total capacity demand does not exceed the available capacity. In what follows we will outline how, based on a feasible schedule S, a detailed allocation of the capacity units, i.e. the machines, to jobs can be derived for each resource type (cf. Kolisch [14]).

As input data we need the start times $F_j - d_j$ of the jobs in increasing order. Associated with each start time t we calculate the set of jobs which are processed in t, $\mathcal{A}(t)$. The assignment begins in $t(1)$, the first start period, and assigns the non–occupied capacity units $1, \ldots, k_{i,r}$ to job i which is the job in $\mathcal{A}(t(1))$ with the smallest job label. Next, the non–occupied capacity units $k_{i,r}+1, \ldots, k_{i,r}+k_{j,r}$ are assigned to the job j with the second smallest label etc. When capacity units have been assigned to all active jobs in the current period, we proceed to the next start period $t(2) > t(1)$. Jobs which start in a period $t < t(2)$ and are still active in $t(2)$ are considered first. To each of them, the same capacity units as in the prior period are assigned. The remaining capacity units are assigned to the jobs which start in $t(2)$ in the same way as outlined above.

5 Impact and Conclusions

We have shown that the problem of just–in–time scheduling of make–to–order assemblies as treated by Agrawal et al. [1] can be modeled as classical RCPSP. Further, we have proven that solving the RCPSP with the serial scheduling scheme jointly with the LST priority rule generates for any transformed assembly instance the same schedule as LETSA with a computational effort not more than LETSA. This has the following impact. First, it allows an integrated view on just–in–time–steered, low–volume, make–to–order assembly. The RCPSP is well established which facilitates the classification of the assembly scheduling problem w.r.t. to existing project scheduling problems. Second, a lot of research has been and is performed for the RCPSP. A recent overview is given in e.g. Brucker et al. [4], Demeulemeester and Herroelen [8], and Neumann et al. [18]. Hence, there are both, optimal methods and heuristic algorithms (for an overview cf.

Kolisch and Hartmann [16]) with theoretical or experimental performance guarantees available. E.g., due to experiments of Davis and Patterson [7] and Hartmann and Kolisch [11] it is known that on instances with 30 jobs the serial scheduling scheme when applied with the LST priority rule has an average deviation from the optimal solution of approximately 5 %. New meta–heuristics such as the genetic algorithm of Valls et al. [25] derive an average deviation of approximately 0,02 %. This clearly demonstrates the advantage of applying state–of–the–art RCPSP–models and –methods to solve just–in–time make–to–order assembly problems.

References

[1] Agrawal A, Harhalakis G, Minis I, Nagi R (1996) Just–in–time production of large assemblies. IIE Transactions 28:653–667

[2] Baker KR, Scudder GD (1990) Sequencing with earliness and tardiness penalties: A review. Operations Research 38(1):22–36

[3] Błażewicz J, Lenstra JK, Rinnooy Kan AHG (1983) Scheduling subject to resource constraints: Classification and complexity. Discrete Applied Mathematics 5:11–24

[4] Brucker P, Drexl A, Möhring R, Neumann K, Pesch E (1999) Resource–constrained project scheduling: Notation, classification, models, and methods. European Journal of Operational Research 112(1):3–41

[5] Cheng TCE, Podolsky S (1993) Just–in–Time Manufacturing – An Introduction. Chapman & Hall, London

[6] Christofides N, Alvarez-Valdés R, Tamarit JM (1987) Project scheduling with resource constraints: A branch and bound approach. European Journal of Operational Research 29:262–273

[7] Davis EW, Patterson JH (1975) A comparison of heuristic and optimum solutions in resource–constrained project scheduling. Management Science 21:944–955

[8] Demeulemeester EL, Herroelen WS (2002) Project Scheduling – A Research Handbook. Kluwer Academic Publishers, Boston

[9] Drexl A (1990) Fließbandaustaktung, Maschinenbelegung und Kapazitätsplanung in Netzwerken: Ein integrierender Ansatz. Zeitschrift für Betriebswirtschaft 60(1):53–69

[10] Elmaghraby SE (1977) Activity Networks: Project Planning and Control by Network Models. Wiley, New York

[11] Hartmann S, Kolisch R (2000) Experimental evaluation of state–of–the–art heuristics for the resource–constrained project scheduling problem. European Journal of Operational Research 127:394–407

[12] Kelley JE (1963) The critical–path method: Resources planning and scheduling. In: Muth JF, Thompson GL (eds.) Industrial Scheduling. Prentice–Hall, New Jersey, pp. 347–365

[13] Kolisch R (1996) Serial and parallel resource–constrained project scheduling methods revisited: Theory and computation. European Journal of Operational Research 90:320–333

[14] Kolisch R (2001) Make–to–Order Assembly Management. Springer, Berlin

[15] Kolisch R, Hartmann S (1999) Heuristic algorithms for the resource–constrained project scheduling problem: Classification and computational analysis. In: Węglarz J (ed.) Project Scheduling — Recent Models, Algorithms and Applications. Kluwer Academic Publishers, Boston, pp. 147–178

[16] Kolisch R, Hartmann S (2005) Experimental evaluation of heuristics for the resource–constrained project scheduling problem: An update. European Journal of Operational Research, in press.

[17] Neumann K, Schwindt C, Trautmann N (2002) Advanced production scheduling for batch plants in process industries. OR Spektrum 24:251–279

[18] Neumann K, Schwindt C, Zimmermann J (2003) Project Scheduling with Time Windows and Scarce Resources, 2. edition. Springer, Berlin

[19] Pinson E, Prins C, Rullier F (1994) Using tabu search for solving the resource–constrained project scheduling problem. In: Proceedings of the 4. International Workshop on Project Management and Scheduling, Leuven, pp. 102–106

[20] Pritsker AAB, Watters LJ, Wolfe PM (1969) Multiproject scheduling with limited resources: A zero–one programming approach. Management Science 16:93–107

[21] Rachamadugu R (1995) Scheduling jobs with proportionate early/tardy penalties. IIE Transactions 27:679–682

[22] Schniederjans MJ (1993) Topics in Just–in–Time Management. Allyn and Bacon, Boston

[23] Sprecher A (1994) Resource Constrained Project Scheduling — Exact Methods for the Multi–Mode Case, volume 1994 of Lecture Notes in Economics and Mathematical Systems. Springer, Berlin

[24] Sprecher A, Kolisch R, Drexl A (1995) Semi–active, active, and non–delay schedules for the resource–constrained project scheduling problem. European Journal of Operational Research 80:94–102

[25] Valls V, Ballestin F, Quintanilla MS (2003) A hybrid genetic algorithm for the RCPSP. Technical report, Department of Statistics and Operations Research, University of Valencia

A Cyclic Approach to Large-Scale Short-Term Planning of Multipurpose Batch Plants

Christoph Schwindt

Institute of Management and Economics
Clausthal University of Technology

Norbert Trautmann

Departement für Betriebswirtschaftslehre
University of Bern

1 Introduction

In the process industries, final products arise from chemical and physical transformations of materials on processing units. In batch production mode, the total requirements for intermediate and final products are divided into individual batches. To produce a batch, at first the input materials are loaded into a processing unit. Then a transformation process, called a *task*, is performed, and finally the output products are unloaded from the processing unit. Typically, a plant is operated in batch production mode when a large number of different products are processed on multi-purpose equipment. That is why we consider multi-purpose processing units, which can operate different tasks. Symmetrically, a task may be executed on different processing units, in which case the duration of the task may depend on the processing unit used. For a practical example of a multi-purpose batch production plant we refer to the case study presented by Kallrath (2002).

The minimum and maximum filling levels of a processing unit give rise to lower and upper bounds on the batch size. The input and the output proportions of the products consumed or produced, respectively, by a task are either fixed or variable within prescribed bounds. In general, storage facilities of limited capacity are available for stocking raw materials, intermediates, and final products. Some products are perishable and must be consumed immediately after production.

Between consecutive executions of different tasks on a processing unit, a changeover with sequence-dependent duration is necessary. Since the changeover times may be considerably large, the plant is generally configured according to a subset of the required final products. Before processing the next set of final products, the plant has to be reconfigured,

which requires the completion of all operations. In this context, the objective of makespan minimization is particularly important for ensuring a high resource utilization and short customer lead times. Given the primary requirements for final products, the short-term planning problem studied in this paper consists in computing a feasible production schedule with minimum makespan.

Various solution methods for this problem are known from literature. Most of them follow a monolithic approach, which addresses the problem as a whole, starting from a mixed-integer linear programming formulation of the problem. In those models, the time horizon is divided into a given number of time periods, the period length being either fixed (time-indexed formulations, cf. e.g., Kondili et al. 1993) or variable (continuous-time formulations, see e.g., Ierapetritou and Floudas 1998 or Castro et al. 2001). The main disadvantage of the monolithic approaches is that the CPU time requirements for solving real-world problems tend to be prohibitively high (cf. Maravelias and Grossmann 2004). To overcome this difficulty, different heuristics reducing the number of variables have been developed (cf. e.g., Blömer and Günther 1998).

A promising alternative approach is based on the decomposition of the short-term planning problem into interdependent subproblems. Such decomposition methods have, for example, been proposed by Brucker and Hurink (2000), Neumann et al. (2002), and Maravelias and Grossmann (2004). The solution approach developed in what follows is based on the hierarchical decomposition into a batching and a batch-scheduling problem presented in Neumann et al. (2002). Batching provides a set of batches for the intermediate and final products needed to satisfy the primary requirements. Batch scheduling allocates the processing units, intermediates, and storage facilities over time to the processing of the batches arising from the batching step. The batching problem can be formulated as a mixed-integer nonlinear program (MINLP) of moderate size, which can be solved using standard mathematical programming software. For solving the batch-scheduling problem, a truncated branch-and-bound method and a priority-rule-based method have been developed by Neumann et al. (2002) and Schwindt and Trautmann (2004), respectively. Within a reasonable amount of CPU time, good feasible solutions to problem instances with up to 100 batches can be computed with both methods. Recently, Gentner et al. (2004) have proposed a decomposition of the batch-scheduling problem which partitions the set of all batches into a sequence of subsets. The assignment of the batches to the individual subsets is determined stepwise by solving a binary linear program in each iteration. This decomposition method is able to approximatively solve batch-scheduling instances with

up to about 3000 batches in the space of several hours of CPU time (cf. Gentner et al. 2004 and Gentner 2005).

In this paper, we present a cyclic scheduling approach to the short-term planning problem. A preliminary version of this method can be found in Trautmann (2005). The basic idea consists in reducing the size of the batch-scheduling problem by computing a cyclic subschedule, which is executed several times. The set of batches belonging to one cycle is determined by solving an MINLP, which also provides the number of cycles needed to satisfy the primary requirements (*cyclic batching problem*). To guarantee that the resulting batch-scheduling problem remains tractable, we impose an upper bound on the number of batches per cycle. The subschedule is then obtained by scheduling the batches on the processing units subject to material-availability and storage-capacity constraints (*cyclic batch-scheduling problem*). The latter problem is solved using the priority-rule based method proposed in Schwindt and Trautmann (2004).

The remainder of this paper is organized as follows. In Section 2 we formulate the cyclic batching problem as an MINLP and briefly discuss structural issues. Section 3 is devoted to the cyclic batch-scheduling problem and the generation of a complete production schedule by efficiently concatenating copies of the subschedule. In Section 4 we report on results of an experimental performance analysis.

2 Cyclic Batching

Let T be the set of all tasks and let β_τ and ε_τ be the batch size and the number of batches for task $\tau \in \mathrm{T}$. By Π_τ^- and Π_τ^+ we denote the sets of input and output products, respectively, of task $\tau \in \mathrm{T}$. $\Pi_\tau := \Pi_\tau^- \cup \Pi_\tau^+$ is the set of all input and output products of task τ, and $\Pi := \cup_{\tau \in \mathrm{T}} \Pi_\tau$ is the set of all products considered. In addition to β_τ and ε_τ, the (negative) proportions $\alpha_{\tau\pi} < 0$ of all input products $\pi \in \Pi_\tau^-$ and the (positive) proportions $\alpha_{\tau\pi} > 0$ of all output products $\pi \in \Pi_\tau^+$ have to be determined for all tasks $\tau \in \mathrm{T}$ such that

$$\sum_{\pi \in \Pi_\tau^+} \alpha_{\tau\pi} = -\sum_{\pi \in \Pi_\tau^-} \alpha_{\tau\pi} = 1 \quad (\tau \in \mathrm{T}) \tag{1}$$

Proportions $\alpha_{\tau\pi}$ and batch sizes β_τ have to be chosen within prescribed intervals $[\underline{\alpha}_{\tau\pi}, \overline{\alpha}_{\tau\pi}]$ and $[\underline{\beta}_\tau, \overline{\beta}_\tau]$, i.e.,

$$\underline{\alpha}_{\tau\pi} \le \alpha \le \overline{\alpha}_{\tau\pi} \quad (\tau \in \mathrm{T},\ \pi \in \Pi_\tau) \tag{2}$$

$$\underline{\beta}_\tau \le \beta \le \overline{\beta}_\tau \quad (\tau \in \mathrm{T}) \tag{3}$$

Let T_π^- and T_π^+ be the sets of all tasks consuming and producing, respectively, product $\pi \in \Pi$ and let $\Pi^p \subset \Pi$ be the set of perishable products. Then equations

$$\alpha_{\tau\pi}\beta_\tau = -\alpha_{\tau'\pi}\beta_{\tau'} \quad (\pi \in \Pi^p, (\tau, \tau') \in \mathrm{T}_\pi^+ \times \mathrm{T}_\pi^-) \tag{4}$$

ensure that the amount of product $\pi \in \Pi^p$ produced by one batch of some task $\tau \in \mathrm{T}_\pi^+$ can immediately be consumed by any task $\tau' \in \mathrm{T}_\pi^-$ consuming π.

By $\Pi^i \subset \Pi$ we denote the set of intermediates. In order to obtain a cyclic solution, which allows us to execute the same subschedule an arbitrary number of times, the amount of an intermediate π produced within one cycle must be equal to the amount of π consumed, i.e.,

$$\sum_{\tau \in \mathrm{T}_\pi} \alpha_{\tau\pi}\beta_\tau\varepsilon_\tau = 0 \quad (\pi \in \Pi^i) \tag{5}$$

Proportions $\alpha_{\tau\pi}$, batch sizes β_τ, and the numbers of batches ε_τ define the set of batches belonging to one cycle. The number of cycles needed is a decision variable $\nu \in \mathbb{Z}_{\geq 0}$ whose value depends on the given primary requirements for final products. Let $\Pi^f \subset \Pi$ be the set of all final products and let ϱ_π be the primary requirement less the initial stock of product $\pi \in \Pi^f$. The final inventory of product π then equals $\nu \sum_{\tau \in \mathrm{T}} \alpha_{\tau\pi}\beta_\tau\varepsilon_\tau$. This amount must be sufficiently large to match the requirements ϱ_π for π, i.e.,

$$\nu \sum_{\tau \in \mathrm{T}_\pi} \alpha_{\tau\pi}\beta_\tau\varepsilon_\tau \geq \varrho_\pi \quad (\pi \in \Pi^f) \tag{6}$$

In addition, the number of batches within one cycle must not exceed the prescribed upper bound $\bar{\varepsilon}$, i.e.,

$$\sum_{\tau \in \mathrm{T}_\pi} \varepsilon_\tau \leq \bar{\varepsilon} \tag{7}$$

Finally, let p_τ be the mean processing time of task τ on the alternative processing units. To minimize the workload to be scheduled in the batch-scheduling step, the objective function is chosen to be the total mean processing time $\nu \sum_{\tau \in \mathrm{T}} p_\tau\varepsilon_\tau$. In sum, the cyclic batching problem reads

$$\text{(C-BP)} \quad \begin{cases} \text{Minimize} & \nu \sum_{\tau \in \mathrm{T}} p_\tau\varepsilon_\tau \\ \text{subject to} & (1) \text{ to } (7) \\ & \varepsilon_\tau \in \mathbb{Z}_{\geq 0} \quad (\tau \in \mathrm{T}) \\ & \nu \in \mathbb{Z}_{\geq 0} \end{cases}$$

For a given value of ν, problem (C-BP) can be transformed into a mixed-binary linear program with binary decision variables θ_τ^μ ($\tau \in \mathrm{T}$, $\mu = 1, \ldots, \overline{\varepsilon}$) being equal to one exactly if $\varepsilon_\tau \geq \mu$ and continuous decision variables $\xi_{\tau\pi}^\mu$ ($\tau \in \mathrm{T}$, $\pi \in \Pi_\tau$, $\mu = 1, \ldots, \overline{\varepsilon}$) with $\xi_{\tau\pi}^\mu = \alpha_{\tau\pi}\beta_\tau$ if $\theta_\tau^\mu = 1$ and $\xi_{\tau\pi}^\mu = 0$, otherwise (see Neumann et al. 2002 for details). Now suppose without loss of generality that $\max_{\pi \in \Pi^f} \rho_\pi > 0$. Due to $\nu > 0$ for any feasible solution to (C-BP), inequality (6) can be rewritten as

$$\frac{\rho_\pi}{\nu} - \sum_{\tau \in \mathrm{T}_\pi} \sum_{\mu=1}^{\overline{\varepsilon}} \xi_{\tau\pi}^\mu \leq 0 \quad (\pi \in \Pi^f) \tag{8}$$

Let ($\overline{\text{C-BP}}$) denote the continuous relaxation of the reformulated batching problem with decision variables θ_τ^μ and $\xi_{\tau\mu}^\mu$. Since for each $\pi \in \Pi^f$ the left-hand side of (8) is a convex function and because all remaining constraints of ($\overline{\text{C-BP}}$) are linear, the feasible region of ($\overline{\text{C-BP}}$) is a convex set. Moreover, the objective function $\nu \sum_{\tau \in \mathrm{T}} p_\tau \sum_{\mu=1}^{\overline{\varepsilon}} \theta_\tau^\mu$ of ($\overline{\text{C-BP}}$) is increasing on the feasible region.

3 Cyclic Batch-Scheduling and Concatenation

In this section we explain our method for solving the batch-scheduling problem. In Subsections 3.1 and 3.2, where we closely follow the presentation of Schwindt and Trautmann (2004), we are concerned with the scheduling of the batches belonging to one cycle. In Subsection 3.3 we show how a complete production schedule for the execution of the ν cycles can be efficiently constructed from the cyclic subschedule.

3.1 Statement of the Cyclic Batch-Scheduling Problem

Recall that solving the batching problem has provided us with the set of batches belonging to one cycle. For what follows, the processing of a batch on a processing unit is called an *operation*. Suppose that $n = \sum_{\tau \in \mathrm{T}} \varepsilon_\tau$ operations $1, \ldots, n$ have to be scheduled. For notational convenience we introduce two fictitious operations 0 and $n+1$ representing the production start and the production end, respectively. $\widetilde{V} := \{1, \ldots, n\}$ is the set of all real operations, and $V := \widetilde{V} \cup \{0, n+1\}$ is the set of all operations. Let $S_i \geq 0$ be the *start time* sought of operation i. Then S_{n+1} coincides with the production makespan, and vector $S = (S_i)_{i \in V}$ with $S_0 = 0$ is called a *schedule*.

Each **processing unit** can be viewed as a unit-capacity *renewable resource with changeovers* (cf. Neumann et al. 2003, Sect. 2.14). Let $\mathcal{R}^\rho$ be the set of all renewable resources and $\mathcal{R}_i^\rho$ be the set of those alternative renewable resources on which operation i can be carried out. For $i \in \widetilde{V}$ and $k \in \mathcal{R}_i^\rho$, the binary decision variable x_{ik} indicates whether or not resource k processes operation i ($x_{ik} = 1$ or $x_{ik} = 0$, respectively). Each real operation i must be executed on exactly one processing unit, i.e.,

$$\sum_{k \in \mathcal{R}_i^\rho} x_{ik} = 1 \quad (i \in \widetilde{V}) \tag{9}$$

Vector $x = (x_{ik})_{i \in \widetilde{V}, k \in \mathcal{R}_i^\rho}$ is called an *assignment* of operations i to processing units k. By $p_i(x)$ and $c_{ij}(x)$ we denote the processing time of operation i and the changeover time from operation i to operation j given assignment x, where we suppose that $p_0(x) = p_{n+1}(x) = 0$ and $c_{0i}(x) = c_{i(n+1)}(x) = 0$ for all $i \in \widetilde{V}$.

Given a resource $k \in \mathcal{R}^\rho$, a schedule S and an assignment x, let $P_k(S, x)$ designate the set of all pairs (i, j) such that $i \neq j$, $x_{ik} = x_{jk} = 1$, and $S_i \leq S_j$. Schedule S is called *process-feasible* with respect to assignment x if no two operations i and j overlap on a processing unit, i.e.,

$$S_j \geq S_i + p_i(x) + c_{ij}(x) \quad (k \in \mathcal{R}^\rho,\ (i, j) \in P_k(S, x)) \tag{10}$$

Now we turn to the **storage facilities**, which can be modeled as so-called *cumulative resources* (cf. Neumann and Schwindt 2002). For each storage facility we introduce one cumulative resource keeping its inventory. Let $\mathcal{R}^\gamma$ be the set of all cumulative resources. For each $k \in \mathcal{R}^\gamma$, a minimum inventory $\underline{R}_k$ (safety stock) and a maximum inventory $\overline{R}_k$ (storage capacity) are given. Assuming that each product π is stocked in a dedicated storage facility k and that no safety stocks are prescribed we obtain $\underline{R}_k = 0$ and $\overline{R}_k = \sigma_\pi$ for all $k \in \mathcal{R}^\gamma$, where σ_π is the storage capacity for product π (with $\sigma_\pi = 0$ if $\pi \in \Pi^p$). Each operation $i \in V$ has a demand r_{ik} for resource $k \in \mathcal{R}^\gamma$. If $r_{ik} > 0$, the inventory of resource k is replenished by r_{ik} units at time $S_i + p_i(x)$. If $r_{ik} < 0$, the inventory is depleted by $-r_{ik}$ units at time S_i. r_{0k} represents the initial stock level of resource k. Suppose that operation i corresponds to an execution of task τ and that resource k is dedicated to product π. The demand of operation i for resource k then is $r_{ik} = \alpha_{\tau\pi}\beta_\tau$.

Let $V_k^+ := \{i \in V \mid r_{ik} > 0\}$ and $V_k^- := \{i \in V \mid r_{ik} < 0\}$ be the sets of operations replenishing and depleting, respectively, the inventory of resource $k \in \mathcal{R}^\gamma$. Schedule S is said to be *storage-feasible* with respect to

assignment x if

$$\underline{R}_k \leq \sum_{i \in V_k^+ : S_i + p_i(x) \leq t} r_{ik} + \sum_{i \in V_k^- : S_i \leq t} r_{ik} \leq \overline{R}_k \qquad (k \in \mathcal{R}^\gamma,\ t \geq 0) \quad (11)$$

Usually, **temporal constraints** of the type $S_j - S_i \geq \delta_{ij}(x)$ for $(i,j) \in E$ with $E \subseteq V \times V$ have to be taken into account as well. The right-hand side $\delta_{ij}(x)$ is a *minimum time lag* between the start of operations i and j. If $\delta_{ij}(x) < 0$, then $-\delta_{ij}(x)$ can be interpreted as a *maximum time lag* between the start of operations j and i. In case of $\delta_{ij}(x) = p_i(x)$, the corresponding temporal constraint is referred to as a *precedence constraint*, and a time lag $\delta_{0i}(x)$ is called a *release date* for operation i. For each operation $i \in \widetilde{V}$ we set $\delta_{0i}(x) := 0$ and $\delta_{i(n+1)}(x) := p_i(x)$. Further time lags may be generated by applying constraint propagation techniques detecting temporal constraints that are satisfied by at least one optimal solution to the batch-scheduling problem. For example, for two operations i, j with $i < j$ belonging to the same task τ we can introduce the time lag $\delta_{ij}(x) = 0$, without loss of optimality.

Based on time lags $\delta_{ij}(x)$ for $(i,j) \in E$ we can compute *distances* $d_{ij}(x)$ between any two operations $i, j \in V$. Distances $d_{ij}(x)$ coincide with the minimum time lags between operations i and j that are implied by the prescribed time lags (see e.g., Neumann et al. 2003, Sect. 1.3). Given an assignment x, a schedule S satisfying

$$S_j \geq S_i + \delta_{ij}(x) \qquad ((i,j) \in E) \quad (12)$$

is called *time-feasible* with respect to x.

A schedule which is time-, process-, and storage-feasible with respect to a given assignment x is called *feasible* with respect to x. A pair (S, x) is a feasible solution to the cyclic batch-scheduling problem if x is an assignment and S is a feasible schedule with respect to x. The cyclic batch-scheduling problem consists in finding a feasible solution (S, x) with minimum makespan S_{n+1}, i.e.,

$$\text{(C-BSP)} \quad \begin{cases} \text{Minimize} & S_{n+1} \\ \text{subject to} & (9) \text{ to } (12) \\ & S_0 = 0 \\ & x_{ik} \in \{0,1\} \quad (i \in \widetilde{V},\ k \in \mathcal{R}_i^\rho) \end{cases}$$

3.2 Priority-Rule Based Method

The basic idea of the priority-rule based solution method is as follows. At first, we choose an assignment x of operations to processing units, where

we balance the workload to be processed on alternative processing units by using a simple greedy heuristic. The method then consists of two phases. During the first phase, we relax the storage-capacity constraints. Using a serial schedule-generation scheme, the operations are iteratively scheduled on the processing units in such a way that the inventory does not fall below the safety stock at any point in time. Based on the resulting schedule, precedence constraints between replenishing and depleting operations are introduced according to a FIFO strategy. Those precedence constraints ensure that the material-availability constraints are always satisfied. In the second phase, which again applies the serial schedule-generation scheme, the operations are scheduled subject to the storage-capacity and the precedence constraints introduced.

In the remainder of this subsection we explain the schedule-generation scheme of the first phase in more detail. We then briefly sketch the modifications needed for using the scheme in the second phase. Since assignment x has been fixed, we omit x in the notation of the processing times and time lags.

Let $Pred(j)$ be the set of predecessors of node j with respect to the strict order $\{(i,j) \in V \times V \mid d_{ij} \geq 0 \text{ and } d_{ji} < 0\}$. It holds that $i \in Pred(j)$ precisely if operation i must be started no later than operation j but conversely, operation j may be started after operation i. Moreover, let $\mathcal{C}$ be the *completed set* of operations i already scheduled in prior iterations and let $S^{\mathcal{C}} := (S_i)_{i \in \mathcal{C}}$ be the *partial schedule* constructed. We say that an operation $j \notin \mathcal{C}$ is *eligible* for being scheduled if (i) all of its predecessors have been scheduled, i.e., $Pred(j) \subseteq \mathcal{C}$ and (ii), there is no cumulative resource k whose inventory level falls below the safety stock after the completion of all operations from set $\mathcal{C} \cup \{j\}$, i.e., $r_k(S^{\mathcal{C}}, \infty) + r_{jk} \geq \underline{R}_k$ for all $k \in \mathcal{R}^\gamma$.

The procedure is now as follows (see Algorithm 1). At first, we initialize the earliest and latest start times ES_i and LS_i for all $i \in V$. In each iteration of the schedule-generation scheme we then determine the set $\mathcal{E}$ of eligible operations j, select one operation $j^* \in \mathcal{E}$ according to priority indices $\pi(j)$, determine the earliest feasible start time $t^* \geq ES_{j^*}$ for operation j^*, schedule j^* at time t^*, and update the earliest and latest start times of the operations i not yet scheduled. Starting with partial schedule $S^{\mathcal{C}}$ where $\mathcal{C} = \{0\}$ and $S_0 = 0$ we perform those steps until all operations have been scheduled, i.e., until $\mathcal{C} = V$.

Sometimes it may happen that due to maximum time lags between scheduled operations $i \in \mathcal{C}$ and the operation j^* selected, the latest start time LS_{j^*} of j^* is strictly smaller than time t^*. Then no feasible start time can be found for operation j^*, and $S^{\mathcal{C}}$ cannot be extended to a feasible schedule. In this case, we perform the following unscheduling step. At

first, we determine the set $\mathcal{U} = \{i \in \mathcal{C} \mid LS_j^* = S_i - d_{j^*i}\}$ of all operations i that must be delayed for being able to schedule j^* at time t^*. Then, we increase the earliest start times of operations i from set $\mathcal{U}$ by adding the release dates $\delta_{0i} = S_i + t^* - LS_{j^*}$, update the distances d_{ij} accordingly, and restart the scheduling procedure. In the implementation shown in Algorithm 1, the number u of unscheduling steps is limited by some upper bound $\overline{u}$.

Algorithm 1: Schedule-generation scheme of phase 1

$u := 0$;
2: $S_0 := 0$, $\mathcal{C} := \{0\}$;
for all $i \in V$ **do** (* initialize ES_i and LS_i *)
  $ES_i := d_{0i}$, $LS_i := -d_{i0}$;
while $\mathcal{C} \neq V$ **do**
  $\mathcal{E} := \{j \in V \setminus \mathcal{C} \mid Pred(j) \subseteq \mathcal{C},\ r_k(S^{\mathcal{C}}, \infty) + r_{jk} \geq \underline{R}_k \text{ for all } k \in \mathcal{R}^\gamma\}$;
  $j^* := \min\{j \in \mathcal{E} \mid \pi(j) = \text{ext}_{h \in \mathcal{E}} \pi(h)\}$;
  $t' := \min\{t \geq ES_{j^*} \mid r_k(S^{\mathcal{C}}, \tau) + r_{j^*k} \geq \underline{R}_k \text{ for all } k \in \mathcal{R}^\gamma,\ \tau \geq t\}$;
  $t^* := \min\{S_{j^*} \geq t' \mid S^{\mathcal{C} \cup \{j^*\}} \text{ is process-feasible}\}$;
  if $t^* > LS_{j^*}$ **then** (* unschedule and restart *)
    $u := u + 1$;
    if $u > \overline{u}$ **then** terminate;
    $\mathcal{U} := \{i \in \mathcal{C} \mid LS_{j^*} = S_i - d_{j^*i}\}$;
    for all $i \in \mathcal{U}$ **do** $d_{0i} := S_i + t^* - LS_{j^*}$;
    update distances d_{ij} for all $i, j \in V$ and **goto** line 2;
  else (* schedule j^* at time t^* *)
    $S_{j^*} := t^*$, $\mathcal{C} := \mathcal{C} \cup \{j^*\}$;
    for all $j \in V \setminus \mathcal{C}$ **do** (* update ES_j and LS_j *)
      $ES_j := \max(ES_j, S_{j^*} + d_{j^*j})$;
      $LS_j := \min(LS_j, S_{j^*} - d_{jj^*})$;
return S;

After having obtained a time- and process-feasible schedule satisfying the material-availability constraints, we link producing and consuming operations according to a FIFO strategy. This means that for each $k \in \mathcal{R}^\gamma$ we iterate the replenishing operations $i \in V_k^+$ according to nondecreasing completion times $S_i + p_i$ and allot the corresponding r_{ik} units to depleting operations $j \in V_k^-$ in the order of nondecreasing start times S_j. For each pair $(i, j) \in V_k^+ \times V_k^-$ for which j consumes units produced by i, we introduce a precedence constraint between i and j by setting $\delta_{ij} := \max(\delta_{ij}, p_i)$. Subsequently, we update the distances d_{ij} and proceed with the second phase of our procedure.

When during the second phase we deal with storage-capacity instead of material-availability constraints, we define the eligible set to be $\mathcal{E} := \{j \in V \setminus \mathcal{C} \mid Pred(j) \subseteq \mathcal{C},\ r_k(S^{\mathcal{C}}, \infty) + r_{jk} \leq \overline{R}_k \text{ for all } k \in \mathcal{R}^\gamma\}$. In the definition of $\mathcal{E}$, we use the predecessor sets $Pred(j)$ from the first phase in order to allow the scheduling of depleting operations before the replenishing operations allotted to them have been added to the partial schedule. The earliest storage-feasible start time of operation j^* is now given by $t' := \min\{t \geq ES_{j^*} \mid r_k(S^{\mathcal{C}}, \tau) + r_{jk} \leq \overline{R}_k \text{ for all } k \in \mathcal{R}^\gamma,\ \tau \geq t + p_{j^*}\}$. In this way, we ensure that any partial schedule $S^{\mathcal{C}}$ is feasible.

3.3 Concatenation

For generating the complete production schedule we proceed as follows. The (sub-)schedule S computed by the priority-rule based method defines precedence relationships between the operations i, j of one cycle being executed on the same processing unit or producing and consuming the same product. Those precedence relationships are translated into time lags δ_{ij}, which ensure that no resource conflict can occur when left- or right-shifting the operations. More precisely, for each pair of operations (i, j) with $S_j \geq S_i + p_i(x) + c_{ij}(x)$ and $x_{ik} = x_{jk} = 1$ for some $k \in \mathcal{R}^\rho$ we introduce the time lag $\delta_{ij} = p_i(x) + c_{ij}(x)$ preventing the overlapping of i and j. For pairs (i, j) with $S_j \geq S_i + p_i(x)$ and $r_{ik} > 0$, $r_{jk} < 0$ for some $k \in \mathcal{R}^\gamma$, the time lags $\delta_{ij} = p_i(x)$ guarantee the availability of the intermediate stocked in resource k. Eventually, we add the time lag $\delta_{ij} = -p_j(x)$ for each pair (i, j) with $S_j + p_j(x) \geq S_i$ and $r_{ik} < 0$, $r_{jk} > 0$ for some $k \in \mathcal{R}^\gamma$, to avoid an excess of the storage capacity of resource k. Moreover, the completion time of the last operation that is processed on a processing unit defines a release date δ_{0i} for the changeover to the first operation i on that unit in the next execution of the subschedule. Analogously, the last change in the inventory level of an intermediate gives rise to a release date δ_{0i} for the first operation i that subsequently produces or consumes that intermediate.

The start and completion times of the operations in the first cycle equal those of subschedule S. For computing the start and completion times of the operations in the next cycle, we solve a temporal scheduling problem which consists in computing an earliest schedule for those operations subject to the precedence relationships between and the release dates for the operations. As it is well-known, this temporal scheduling problem can be solved efficiently by longest path calculations. By iteratively concatenating the ν subschedule copies in this way, we finally obtain the production schedule sought.

4 Experimental Performance Analysis

We have compared the new heuristic to the decomposition approach by Gentner et al. (2004). The performance analysis was based on a test set introduced in Gentner (2005), which has been constructed by varying the primary requirements for final products in the case study of Kallrath (2002). For each instance, we have computed a solution to the cyclic batching problem using Frontline Systems' Solver package. The subschedules have been computed by a randomized multi-pass version of the priority-rule based method presented in Section 3. The tests have been performed on an 800 MHz Pentium III personal computer. The results for the decomposition approach have been reported in Gentner (2005) and refer to a 1400 MHz Pentium IV personal computer.

The results obtained for the 13 problem instances are shown in Table 1. For each problem instance the new method is able to find a markedly better solution. Especially for larger problem instances, the required CPU time is significantly smaller than for the decomposition approach. Having prescribed an upper bound of $\overline{\varepsilon} = 100$ batches, about 75 seconds are required for solving the cyclic batching problem. The priority-rule based method has been stopped after 15 seconds of CPU time. The concatenation has always required less than one second of CPU time.

Table 1: Computational results

Instance	Gentner (2005) Makespan	Gentner (2005) t_{cpu} [s]	This paper # batches	This paper Makespan	This paper t_{cpu} [s]
WeKa0_1	352	38	176	264	89
WeKa0_2	474	53	264	390	89
WeKa0_3	612	120	352	516	89
WeKa0_4	738	209	440	642	89
WeKa0_5	906	178	528	768	89
WeKa0_6	1046	215	616	894	90
WeKa0_7	1199	323	704	1020	91
WeKa0_8	1334	281	792	1146	91
WeKa0_9	1548	399	880	1272	91
WeKa0_10	1740	431	968	1398	91
WeKa0_15	2123	644	1408	2028	91
WeKa0_20	2899	1500	1848	2658	91
WeKa0_30	4416	5235	2728	3918	92

5 Conclusions

We have considered a short-term planning problem of batch production in the process industries. We have proposed a heuristic solution method for solving large-scale instances of this problem, consisting of the three steps cyclic batching, cyclic batch-scheduling, and concatenation. Because each of those steps has to be performed only once, the computational requirements of the heuristic are moderate. In an experimental performance analysis, we have shown that the new method clearly outperforms the best solution approach known from literature.

An important area of future research is, for example, the design of efficient solution methods for the case of continuous production, where the production and consumption rates of products are decision variables as well. Moreover, procedures for robust and reactive short-term planning should be developed, which are able to cope with uncertainty with respect to planning data like primary requirements, processing times, or yields.

References

[1] Blömer F, Günther HO (1998) Scheduling of a multi-product batch process in the chemical industry. Computers in Industry 36:245–259

[2] Brucker P, Hurink J (2000) Solving a chemical batch scheduling problem by local search. Annals of Operations Research 96:17–36

[3] Castro P, Barbosa-Póvoa AP, Matos H (2001) An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. Industrial & Engineering Chemistry Research 40:2059–2068

[4] Gentner K (2005) Dekompositionsverfahren für die ressourcenbeschränkte Projektplanung. Shaker Verlag, Aachen

[5] Gentner K, Neumann K, Schwindt C, Trautmann N (2004) Batch production scheduling in the process industries. In: Leung JYT (ed.) Handbook of Scheduling: Algorithms, Models, and Performance Analysis. CRC Press, Boca Raton, pp. 48/1–48/21

[6] Ierapetritou MG, Floudas CA (1998) Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes, Industrial & Engineering Chemistry Research 37:4341–4359

[7] Kallrath J (2002) Planning and scheduling in the process industry. OR Spectrum 24:219–250

[8] Kondili E, Pantelides CC, Sargent RWH (1993) A general algorithm for short-term scheduling of batch operations: I. MILP Formulation. Computers and Chemical Engineering 17:211–227

[9] Maravelias CT, Grossmann IE (2004) A hybrid MILP/CP decomposition approach for the continuous time scheduling of multipurpose batch plants. Computers and Chemical Engineering 28:1921–1949

[10] Neumann K, Schwindt C (2002) Project scheduling with inventory constraints. Mathematical Methods of Operations Research 56:513–533

[11] Neumann K, Schwindt C, Trautmann N (2002) Advanced production scheduling for batch plants in process industries. OR Spectrum 24:251–279

[12] Neumann K, Schwindt C, Zimmermann J (2003) Project Scheduling with Time Windows and Scarce Resources. Springer, Berlin

[13] Schwindt C, Trautmann N (2004) A priority-rule based method for batch production scheduling in the process industries. In: Ahr D, Fahrion R, Oswald M, Reinelt G (eds) Operations Research Proceedings 2003. Springer, Berlin, pp. 111–118

[14] Trautmann N (2005) Operative Planung der Chargenproduktion. Deutscher Universitäts-Verlag, Wiesbaden

Models and Methods for Workforce Scheduling: An Application to Casino Operations

Christoph Stark and Jürgen Zimmermann

Department of Operations Research
Clausthal University of Technology

Martin Morlock

Department of Economics
University of Gießen

1 Introduction

Labor costs become more and more important for a company's success. This holds especially true for high-wage countries like Germany and for labor-intensive service industries like such as call centers or casinos. Efficient planning and allocation of personnel resources is therefore essential and quite a lot of models and methods for this so-called "workforce scheduling problem" have been developed thus far. Nevertheless, real problems are very different, complex, and specific. There exists no standard procedure to find optimal – or at least suboptimal – solutions (if the complexity of the problem is too great for exact algorithms). The formulation of the problem has to take into account the unique situation of a company and subsequently an appropriate solution method has to be adapted.

A very specific situation appears in casinos. These often glamorous gambling places generate an atmosphere of luxury and luck, and in this context, service and organization have to be perfect. All tasks should be performed flawlessly, producing considerable stress for the employees. Difficult in this context is that very different tasks have to be scheduled, and for these tasks quite different qualifications are necessary. The most popular betting games in a casino are roulette, black jack, poker, and baccarat, each of these games affording special instruction courses for the employees involved. Roulette for example requires a team of four or five employees in different positions with different qualifications. In addition to the staff for the betting games, highly qualified personnel for supervision is necessary.

Besides the problem of coordinating tasks and employees with different qualifications and with respect to maximal working time, other characteristics have to be considered. A lot of difficulties to construct an optimal or suitable workforce schedule stem from the fact that on the one hand

casinos are closed only a few days per year and normally opening hours are more than 12 hours every day. On the other hand the enormous concentration required by the employees to do their job call for restricted working hours and regular idle times. These restrictions are not only due to reglementations by law. Also, there should be a certain regularity and balance between early and late shifts. Otherwise, indisposition or illness may occur.

The construction of good workforce schedules is of great interest. In Germany, we have 61 casinos with a cross gambling profit of round about one billion Euros. Up to now, workforce scheduling in casinos is mostly done without support of quantitative methods. This is very time consuming and not at all satisfactory. No doubt, simple schemes often used in combination with the planner's long experience and skill have the advantage of being understood and accepted. Computer programs sometimes have the lack of generating solutions which are optimal in the framework of the model but maladjusted to reality, because the model neglects some minor details. To overcome these difficulties, quantitative models and methods, which are lucid and flexible thus offering the possibility of generating alternative solutions, should be applied. The approach presented in the sequel, which is based on previous work displayed in [6] and [7], has a focus to both aspects.

2 Problem Formulation

In what follows, we first describe some typical restrictions of workforce scheduling problems in casinos (cf. Section 2.1). In Section 2.2 we present a network-based representation for the underlying workforce scheduling problem, which is vaguely similar to the formulation given in [4]. Finally, we propose two equivalent mathematical programming formulations in Sections 2.3 and 2.4.

2.1 Problem Description

In a casino, workforce scheduling is subject to manifold restrictions. We have to distinguish between hard restrictions that must be obeyed and soft restrictions which should be obeyed. Since there may be no feasible solution which satisfies all soft restrictions, violations of soft restrictions will be penalized. Moreover, we discern four different shift types: early shifts (E), mid-day shifts (M), late shifts (L), and long late shifts (LL). Each task that has to be assigned to an employee belongs to exactly one of these shift types.

Hard restrictions:

1. No employee may be assigned to a task he is not qualified for.
2. In between two tasks that are assigned to an employee, 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 at least 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 (long) late shift.
3. A vacation should be preceded by an early shift and succeeded by a (long) late shift.

Notice that due to the above soft restrictions, a "good" work sequence should usually resemble a sequence of shifts of the form $(LL, \ldots, L, \ldots, M, \ldots, E, \ldots, DO, DO, \ldots, LL, \ldots)$, where DO denotes a day off.

2.2 Network Representation

We consider a planning horizon of $h = 1, \ldots, H$ days. Let J_h be the set of all tasks (jobs) that take place on day h. Additionally, let each set J_h contain exactly two fictitious tasks:

- One dummy task, which is used to admit assignments of superfluous employees to tasks on day h. In the environment of casinos, the assignment of an employee to a dummy task represents for instance the assignment to a reserve pool, which contains standby employees. Alternatively, dummy tasks may represent demonstration services that serve to familiarize customers with some of the games offered by a casino.
- One task (day off), which is interpreted as a day off assignment.

We define $J := \bigcup_{h=1}^{H} J_h$ to be the superset of all tasks. With each task $u \in J$ we associate two events s_u and e_u representing the start (initial event) and end (terminal event) of task u, respectively. Changeovers from a task on day h to another task on day $h+1$, $h = 1, \ldots, H-1$, are modelled by

so-called transitions. A transition between two tasks $u \in J_h$ and $v \in J_{h+1}$ is associated with the terminal event of task u and the initial event of task v and can thus be uniquely identified by the pair (e_u, s_v). Transition (e_u, s_v) is considered requiresfeasible, if it respects hard restriction 2, i.e., if there are at least 11 hours of idle time in between the end of task u and the start of task v.

Let K denote the set of all employees that are available for at least one day of the planning horizon. For each employee $k \in K$ we consider a network $G^k = (N^k, A^k)$ (cf. Figure 1). The set of nodes N^k consists of the initial events s_u and the terminal events e_u for each task $u \in J$ that employee k is suitably skilled for (cf. hard restriction 1) and of a source α and a sink ω. It is assumed that all employees are skilled for dummy tasks and days off. In case an employee is not available for a specific day, for instance due to vacation plans, all associated tasks are omitted from N^k and instead a new fictitious task is introduced, which represents the employee's unavailability. The set of arcs A^k contains an arc (s_u, e_u) connecting the initial and terminal events of each task $u \in J : s_u, e_u \in N^k$. Moreover, there is an arc (e_u, s_v) for each feasible transition in between two tasks $u \in J_h : e_u \in N^k$ and $v \in J_{h+1} : s_v \in N^k$, $h = 1, \ldots, H-1$. Finally, A^k contains the arcs (α, s_u) for each $u \in J_1 : s_u \in N^k$ and (e_v, ω) for each $v \in J_H : e_v \in N^k$. For each arc $(i,j) \in A^k$ we introduce a cost c_{ij}^k that serves to penalize violations of the soft restrictions described in Section 2.1 and takes into consideration employee k's preference and qualification for the task at hand. For example, in order to observe soft restriction 2 (at least two days off in a row), it has to be ensured that transitions connecting two days off are at least as "cheap" (with respect to c_{ij}^k) as any other transition on the day under consideration. Similarly, transitions connecting a day off and a task belonging to a late shift have to be "cheaper" than transitions connecting days off and early or mid-day shifts. Observe that a path from node α to node ω in G^k corresponds to a feasible work sequence with respect to hard restrictions 1 and 2.

The superimposition of networks G^k for all employees $k \in K$ can be interpreted as a multi–commodity flow network $G = (N, A)$ with $N := \bigcup_{k \in K} N^k$ and $A := \bigcup_{k \in K} A^k$, where each employee k represents a unique commodity. For ease of illustration we stipulate that the set of arcs A is disaggregated into four mutually disjoint subsets $A_J \cup A_D \cup A_O \cup A_T = A$, where

- A_J denotes the set of all arcs corresponding to a non-fictitious task,
- A_D denotes the set of all arcs corresponding to a dummy task,
- A_O denotes the set of all arcs corresponding to a day off, and

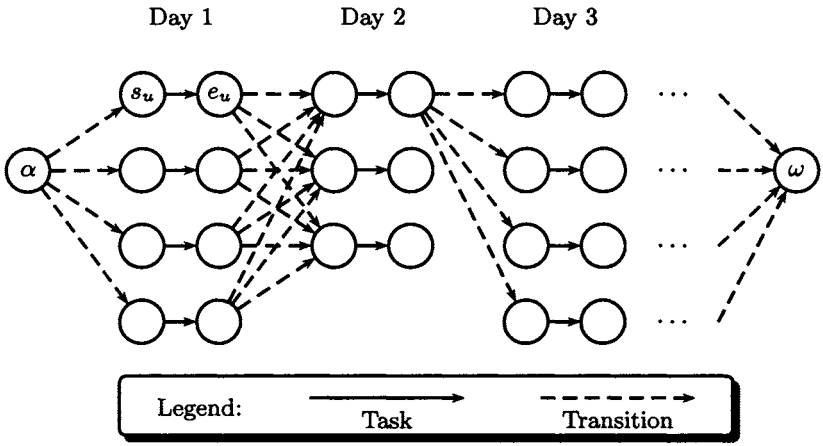


Figure 1: Sketch of network $G^k = (N^k, A^k)$

- A_T denotes the set of all arcs corresponding to a feasible transitions between two tasks.

Except for the representation of days off, the proposed multi–commodity flow network closely resembles the formulation given in [7].

2.3 Arc-Based Mathematical Formulation

Let ϕ_{ij}^k be a binary flow variable for each arc $(i,j) \in A^k$, $k \in K$, and d_{ij} be a nonnegative artificial variable for each $(i,j) \in A_J$, which may be interpreted as the assignment of a dummy employee to the task corresponding to arc (i,j). Moreover, $\mathcal{S}(i)$ denotes the set of all successors of node i in G^k and analogously $\mathcal{P}(j)$ denotes all predecessors of node j in G^k. Given these prerequisites, we propose the following mathematical program to determine a feasible work sequence for each employee.

$$\text{Min.} \quad \sum_{k \in K} \sum_{(i,j) \in A^k} c_{ij}^k \phi_{ij}^k + \sum_{(i,j) \in A_J} M d_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{(\alpha,j) \in A^k} \phi_{\alpha j}^k = 1 \qquad (k \in K) \tag{2}$$

$$\sum_{(i,\omega) \in A^k} \phi_{i\omega}^k = 1 \qquad (k \in K) \tag{3}$$

$$\sum_{j \in \mathcal{S}(i)} \phi_{ij}^k - \sum_{i \in \mathcal{P}(j)} \phi_{ij}^k = 0 \qquad (k \in K,\ i \in N^k \setminus \{\alpha, \omega\}) \tag{4}$$

$$\sum_{k \in K} \phi_{ij}^k + d_{ij} = 1 \qquad ((i,j) \in A_J) \tag{5}$$

$$d_{ij} \geq 0 \qquad ((i,j) \in A_J) \tag{6}$$

$$\phi_{ij}^k \in \{0,1\} \qquad (k \in K,\ (i,j) \in A^k) \tag{7}$$

Restrictions (2) – (4) are canonical flow balance constraints which – together with constraints (7) – ensure that each employee k is assigned exactly one path from α to ω in G^k. Constraints (5) make sure that each non-fictitious task is assigned to exactly one (dummy) employee. Objective function (1) minimizes the total cost incurred and – given a large number M which is e.g. greater than the length of a longest path from α to ω – ensures the utilization of a minimal number of dummy employees. Dummy employees are interpreted as temporary employees such as students, who work in a casino on an irregular basis. In a post processing phase, we therefore need to assign each task covered by a dummy employee to exactly one employee out of a set of available temporary employees.

Note that a feasible solution to problem (1) – (7) respects only hard restrictions 1 and 2. In order to account for hard restrictions 3 and 4 we need to introduce the concept of resource constraints for a path from α to ω in G^k. Let Γ be a set of resources γ. With each arc $(i,j) \in A$ we associate a resource consumption r_{ij}^γ for each resource $\gamma \in \Gamma$ (cf. Figure 2). In principle, we accumulate the resource consumptions along a path from α to ω and and we use $R_i^{\gamma k}$ to denote the total consumption of resource $\gamma \in \Gamma$ along a path from node α to some node i in G^k. In case a path from α to i is extended by arc $(i,j) \in A^k$ the total resource consumption at node j is given by $R_j^{\gamma k} := \max\{0, R_i^{\gamma k} + r_{ij}^\gamma\}$. Without loss of generality we assume that $R_\alpha^{\gamma k}$ equals 0 for all $\gamma \in \Gamma$, $k \in K$. For all $k \in K$, let $\bar{R}_i^\gamma$ be an upper bound on the total resource consumption $R_i^{\gamma k}$ of each resource γ and for each node $i \in N^k$. A path between two nodes in G^k is considered admissible exactly if $0 \leq R_i^{\gamma k} \leq \bar{R}_i^\gamma$, $\gamma \in \Gamma$, holds true for every node i that is part of the path under consideration. Notice that unlike r_{ij}^γ and $\bar{R}_i^\gamma$ the total resource consumption $R_i^{\gamma k}$ depends on employee $k \in K$.

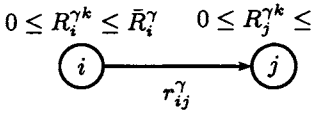


Figure 2: Resource consumption and resource constraints

To ensure that no employee works more than 10 days in a row (hard restriction 3) we introduce a resource γ_1 which serves to count the number

of consecutive work days. With each arc $(i,j) \in A_J \cup A_D$ corresponding to a (dummy) task we associate a resource consumption $r_{ij}^{\gamma_1} := 1$, and for each arc $(i,j) \in A_O$ that corresponds to a day off, we set $r_{ij}^{\gamma_1} := -H$. For all remaining arcs $(i,j) \in A_T$ we set $r_{ij}^{\gamma_1} := 0$. Moreover, we introduce an upper bound $\bar{R}_{s_u}^{\gamma_1} := 9$ for each initial event $s_u \in N$ belonging to a (dummy) task u, and we set $\bar{R}_i^{\gamma_1} := \infty$ for all remaining nodes $i \in N$.

Hard restriction 4 implies that for every 10 days on duty an employee is entitled to at least 4 days off. Consequently, we introduce a resource γ_2 which keeps track of the minimum number of days off an employee is entitled to and set $r_{ij}^{\gamma_2} := 4/10$ for each arc $(i,j) \in A_J \cup A_D$ corresponding to a non-fictitious task or a dummy task. With each arc $(i,j) \in A_O$ representing a day off, we associate a resource consumption $r_{ij}^{\gamma_2} := -1$. $r_{ij}^{\gamma_2} := 0$ for all remaining arcs $(i,j) \in A_T$. For each initial event $s_u \in N$ belonging to a (dummy) task u we set $\bar{R}_{s_u}^{\gamma_2} := 4$. With all remaining nodes $i \in N$ we associate an upper bound $\bar{R}_i^{\gamma_2} := \infty$.

Given the aforementioned resources and corresponding resource consumptions as well as upper bounds on the total resource consumptions, we obtain the following constraints depending on decision variables ϕ_{ij}^k and $R_i^{\gamma k}$.

$$\phi_{ij}^k \left(R_i^{\gamma k} + r_{ij}^{\gamma} - R_j^{\gamma k} \right) \leq 0 \quad (k \in K,\ (i,j) \in A^k,\ \gamma \in \Gamma) \tag{8}$$

$$0 \leq R_i^{\gamma k} \leq \bar{R}_i^{\gamma} \quad (k \in K,\ i \in N^k,\ \gamma \in \Gamma) \tag{9}$$

These constraints ensure that all paths from α to ω in G^K, $k \in K$, that are part of a solution to problem (1) – (7), are feasible with respect to resources γ_1 and γ_2. Hence, a solution to problem (1) – (9) respects all our hard restrictions.

2.4 Path-Based Mathematical Formulation

Recall from network theory that flows can be defined not only on arcs, but also an paths and cycles (cf. for instance [1, pp. 79]). Since G^k, $k \in K$, is acyclic, any feasible solution to problem (1) – (9) can be represented as the sum of flows on directed paths from α to ω. In what follows, we present a path-based formulation for the workforce scheduling problem under consideration, which is equivalent to problem (1) – (9). This formulation facilitates the derivation of the branch-and-price algorithm in Section 3.1.

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

$$\delta_{ij}^{p^k} := \begin{cases} 1 & \text{if } (i,j) \in p^k \\ 0 & \text{otherwise} \end{cases} \quad (p^k \in P^k,\ k \in K)\ .$$

An arc flow as introduced in Section 2.3 can be decomposed into path flows by

$$\phi_{ij}^k = \sum_{p^k \in P^k} \delta_{ij}^{p^k} x^{p^k} \ . \tag{10}$$

Moreover, the cost per unit flow for a given path $p^k \in P^k$, $k \in K$, is

$$c^k(p^k) := \sum_{(i,j)\in p^k} c_{ij}^k \ .$$

Using equation (10) we substitute the arc flow variables ϕ_{ij}^k in problem (1) – (9) with path flow variables x^{p^k} and thus obtain the equivalent optimization problem

$$\text{Min.} \quad \sum_{k\in K} \sum_{p^k \in P^k} c^k(p^k) x^{p^k} + \sum_{(i,j)\in A_J} M d_{ij} \tag{11}$$

$$\text{s.t.} \quad \sum_{p^k \in P^k} x^{p^k} = 1 \qquad (k \in K) \tag{12}$$

$$\sum_{k\in K} \sum_{p^k \in P^k} \left(\delta_{ij}^{p^k} x^{p^k}\right) + d_{ij} = 1 \qquad ((i,j) \in A_J) \tag{13}$$

$$\sum_{p^k \in P^k} \left(\delta_{ij}^{p^k} x^{p^k}\right)\left(R_i^{\gamma k} + r_{ij}^{\gamma} - R_j^{\gamma k}\right) \le 0 \qquad (k \in K, (i,j) \in A^k, \gamma \in \Gamma) \tag{14}$$

$$0 \le R_i^{\gamma k} \le \bar{R}_i^{\gamma} \qquad (k \in K, i \in N^k, \gamma \in \Gamma) \tag{15}$$

$$d_{ij} \ge 0 \qquad ((i,j) \in A_J) \tag{16}$$

$$x^{p^k} \in \{0,1\} \qquad (k \in K,\ p^k \in P^k) \tag{17}$$

Restrictions (12) together with (17) ensure that each employee k is assigned exactly one path from α to ω in G^k. Constraints (13) make sure that each non-fictitious task is covered by at most one employee k. Objective function (11) minimizes the total cost of all paths from α to ω as well as the utilization of dummy employees. Due to the structural properties of G^k, $k \in K$, (cf. Section 2.2) and due to resource constraints (14) and (15), a feasible solution to problem (11) – (17) is also feasible with respect to hard restrictions 1 through 4. Observe that because of nonlinear constraints (14) as well as the huge number of decision variables x^{p^k} problem (11) – (17) is difficult to solve. Moreover, notice that (11) – (17) contains the well-known set-partitioning problem, which is in itself known to be $\mathcal{NP}$-hard. However, using the branch-and-price solution procedure given in Section 3.1, the path-based formulation of the underlying problem can actually be solved more efficiently than the aforementioned arc-based formulation.

3 Solution Procedures

In the following Sections 3.1 and 3.2 we describe an exact solution procedure for problem (11) – (17) (cf. [7]) as well as a heuristic local search algorithm for the underlying workforce scheduling problem (see [6]). A heuristic solution provides an upper bound for the objective function value of problem (11) – (17) and can therefore be used to improve the performance of the exact algorithm given in the following Section.

3.1 An Exact Branch-and-Price Algorithm

Branch-and-Price (cf. e.g. [2]) is a powerful method for solving integer programs with a large number of decision variables. The basic idea is to use column generation in order to solve the linear relaxation – the so called master problem Π – of the underlying integer program. To this end, a restricted master problem Π' is solved to optimality, which contains only a small subset of all possible decision variables and corresponding columns of the coefficient matrix. Afterwards, a subproblem is formulated which serves to determine further columns and associated variables with negative reduced objective function values. These columns are appended to the restricted master problem which is then reoptimized. The process of solving Π' and determining and appending favorable columns to Π' is repeated until no further columns with negative reduced objective function values exist, which implies that the current optimal solution of Π' is also optimal with respect to Π. The described column generation procedure is embedded into a branch-and-bound algorithm in order to generate integral solutions.

3.1.1 Restricted Master Problem

For the underlying workforce scheduling problem (11) – (17), we obtain the master problem Π by replacing restrictions (17) with $x^{p^k} \geq 0$ for all $k \in K$, $p^k \in P^k$. The restricted master problem Π' is obtained by considering an appropriate subset $P'^k \in P^k$ of variables x^{p^k}, each of which represents a feasible path from α to ω in G^k regarding the resource constraints (14) and (15). A proper subset $P'^k \in P^k$ is given by any basic feasible solution to Π or can be generated by constructing a set of paths from α to ω, one path for each employee k, such that the paths are arc-disjoint with respect to the arcs in A_J. This can be easily done by e.g. constructing paths that traverse only arcs that correspond to dummy tasks and days off as well as appropriate transitions. Since we require each path in P'^k to be resource-feasible, we are able to drop restrictions (14) and (15) from Π'. Thus, Π'

can be solved using regular linear programming algorithms.

3.1.2 Subproblem

Once an optimal solution of the restricted master problem Π' has been obtained, we need to solve a subproblem in order to determine additional decision variables x^{p^k}, $p^k \in P^k$, with negative reduced objective function coefficients $\tilde{c}^k(p^k)$. Since an optimal solution to Π' provides shadow prices μ^k for each employee $k \in K$ (cf. restrictions (12)) and η_{ij} for each arc $(i,j) \in A_J$ (cf. restrictions (13)), it follows that

$$\tilde{c}^k(p^k) := \sum_{(i,j)\in p^k} \underbrace{(c_{ij}^k - \eta_{ij})}_{\text{modified cost } \bar{c}_{ij}^k} - \mu^k$$

for each feasible path $p^k \in P^k$, where $\eta_{ij} := 0$ for all $(i,j) \in A \setminus A_J$. From simplex theory we know that an optimal solution to Π must satisfy the optimality conditions

$$\begin{aligned} \tilde{c}^k(p^k) &\geq 0 \quad \text{for all feasible } p^k \in P^k, k \in K, \text{ and} \\ \tilde{c}^k(p^k) &= 0 \quad \text{for all } p^k \text{ with } x^{p^k} > 0\,. \end{aligned}$$

These conditions imply that μ^k is the length of a shortest path from α to ω in G^k with respect to the modified cost $\bar{c}_{ij}^k$. Hence, a path $p^k \in P^k$, $k \in K$, with minimum $\tilde{c}^k(p^k)$ can be determined by finding a shortest resource-feasible path from α to ω in G^k with respect to the modified cost $\bar{c}_{ij}^k$. If for all $k \in K$ the length $\sum_{(i,j)\in p^k} \bar{c}_{ij}^k$ of such a shortest path is greater than or equal to μ^k, i.e., $\tilde{c}^k(p^k) \geq 0$, then the optimal solution which has been previously determined for Π' is optimal for Π as well. Otherwise, i.e., $\sum_{(i,j)\in p^k} \bar{c}_{ij}^k < \mu^k$ holds, we append the column associated with path p^k as well as variable x^{p^k} to Π' and reoptimize the restricted master problem.

An algorithm which can be used to determine a shortest path under resource constraints is given in [3]. Its fundamental idea is to use multidimensional labels with label components for the total resource consumption $R_i^{\gamma k}$, $\gamma \in \Gamma$, at a given node i as well as the corresponding length of the path from node α to node i. Observe that different from regular shortest path algorithms, it is usually necessary to simultaneously retain several labels at each node in G^k in order to determine a shortest path under resource constraints. As described in [3], it is therefore beneficial to apply dominance criteria in order to reduce the number of labels to be considered at each node.

3.1.3 Branch-and-Bound

In order to determine an integral solution to the underlying workforce scheduling problem, it is possible to embed the discussed column generation approach into a branch-and-bound algorithm. An optimal solution to Π will then serve as a lower bound on the objective function value of problem (11) – (17). In order to branch on problem (11) – (17) or any of its subproblems, we propose to use variable dichotomy, i.e., at each node in the enumeration tree we select exactly one variable x^{p^k} which is fractional in an optimal solution of the corresponding linear relaxation. Two subproblems are then obtained as follows:

- We set $x^{p^k} := 1$, i.e., employee k must work the schedule associated with path p^k. Thus, we delete all variables (columns) $x^{\bar{p}^k}$ with $\bar{p}^k \in P^k$, $\bar{p}^k \neq p^k$. Moreover, it is possible to delete all work sequences for employees $k' \neq k$ which cover a task that has been assigned to employee k. This means we delete all variables $x^{p^{k'}}$ $(p^{k'} \in P^{k'}, k' \in K, k' \neq k)$ for which there is an arc $(i,j) \in p^k \cap A_J$ with $\delta_{ij}^{p^k} = \delta_{ij}^{p^{k'}}$. No further columns are generated for employee k in subsequent subproblems.

- We set $x^{p^k} := 0$, i.e., employee k must not work the schedule associated with path p^k. Hence we delete variable (column) x^{p^k} and we need to make sure that path p^k is not generated for employee k in subsequent subproblems, i.e., nodes of the enumeration tree. To this end, we store a list of forbidden paths for each employee k and each node of the enumeration tree. In case a forbidden path is obtained using the aforementioned shortest path algorithm, we compute the next-shortest path. This can be done by means of a k-shortest path algorithm, such as Yen's algorithm described in [8], which can be easily adapted towards the case with resource constraints. Its basic concept is to systematically and temporarily delete arcs from A which are part of a previously computed shortest path. After each such deletion, a new shortest path is determined that is a candidate for the next-shortest path.

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 enumeration tree are increasingly easy to solve.

3.2 A Heuristic Local Search Algorithm

In Section 3.2.1, we describe a construction heuristic which is usually able to determine good solutions to the underlying workforce scheduling problem. We will then show how this initial solution can be improved by means of a Hill Climbing algorithm as described in Section 3.2.2.

3.2.1 Construction Procedure

An initial solution is obtained by determining an assignment of employees to tasks for each day of the planning horizon. To this end we formulate an assignment problem for each day which can be visualized as a bipartite graph (cf. Figure 3). Source nodes correspond to available employees and sink nodes correspond to daily tasks as well as days off, which are allocated as follows. The number of days off an employee is entitled to depends on the number of days he is available for work, which is presumed to be known in advance. The total number of days off is then allocated to individual days according to a predefined strategy. Sources and sinks are fully interconnected by directed arcs, which represent an assignment of employees to tasks or days off. To ensure that the number of sources equals the number of sinks, we insert either dummy employees or dummy tasks.

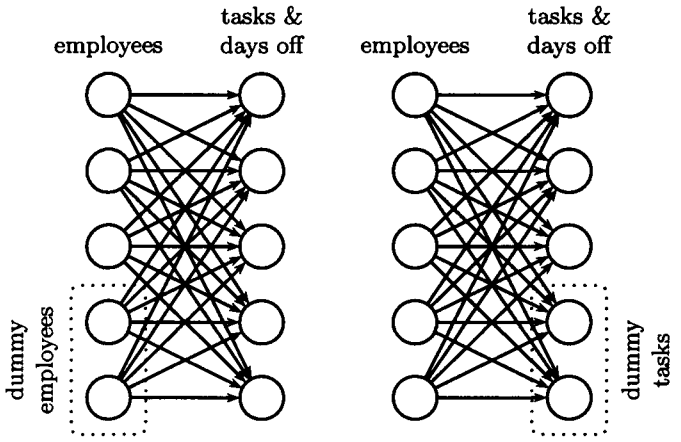


Figure 3: An assignment problem for a single day

The most important step during the initialization of a daily assignment problem is the determination of appropriate arc weights. Our goal is to determine arc weights which lead to work sequences in accordance with

the underlying restrictions. For this reason, we consider three factors when determining arc weights:

- *Qualification:* Arc weights are used to penalize assignments of employees to tasks for which they are not (sufficiently) skilled.
- *Previous assignments:* Considering the number of uninterrupted work days, the shift types of the last three assignments, and the shift type of the assignment under consideration, we penalize long periods of work as well as illegal or unfortunate work sequences w.r.t. soft restrictions 1 - 3.
- *Day off assignments:* We consider the number of days off an employee has yet to receive, the remaining number of days until the end of the planning horizon, and the shift types of previous assignments. Then we penalize supernumerous and insufficient assignments of days off as well as assignments that are likely to violate the underlying soft restrictions.

The assignment problem for each day can be polynomially solved by means of the well-known Glover-Klingman algorithm (cf. e.g. [5, pp. 294]). Note that the optimal solution of an assignment problem depends on solutions that were obtained for assignment problems on previous days.

3.2.2 Improvement Procedure

The initial solution obtained in Section 3.2.1 is improved by two Hill Climbing procedures that pursue two hierarchically ordered objectives: (a) improving the quality of work sequences and (b) evenly distributing attractive tasks among employees. Neighboring solutions are obtained by swapping the assignments between two employees for one and the same day of the planning horizon. Swapping is permitted only if the resulting neighboring solution complies with hard restrictions 1 through 4.

The first Hill Climbing procedure serves to improve the quality of individual work sequences regarding their violation of soft restrictions 1 - 3. Starting with $h = H$, we swap two assignments on day $h - 1$, if this improves the objective function value of the assignment problem for day h. This step is repeated for $h = H - 1, \ldots, 2$.

The second Hill Climbing procedure pursues the objective of evenly distributing the assignments of attractive tasks, i.e., tasks for which employees express high preferences, among all employees throughout the entire year. To this end, we determine the ratio of unattractive versus attractive tasks an employee has been assigned to throughout the year. Our objective is

to minimize the absolute deviation of this ratio for a given employee from the mean ratio over all employees. For each day $h = 1, \ldots, H$, we therefore swap two assignments on day h if this decreases the afore mentioned deviation.

The proposed neighborhood operator ensures that, given an initial allocation of days off to each day of the planning horizon, a globally optimal solution can be reached from any starting solution.

4 Conclusions

In this paper, we have presented a sophisticated network formulation for workforce scheduling problems which is rather lucid as well as easy to explain because of the close relationship between paths in the network and work sequences for individual employees. It should be noted that due to the concept of resource restrictions on paths the proposed multi-commodity network flow formulation is very flexible and allows for an easy incorporation of many additional practical restrictions. This permits its application to a lot of practical workforce scheduling problems outside of the casino industry.

Since the proposed branch-and-price procedure is only able to solve small problem instances to optimality, we have moreover described a constructive heuristic based on the iterative solution of general assignment problems and a simple hill climbing procedure. For the future, however, it seems worthwhile to develop heuristic procedures that are based on the aforementioned multi-commodity network flow representation. This would provide a basis for a clean performance comparison as well as the possibility to implement hybrid approaches.

References

[1] Ahuja RK, Magnanti TL, Orlin JB (1993) Network Flows. Prentice Hall, Englewood Cliffs

[2] Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch–and–price: Column generation for solving huge integer programs. Operations Research 46:316–329

[3] Desrosiers J, Dumas Y, Solomon MM, Soumis F (1995) Time-constrained routing and scheduling. In: Ball MO, Magnanti TL, Monma CL, Nemhauser GL (eds.) Handbooks in OR & MS, Vol. 8, Elsevier, Amsterdam, pp. 35–139

[4] Millar HH, Kiragu M (1998) Cyclic and non-cyclic scheduling of 12h shift nurses by network programming. European Journal of Operational Research 104:582–592

[5] Neumann K, Morlock M (2002) Operations Research. Hanser, München

[6] Stark C, Zimmermann J (2004) Scheduling regular and temporary employees with qualifications in a casino. In: Ahr D, Fahrion R, Oswald M, Reinelt G (eds.) Operations Research Proceedings 2003, Springer, Berlin, pp. 173–180

[7] Stark C, Zimmermann J (2005) An exact branch-and-price algorithm for workforce scheduling. In: Fleuren H, den Hertog D, Kort P (eds.) Operations Research Proceedings 2004, Springer, Berlin, pp. 207–212

[8] Yen JY (1971) Finding the k shortest loopless paths in a Network. Management Science 17:712–716

Part III
Operations Research in Logistics

Pile Problems and Their Mathematical Treatment

Lorenz Hempel and Roland Schmiedel

Institut für Mathematik / Physik
Bauhaus-Universität Weimar

Lutz Kämmerer

TXTe - solutions GmbH, Halle (Saale)

This paper is dedicated to **Prof. Dr. Klaus Neumann** on the occasion of his entrance into the retirement.
His working at the university of Karlsruhe is connected with a lot of activities not only inside the university but also outside all over the world. The mathematicians at the Bauhaus University Weimar, one of the universities in Thuringia, look back to a long time of productive cooperation with him gratefully. Especially we thank for his active support in the organization and carrying out of our three-annual conference IKM (International Conference about Application of Computer science and Mathematics in Architecture and Civil Engineering), where he was one of the most engaged and reliable members of the program committee over many years.

1 Introduction

There are lots of situations in which some items stacked up to a pile are to transfer in another pile with another order of the items. The question in this situation is how to do it in the best way using only the available resources and devices.
Such problems we call pile problems.
We meet them for example in connection with container transport, where the containers in a harbor are to load into or to unload from the containerships, or in connection with the transport and the assembly of large concrete elements, or in connection with the shunt of trains in a marshalling yard, but also in connection with some patience plays (freecell, Bakers patience) or the well known "Tower of Hanoi".
In the following we will investigate some types of pile problems and try to answer such questions like

- Is the pile problem solvable by the available resources?
- Which is the solution with the minimal number of steps?

- How to find this solution?

To describe pile problems more precisely we use the terms of graph theory. For a given directed graph $G = (V, A)$ and its vertices $v \in V$ we use the following well-known notations: $N_G^+(v)$ is the set of all successors of v, $N_G^-(v)$ is the set of all predecessors of v. $d_G^+(v)$ and $d_G^-(v)$ are the numbers of outgoing and incoming arcs of v, respectively. For two given graphs $G_1 = (V_1, A_1)$ and $G_2 = (V_2, A_2)$ the *union of* G_1 *and* G_2 is defined as: $G_1 \cup G_2 := (V_1 \cup V_2, A_1 \cup A_2)$. A *cycle* in a directed graph $G = (V, A)$ is a closed path in G. We call a directed graph $G = (V, A)$ a *pile*, if and only if G is finite and does not contain a loop or a cycle. The vertices of G represent the items or elements of the pile, the arcs of G represent the order in which the elements are piled up. The moving of an element of a pile to another place is called a *step*. Another place can be a place in another pile or in an auxiliary place (cellar place) if it exists.
Figures 1 and 2 show an example of a pile problem we will consider. The

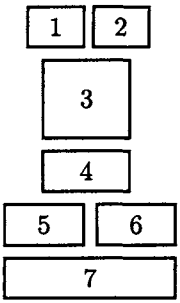


Figure 1: Starting situation

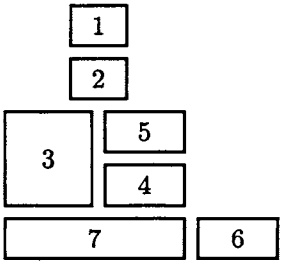


Figure 2: Required final situation

graphs G_S and G_F, representing the starting and the final pile, respectively, are shown in Figs. 3 and 4. Note that the arcs in G_S are directed in the opposite way to those in G_F.

Now the question is, how many elements must be put aside at least from the starting pile in order to pile up the required final structure. Erecting this final pile is done step by step by moving elements from top of the starting pile on top of the final pile we have yet. The above example can be solved by putting five elements aside: 1, 2, 3 ,4 and 5.

Definition 1 : *We call* $Q = (G_S, G_F, k)$ *a pile problem if and only if* $G_S = (V, A_S)$ *and* $G_F = (V, A_F)$ *are piles and* k *is a natural number.* G_S *is called starting pile,* G_F *is called final pile;* k *is the number of elements that are allowed to put aside. The vertices* v *from* V *are called elements of*

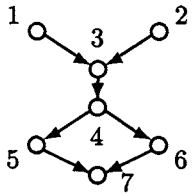


Figure 3: Starting pile G_S

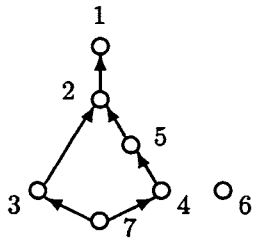


Figure 4: Final pile G_F

the piles.
$G'_S = (V', A')$ is a subpile of the starting pile G_S if G' is an induced subgraph of G_S and with vertex $v \epsilon V'$ all successors of v in G_S belong also to V'.
$G'_F = (V', A')$ is a subpile of the final pile G_F if G' is an induced subgraph of G_F and with the vertex $v \epsilon V'$ all predecessors of v in G_F belong also to V'.
A vertex v is top of a starting pile or its subpile G if v has no predecessors in G.
A vertex v is top of a final pile or its subpile G if v has no successors in G.

Definition 2 : *$S = (G_1, G_2, H, M)$ is called configuration of a pile problem $Q = (G_S, G_F, k)$, if and only if*

1. *$G_1 = (V_1, A_1)$ is a subpile of G_S that represents that part of the starting pile that still remains on the original place, $G_2 = (V_2, A_2)$ is a subpile of G_F that represents the part of the required final pile that is yet erected on the final place,*

2. *M is the set of elements that have been put aside so far, H is the set of elements that are currently resting on auxiliary places, $H \subseteq M$.*

For two given configurations $S = (G_1, G_2, H, M)$ and $S' = (G'_1, G'_2, H', M')$ of Q S' is called move from S if and only if either

1. *there is a top v of G_1 with $V'_1 = V_1 \backslash \{v\}$ and $V'_2 = V_2 \cup \{v\}$ (a transfer from the top of G_1 to the top of G_2), or*

2. *there is a top v of G_1 with $V'_1 = V_1 \backslash \{v\}$ and $M' = M \cup \{v\}$ and $H' = H \cup \{v\}$ (a transfer from the top of G_1 to an auxiliary place), or*

3. *there is a* $v \in H$ *with* $V_2' = V_2 \cup \{v\}$ *and* $H' = H \backslash \{v\}$ *(a transfer from an auxiliary place to* G_2*)*

Definition 3 : *Let* $Q = (G_S, G_F, k)$ *be a pile problem. A sequence* $\mathcal{S} = [S_1, \ldots, S_l]$ *of moves* $S_i = (G_1^i, G_2^i, H^i, M^i)$ *is called valid sequence of moves of* Q *if and only if*

1. $G_1^1 = G_S$, $G_2^1 = (\emptyset, \emptyset)$, $M^1 = \emptyset$ *and*
2. *for* $i \in \{1, \ldots, l-1\}$ S_{i+1} *is move from* S_i.

Then we call the configuration S_i *a valid configuration of* Q.
We call Q *solvable if and only if there is a valid sequence of moves* $\mathcal{S} = [S_1, \ldots, S_l]$ *such that for* $S_l = (G_1^l, G_2^l, H^l, M^l)$

$$G_2^l = G_F$$

holds. We call $\mathcal{S}$ *then a* l*-solving-sequence of* Q.

2 The Special Case of Chain Pile Problems

We now want to turn to a more simple pile structure: We call a pile $G = (V, A)$ with $V = \{v_1, \ldots, v_n\}$ a *chain pile* if and only if $A = \{(v_i, v_{i+1}) : i \in \{1, \ldots, n-1\}\}$ holds, i.e. G is simply a directed path. The pile is a stack. Hence we investigate piles where the elements are ordered like permutations. Again we want to transform a starting pile into a final pile consisting of the same elements. This leads to the new term of *chain pile problems*: $P = (G_S, G_F, G_R, k)$ is called *chain pile problem* if and only if $G_S = (V, A_S)$ and $G_F = (V, A_F)$ are chain piles, $G_R = (V, A_R)$ is a directed graph – the *graph of rules or rule system* – and k is a natural number – the number of auxiliary places that are available.

The rules from G_R describe, which elements can rest on other ones: For a given chain pile problem P a vertex u may rest on top of a vertex v if and only if there is the arc (u, v) in the graph of rules G_R, $(u, v) \in A_R$. These rules hold on all places, on the auxiliary places also.
Without loss of generality may be G_S allways the permutation $(1, 2, 3, ..., n)$ and G_F the permutation $\pi = (i_1, i_2, i_3, ..., i_n)$.
The well known "Tower of Hanoi" is an example for such a chain pile problem. The starting and the final pile are stacks both given by the permutation $(1, 2, 3, ..., n)$ of the n elements. The graph of rules consists of the vertices $1, 2, 3, ..., n$ and of the arcs (i, j) where $i < j$. The number of auxiliary places is $k = 1$.
"Tower of Hanoi" is a solvable pile problem and needs with $2^n - 1$ an

exponential number of steps.
For chain pile problems some interesting questions are

1. Is the pile problem solvable by the rule graph for given permutation π and given number k of auxiliary places?
2. Which is the minimal rule system for given k and given π, i.e. the graph G_R with a minimal number of arcs for wich the problem is solvable?
3. Which is the minimal number of auxiliary places for which the pile problem is solvable for given π and given G_R?
4. For which permutations π is a pile problem solvable for given k and given G_R?
5. Which is the minimal number of steps to solve a pile problem for given π, given G_R and given k, if the problem is solvable?
6. Is there a polynomial bound for this number of steps for all solvable permutations π?

For some of these questions the answer is known, for example

Theorem 1 : *The minimal rule system of the "Tower of Hanoi" problem is given by the graph* $G_{R_n} = (V_{R_n}, A_{R_n})$ *with*

$$V_{R_n} = \{1, 2, 3, ..., n\} \text{ and}$$
$$A_{R_n} = \{(i, i+3), (i, i+5), ..., (i, n - P(i,n)) : i \in \{1, 2, ..., n-3\}\} \cup \{(i, i+1) : i \in \{1, 2, ..., n-1\}\}$$

where

$$P(i,j) := \begin{cases} 1 & \text{for } i+j \text{ even} \\ 0 & \text{for } i+j \text{ odd} \end{cases} .$$

G_{R_n} *is a bipartite graph.*

The proof of the theorem you can find in [3].

For a general chain pile problem the rule graph must contain at least all arcs of G_S and all arcs of G_F , otherwise the problem is not solvable.
For such problems with $G_R = G_S \cup G_F$ obviously holds

Theorem 2 : *Each chain pile problem* $P = (G_S, G_F, G_S \cup G_F, n/2 + 1)$ *is solvable.*

The investigation of question 4 is more difficult when k is smaller then $n/2$. In [3] are given some results to this question under further assumptions. These assumptions are

1. All elements of the starting pile must be moved to auxiliary places, no elements may be put to the final place while the starting place is non empty (it is not allowed to put an element directly from the starting pile to the final pile).

2. It is not allowed to put elements back to the starting pile.

3. It is not allowed to take elements away from the final pile.

4. The number of auxiliary places is $k = 2$.

Then holds

Theorem 3 *: In a chain pile problem $P = (G_S, G_F, G_S \cup G_F, 2)$ with the additional assumptions 1. - 4. and for $n > 4$ under the $n!$ permutations of the n, $(n > 4)$ elements are exactly $8n - 16$ for which the pile problem is solvable. Each of these valid final piles can be given in an explicit way (see [3]). Moreover each of the $8n - 16$ solvable chain pile problems can be solved using only at most $5n - 5$ steps.*

The proof of this theorem is possible by considering a lot of different cases and will be hold back here. It is to find in [3].

3 Solvability and Complexity of Pile Problems

Now again we consider a pile problem in accordance with definition 1.

We want to examine the structure of the problem and give some properties.

Lemma 1 *: Let $Q = (G_S, G_F, k)$ be a solvable pile problem. Then for any cycle C in $G_S \cup G_F$ there is a vertex x_C with (x_C, y_C) in C and $(x_C, y_C) \in A(G_S)$. x_C must be put aside in order to pile up G_F.*

Proof: We consider a cycle $C = [x_1, x_2, \ldots, x_w, x_1]$ in $G_S \cup G_F$. Let x_i be the vertex in C, that is put first of all vertices $x_1, \ldots, x_w$ to the final pile. The arc (x_{i-1}, x_i) can't be in G_F, because x_i has to be put to the final pile before x_{i-1}. That's why $(x_{i-1}, x_i) \in A(G_S)$ holds. That means: x_{i-1} must be taken away from the starting pile before x_i, but may be put to the final pile after x_i. This can be realized only if x_{i-1} is put to an auxiliary place. The arc (x_{i-1}, x_i) is the arc (x_C, y_C) mentioned in the lemma. $\diamond$
As a consequence we have

Theorem 4 ([3]) *: A pile problem $Q = (G_S, G_F, 0)$ is solvable if and only if $G_S \cup G_F$ does not contain any cycles.*

Computing the minimal number of elements that must be put aside to erect the final pile turns out to be NP-hard:

Theorem 5 : *The set*
$SP := \{ Q = (G_S, G_F, k) : Q$ *is a solvable pile problem* $\}$ *is NP-complete.*

To proof Theorem 5 SP is reduced to the problem FEEDBACK-VERTEX-SET, which is NP-complete by [1]. A full-length proof is given in [3].

Remark 1 : *By theorem 5 follows that the determination of the minimal* k *for which* $Q = (G_S, G_F, k)$ *is solvable also is NP-hard.*

By Theorem 4 we know that cycles are the reason for putting elements aside when piling up the required final structure. Taking a closer look to the structure of these cycles one realizes that they consist by turns of paths P_i^S from G_S and paths P_i^F from G_F. To reach the final situation by Lemma 1 every cycle in $G_S \cup G_F$ must be destroyed. Break such a cycle C can only be done by moving elements from top of the starting pile or one of its subpiles to an auxiliary place. That means we have to choose a path P_i^S in C the first vertex v of which $(d^-_{P_i^S}(v) = 0)$ will be put aside.

For simple-structured cycles there is no choice:

Lemma 2 *[3] : Let* $Q = (G_S, G_F, k)$ *be a solvable pile problem and let* $C = [a_1, \ldots, a_x, b_1, \ldots, b_y, a_1]$ *be a cycle in* $G_S \cup G_F$ *such that* $P_1 = [a_1, \ldots, a_x, b_1]$ *is contained in* G_S *and* $P_2 = [b_1, \ldots, b_y, a_1]$ *is contained in* G_F. *Then the vertex* a_1 *has to be put aside in any case when piling up the required final structure* G_F.

Proof : By Lemma 1 we already know that there is a $a_i \in \{a_1, \ldots, a_x\}$ which must be put aside in order to reach the final situation G_F. Due to $P_1' = [a_1, \ldots, a_x]$ is contained in G_S this must be a_1. $\diamond$

Definition 4 : *A directed subgraph* $G' = (V', A')$ *of* $G = (V, A)$ *is called a strongly connected component if for every pair* (x, y) *of vertices from* V' *there is a directed path from* x *to* y *in* G' *and* G' *is maximal, i.e. all other subgraphs* $G'' \subseteq G$ *with* $G' \subset G''$ *are not strongly connected.*

Lemma 3 : *Let* G_S *and* G_F *be a starting and a final pile respectively. Furthermore* $G_S \cup G_F$ *is strongly connected and* $n := |V(G_S)|$ *and* $m := |A(G_S \cup G_F)|$. *Then there is a* k *with* $1 \leq k \leq m - n + 1$, *such that* $Q := (G_S, G_F, k)$ *is a solvable pile problem.*

Proof : $G_S \cup G_F$ is strongly connected, therefore it contains a cycle. By Lemma 2 at least one vertex must be put aside, hence $1 \leq k$ holds.

There is always an element v for which at least one arc in $G_S \cup G_F$ is deleted when v putting aside. A spanning tree of $G_S \cup G_F$ contains $n-1$ arcs, therefore at most $m-(n-1)$ arcs must be deleted, in order to destroy all cycles in $G_S \cup G_F$. That means at most $m-(n-1)$ elements have to put aside, hence $k \leq m-n+1$ holds. $\diamond$

The bounds given in Lemma 3 can not be tightened as the Figs. 5 and 6 illustrate.

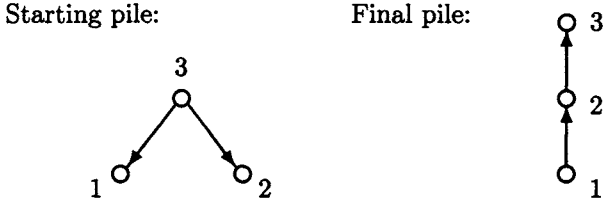


Figure 5: Starting and final pile of a pile problem that shows that the lower bound is tight

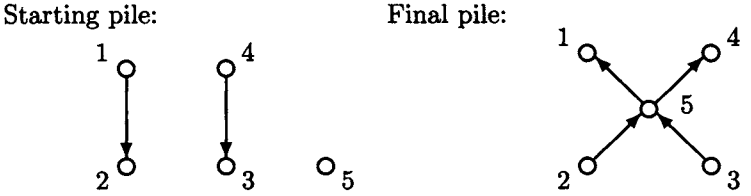


Figure 6: Starting and final pile of a pile problem demonstrating that the upper bound is also tight

Theorem 6 : *Given a pile problem $Q = (G_S, G_F, k)$ with k is the smallest number for which Q remains solvable. dc denotes the number of cycles in $G_S \cup G_F$ that are disjoint in vertices and l is the number of non-trivial strongly connected components $Q^1, \ldots, Q^l$ of $G_S \cup G_F$. Furthermore every*

component Q_i contains n_i vertices and m_i arcs. Then it holds

$$l \leq dc \leq k \leq \sum_{i=1}^{l} \min\{m_i - n_i + 1,\ n_i - 1\}.$$

Proof : There may be more than one disjoint cycles in a non-trivial strongly connected component – at least there is one – but no cycle contains vertices of several strongly connected components. Therefore $dc \geq l$ holds.

By Lemma 1 every cycle has to be cut open by putting an element aside. By putting k elements aside, only k disjoint cycles can be destroyed. It follows $k \geq dc$.

For any strongly connected component with n_i vertices and m_i arcs obviously there have to be put aside at most $n_i - 1$ vertices. According to Lemma 3 at most $m_i - n_i + 1$ elements have to be moved to auxiliary places. Therefore we can choose the minimum of either of these values for every strongly connected component of Q:
$\sum_{i=1}^{l} \min\{m_i - n_i + 1,\ n_i - 1\} \geq k$. $\diamond$

With the argumentation from the proof of Theorem 6 we can conclude that it is sufficient to consider the strongly connected components of a pile problem Q in order to decide whether Q is solvable or not. Hence for an algorithm it is useful to decompose Q into subproblems $Q^1, \ldots, Q^l$ by determining the strongly connected components of Q.

Definition 5 : *Let $Q = (G_S, G_F, k)$ be a pile problem. We call the set $\mathcal{Q} := \{Q^i = (G_S^i, G_F^i, k_i) : i = 1, \ldots, l\}$ of pile problems scc-decomposition of Q, if and only if all $G_S^i \cup G_F^i$ are strongly connected components of $G_S \cup G_F$, $V(G_S) = \cup_{i=1}^{l} V(G_S^i)$ and $V(G_F) = \cup_{i=1}^{l} V(G_F^i)$ holds and all G_S^i and G_F^i are (induced) subgraphs of G_S and G_F, respectively.*

Lemma 4 : *Let $Q = (G_S, G_F, k)$ be a pile problem and $\mathcal{Q} := \{Q^i = (G_S^i, G_F^i, k_i) : i = 1, \ldots, l\}$ such a scc-decomposition of Q that all $Q^i = (G_S^i, G_F^i, k_i)$ are solvable but $Q'^i = (G_S^i, G_F^i, k_i - 1)$ are not. That means k_i is minimal for Q^i. Then it follows*

$$Q \text{ is solvable if and only if } k \geq \sum_{i=1}^{l} k_i \text{ holds.}$$

Proof : ("⇒") $G_S \cup G_F$ consists of the strongly connected components $G_S^1 \cup G_F^1, \ldots, G_S^l \cup G_F^l$. To solve Q exactly c_i vertices have to put aside from the component $G_S^i \cup G_F^i$ $(i = 1, \ldots, l)$. Altogether k elements are put aside, i.e. $\sum_{i=1}^{l} c_i = k$.

Assume that Q is solvable and $k < \sum_{i=1}^{l} k_i$ holds. Thus $\sum_{i=1}^{l} c_i < \sum_{i=1}^{l} k_i$ holds. Therefore there is at least one component $G_S^j \cup G_F^j$, where c_j vertices $v_1, \ldots, v_{c_j}$ are put aside and $Q^j = (G_S^j, G_F^j, c_j)$ is solvable with $c_j < k_j$, because the whole problem Q is solvable. This contradicts our assumption of k_j being minimal.

("$\Leftarrow$") It holds $k \geq \sum_{i=1}^{l} k_i$. Then Q is solvable and there is no cycle leading through more than one strongly connected component. Hence only $\sum_{i=1}^{l} k_i \leq k$ elements must be put aside in order to solve Q. $\diamond$

There are now two ways of treating the problem algorithmically: First a pile problem Q can be solved by complete enumeration, which will be very time-consuming, due to its NP-completeness. Secondly it can be examined by an heuristic procedure that will work much faster but will deliver only an approximate solution. We will now have a closer look on these methods.

4 Solving Pile Problems

4.1 Branching & Bounding

To compute the minimal number of elements that have to put aside we use a simple technique: Step by step we simulate the moving of the elements from the starting pile to the final pile place. If there is no element on top of the starting pile that can be moved directly to the final pile we need to put an element aside. By Lemma 4 it is sufficient to consider all pile problems Q^i of the scc-decomposition of Q. The smallest number of elements that have to put aside to solve Q is the sum of all minimal numbers concerning the problems Q^i. Therefore we split $Q = (G_S, G_F, k)$ in subproblems $Q^i = (G_S^i, G_F^i, k_i)$ by decomposing Q in strongly connected components of $G_S \cup G_F$.

After checking that the component Q^i that currently is to investigate is non-trivial we check if there is a cycle in Q^i containing a vertex v that rests on top of the starting pile and which satisfies Lemma 2. In this case we put v aside on an auxiliary place. After this we may have access to new vertices on top of the starting pile. Now we can search for vertices on top of starting pile which either can be moved directly to the final pile or which satisfy Lemma 2. We repeat these steps as long as there are such vertices.

But once we reach a point where we don't find such vertices – we have to move an element from the starting pile to an auxiliary place. The problem is now that we do n't know the right vertex to choose, that's why we have to make the whole computation for every vertex on top of the starting pile: We move one vertex to an auxiliary place and solve the reduced problem recursively. This backtracking-procedure can be shortened to a

simple branch-and-bound algorithm. As a result we obtain the smallest number of elements that must be moved to auxiliary place to pile up the required final situation of Q.

4.2 Heuristics

The heuristic procedure we consider works similar to the branch-and-bound algorithm. At the point where there is neither a vertex to move directly to the final pile nor a vertex satisfying Lemma 2, we have to choose only one element to move from the starting pile to an auxiliary place. This decision is not corrigible, that's why our decision should be depending on a well defined criterion. For that reason we assign a value to all possible elements (the elements on top of starting pile) by a function f and choose a vertex v for which the value $f(v)$ becomes a maximum.

Clearly, the quality of the solution as well as the efficiency of the algorithm strongly depends on such a function f. Several functions were investigated:

Definition 6 *: Let $Q = (G_S, G_F, k)$ be a pile problem and $v_1, \ldots, v_l$ the elements on top of G_S or one of its subpiles. E_i denotes the set of successors of v_i in G_S which have no predecessors except v_i. We define the following functions for $i = 1, \ldots, l$:*

- $f_1(v_i) := d^+_{G_S}(v_i) + d^-_{G_F}(v_i)$, *the number of elements that are directly below v_i in the starting pile plus the number of elements that rest on top of v_i in the final pile.*
- $f_2(v_i) := |\{v \in E_i : v$ *can be put directly to the final pile.* $\}|$,
- $f_3(v_i) := |\{v \in E_i : v$ *satisfies Lemma 2.*$\}|$,
- $f_4(v_i) := f_2(v_i) + f_3(v_i)$.

Hence $f_4(v_i)$ is the number of vertices that either satisfy Lemma 2 or can directly be moved to the final pile right after v_i is taken away from the starting pile. The best results were obtained by using the function f_4. The time complexity for the heuristic algorithm with the function f_4 is $O(n^2 + nm)$ [3].

4.3 Behavior of the Algorithms

Before starting some experiments a number of example piles are to generate. The vertices of a directed acyclic graph can be ordered into certain levels,

two vertices are in the same level if the longest paths leading from top of the pile to the vertices have equal lengths.

The *average width* of such a graph G is the arithmetic mean of the number of vertices in the levels of G. The *height* of G is the number of the levels – the length of the longest path in G.

When doing our experiments we varied the average width and the height of the piles to generate, as well as the density of arcs in the piles.

When making the first experiments it happened that a lot of pile problems were solvable in reasonable time. This means: In the branch-and-bound algorithm there is no branching necessity – in the heuristic algorithm none of the vertices must be chose by f_4. Step by step we can find vertices until the whole problem is done. Of course, the results by the branch-and-bound and the heuristic algorithm are the same. In Table 1 we show the percentage of all examples that can be solved without branching/using f_4. The examples we considered here had at most 100 vertices, the density of arcs is 25 per cent. It means 25 per cent of all possible arcs are chosen randomly. We considered the dependency of the results on the structure of the graphs – the relation between height and width. The number of investigated examples is given in Table 3.

Table 1: Percentage of pile problems solvable in reasonable time with respect to the given parameters.

Density of arcs: 25 per cent, $\|V\| \leq 100$ r_S=height/width in the starting pile r_F=height/width in the final pile			
	$r_F = 10$	$r_F = 1$	$r_F = 0.1$
$r_S = 10$	97%	94%	91%
$r_S = 1$	96%	90%	83%
$r_S = 0.1$	90%	80%	63%

All statements about the heuristic algorithm in the following are with regard to those examples that cannot be solved in reasonable time, i.e. branching is always necessary.

Quality of the Approximate Solution We started a series of experiments to measure the quality of the heuristic algorithm – for that reason we considered a density of arcs in the piles of 5, 10, 15, 20 and 25 per cent. In Table 2 we give the percentage of optimal solutions (O) – the number

of elements to put to auxiliary place is minimal – and the percentage of "good" (G) solutions – the number exceeds the minimum at most by one.

Table 2: Percentage of optimal and good solutions provided by the heuristic algorithm for different density of arcs

Density of arcs: 25 per cent, $|V| \leq 50$
r_S=height/width in the starting pile
r_F=height/width in the final pile

	$r_F = 10$	$r_F = 1$	$r_F = 0.1$
r_S= 10	O: 93% G: 99%	O: 94% G: 99%	O: 88% G: 98%
r_S = 1	O: 91% G: 99%	O: 88% G: 99%	O: 82% G: 99%
r_S = 0.1	O: 84% G: 98%	O: 78% G: 97%	O: 74% G: 97%

Density of arcs: 15 per cent, $|V| \leq 50$
r_S=height/width in the starting pile
r_F=height/width in the final pile

	$r_F = 10$	$r_F = 1$	$r_F = 0.1$
r_S = 10	O: 86% G: 99%	O: 81% G: 98%	O: 82% G: 98%
r_S = 1	O: 81% G: 97%	O: 77% G: 96%	O: 77% G: 98%
r_S = 0.1	O: 81% G: 96%	O: 76% G: 96%	O: 82% G: 95%

Density of arcs: 10 per cent, $|V| \leq 50$
r_S=height/width in the starting pile
r_F=height/width in the final pile

	$r_F = 10$	$r_F = 1$	$r_F = 0.1$
r_S = 10	O: 71% G: 94%	O: 71% G: 95%	O: 79% G: 96%
r_S = 1	O: 70% G: 91%	O: 76% G: 94%	O: 79% G: 96%
r_S = 0.1	O: 77% G: 96%	O: 79% G: 94%	O: 81% G: 97%

Density of arcs: 5 per cent, $|V| \leq 50$
r_S=height/width in the starting pile
r_F=height/width in the final pile

	$r_F = 10$	$r_F = 1$	$r_F = 0.1$
r_S = 10	O: 75% G: 94%	O: 75% G: 94%	O: 84% G: 98%
r_S = 1	O: 79% G: 99%	O: 86% G: 97%	O: 91% G: 99%
r_S = 0.1	O: 87% G: 98%	O: 88% G: 99%	O: 95% G: 99%

Computing Times: Branch-and-Bound vs. Heuristic The computing times for the branch-and-bound and the heuristic algorithm are odd: The considered examples can be divided into two sets. The first and larger one contains piles for which the computing time is almost the same for both algorithms. The computing time $T_{\text{Heuristic}}(Q)$ for the heuristic algorithm is less than one second for these examples. (All given computing times are with regard to a Intel Pentium 133 with 32 MByte main memory.) The computing time $T_{\text{Branch-and-Bound}}(Q)$ of the exact method is also less than one second.

In the second set we find those piles for which the heuristic algorithm is much faster than the exact method, we find computing times from 30 seconds up to several hours, but the heuristic algorithm takes only a few seconds for the same examples. For bigger sized problems ($n \geq 200$) it's

difficult to decide whether the above statement is true or not, the heuristic algorithm also works very fast, but it's almost impossible to prove the quality of the results because it's impossible to calculate the exact values.

An overview about the necessary computing time of the branch-and-bound-algorithm is given with Table 3. We considered the same examples as in Table 2.

Table 3: Percentage of examples solvable with the branch-and-bound algorithm in different computing times depending on the density of arcs

Computing time	Density of arcs: 5%	10%	15%	20%	25%
<1s	87	74	91	82	59
1 – 5s	9	17	6.7	15	36
6 – 120s	3	5	1.7	2.1	4
121 – 7200s	0.8	3	0.4	0.7	0.6
> 7200s	0.2	1	0.2	0.2	0.4
Number of the considered examples	1240	2785	3043	2242	1623

Finally, the combination of both algorithms seems to be a good procedure: First is to check if the branch-and-bound algorithm returns a result within a defined time interval and only if it doesn't, the heuristic algorithm should be used to solve the problem.

5 Non-Unique Pile Problems

An obviously generalization of the pile problems considered so far is a non unique assignment of the elements of the starting pile to the elements of the final pile. This leads to the term of non-unique pile problems.

The practical background of such a model is the distinction between the elements of the starting pile and the available positions of these elements in the final pile.

Definition 7 : $R = (G_S = (V_S, A_S),\ G_F = (V_F, A_F),\ f,\ k)$ *is called non-unique pile problem, if and only if*

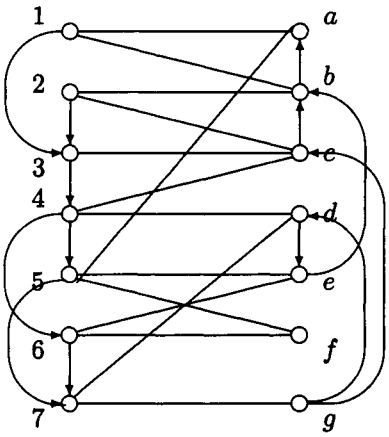
1. *the directed graphs* G_S *and* G_F *represent the starting pile and the final pile, respectively,*

2. $k \geq 0$ *is a natural number,*

3. V_S *and* V_F *are disjoint and it holds* $|V_S| = |V_F|$ *and*

4. f *is a mapping with* $f : V_S \longmapsto 2^{V_F}$.

Such a mapping f also can be interpreted as a set of undirected edges in a bipartite graph consisting of the elements of the starting pile and elements from the final pile. We illustrate this fact in Fig. 7 with the following function $f : \{1,2,3,4,5,6,7\} \longmapsto 2^{\{a,b,c,d,e,f,g\}}$ and the example from Figs. 1 and 2.



$$
\begin{aligned}
f(1) &= \{a,b\},\\
f(2) &= \{b,c\},\\
f(3) &= \{c\},\\
f(4) &= \{c,d\},\\
f(5) &= \{a,e,f\},\\
f(6) &= \{e,f\}.\\
f(7) &= \{d,g\}.
\end{aligned}
$$

Figure 7: Example mapping for the pile problem given by the Figs. 1 and 2 and its representation as undirected arcs in the bipartite graph that corresponds to the given pile

Definition 8 : *Let* $G_S = (V_S, A_S)$ *and* $G_F = (V_F, A_F)$ *be piles and* $f : V_S \longmapsto 2^{V_F}$ *a mapping. Furthermore* V_S *and* V_F *be disjoint. Then the mapping-graph* $G(f) := (V, E)$ *with respect to* f *is defined as follows:*

1. $V := V_S \cup V_F$ *and*

2. $E := \{\{u,v\} : u \in V_S, v \in V_F \text{ and } v \in f(u)\}$.

To transform the starting pile G_S of such a non-unique pile problem to the required final pile G_F it is obviously necessary to find a one-to-one correspondence between the elements of the starting pile and the positions in the final pile.

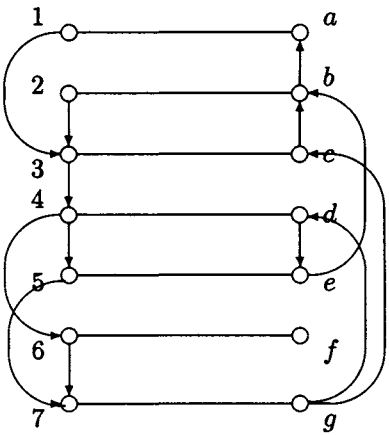
Definition 9 *: A non-unique pile problem*
$R = (G_S = (V_S, A_S),\ G_F = (V_F, A_F),\ f,\ k)$ *is called solvable, if and only if*

1. *there is a unique mapping* $g : V_S \longmapsto V_F$, *such that for all* $v \in V_S$ *holds* $g(v) \in f(v)$ *and*

2. $Q(g) := (G_S,\ G' := (V_S, A),\ k)$ *is a solvable (unique) pile problem with* $A := \{(u, v) : (g(u), g(v)) \in A_F\}$.

Such a correspondence is equivalent to a perfect matching E in $G(f)$. Therefore we use also the notation $Q(E)$ for $Q(g)$.

The existence of such a perfect matching hence is a necessary condition for solving a non unique pile problem R. For the example from Figs. 1 and 2 there is the following function $g : V(G_S) \longmapsto V(G_F)$, such that for all $v \in V(G_S)$ $g(v) \in f(v)$ holds.



$g(1) = a,$
$g(2) = b,$
$g(3) = c,$
$g(4) = d,$
$g(5) = e,$
$g(6) = f,$
$g(7) = g.$

Figure 8: A unique mapping g of f and the corresponding perfect matching in G(f) for the pile problem defined by Figs. 1 and 2

5.1 Choosing a Unique Mapping

Before investigating a given pile problem we have to verify that the given mapping f allows us to find a unique mapping $g \in f$. By [2] a maximum matching for bipartite graphs can be found in $O(|V|^{\frac{1}{2}}|E|)$ steps.

Again we are now interested in the smallest number k of elements that are to be moved to auxiliary places to transform the starting pile of a non-unique pile problem into the final pile with regard to the assignments given by f.

With Theorem 5 it follows immediately that it is NP-complete to decide whether a given non-unique pile problem $R = (G_S, G_F, f, k)$ is solvable or not.

For that reason we like to know a perfect matching E in $G(f)$ for which the resulting unique pile problem $Q(E)$ is solvable with the minimal number of elements to be put aside. To compute this smallest number k we can reduce R to a number of unique pile problems and solve all the concluding unique pile problems. But in general there are exponentially many different perfect matchings in $G(f)$. Therefore determining E in this way is a difficult task. Calculating E is intractable at all:

Theorem 7 *: Given a non-unique pile problem $R := (G_S, G_F, f, k)$. Calculating a perfect matching E of $G(f)$ for which the resulting unique pile problem $Q(E)$ is solvable with a minimal number of elements that need to be moved to auxiliary places is NP-hard.*

5.2 Strategies of Solving

With Theorem 7 it seems to be useful to determine the smallest number of necessary auxiliary places approximately. There are the following two ways:

- determining a perfect matching E of $G(f)$ and calculating the necessary number of auxiliary piles of the resulting unique pile problem $Q(E)$,
- calculating a perfect matching and a corresponding sequence of moves of the elements simultaneously.

In this paper we only want to investigate the first way. It's obviously a good idea to evaluate each of the assigning edges in $G(f)$ and to calculate a minimum weighted perfect matching E for $G(f)$ after that.

There are several criteria to evaluate these edges, for instance:

1. value with regard to the components, the edge is contained in,
2. value with regard to the depth of the vertices the edge is incident with.

5.2.1 Components

Now we introduce the term *component*, which is similar to the well known strongly connected component. We consider the graph $G_{\vec{M}}$, which yields by taking the union of the starting pile and final pile and additionally the edges of G_M which now should be directed in either direction: $G_{\vec{M}} := (V_S \cup V_F,\ A_S \cup A_F \cup \vec{M})$ with $\vec{M} := \{(u,v),(v,u)$ for all $\{u,v\}$ from $G(f)\}$.

Definition 10 *We call a subgraph G of $G_{\vec{M}}$ component of $G_{\vec{M}}$ if and only if every pair of vertices in G is contained in a cycle $C = [v_1,\ldots,v_k,v_{k+1} = v_1]$ of G, such that for all $i \in \{1,\ldots,k-1\}$ from $(v_i,v_{i+1}) \in \vec{M}$ follows $(v_{i+1},v_{i+2}) \notin \vec{M}$, that means in C there are no two consecutive arcs from $\vec{M}$, and G has to be maximal.*

Now $G_{\vec{M}}$ can be divided in such components and we can evaluate the edges in $G(f)$:

Every edge in $G(f)$ the corresponding arc in $G_{\vec{M}}$ lies inside a component will be valued with 1, for arcs that link two different components in $G_{\vec{M}}$ we value the corresponding edge in $G(f)$ with 0.
Note that a division in such components is a division of the set of arcs, there may be vertices that are contained in more than one component.

5.2.2 Difference of Depths

G_S and G_F represent starting pile and final pile, respectively, therefore they are cycle-free. Hence for every vertex v of those graphs we can assign a depth $t(v)$ in either graph:

Definition 11 *: Let G be a cycle-free, directed graph. For every vertex $v \in V(G)$ the depth $t(v)$ is defined as follows:*

$$t(v) := \begin{cases} 0 & ;\ \textit{if } v \textit{ has no predecessor,} \\ \max\{t(v'), v' \textit{ predecessor of } v\} + 1 & ;\ \textit{else.} \end{cases}$$

Every edge $\{v_1,v_2\}$ in $G(f)$ links vertices of V_S with vertices of V_F, therefore these edges can be valued with the difference of the depths of v_1 and v_2: $|t(v_1) - t(v_2)|$.

5.3 Results

In our last section we want to report about results obtained by implementing the methods discussed in Sect. (5.2).

To compare both methods and to get a feeling of the quality of the results we proceeded like in Sect. (4.3): Starting pile and final pile as well

as a graph $G(f)$ – hence a complete non unique pile problem R – were generated randomly. Afterwards we calculated all possible perfect matchings of $G(f)$ and every resulting unique pile problem was solved exactly. At the same time R was solved by the algorithms described in Sects. (5.2.1) and (5.2.2). We investigated 1000 examples, each of which had 20 elements and a density of arcs of 25 per cent. In Table 4 we show the number of exact solutions, solutions that differ about one, two and more than four vertices. Due to the small size of the considered examples (20 elements), the computing times of the heuristic algorithm are not worth mentioning.

Table 4: Quality of both heuristics in comparison for 1000 examples

	number of exact solutions	results differing about one vertex	results differing about two vertices	results differing about more than four vertices
Value by components	140	285	273	61
Value by depth	190	351	248	24

6 Concluding Remarks

The literature about pile problems is quite rarely and many interesting questions in this connection are not investigated until now.
Especially for chain pile problems the number of solvable permutations for more then two auxiliary places up to $n/2$ are unknown. For this, seems, it is necessary to find an other way of proof as the way of case distinction.
This question is also of interest when the additional assumptions 1. - 3. are dropped.
For general pile problems in our investigations was always considered the case that an auxiliary place can take up one element only, but the number of such auxiliary places was not limited. From the point of view of practical applications is also of interest when auxiliary place can take up several elements. This elements can piled up to a stack or to a pile. The number of elements or the height of such an auxiliary pile can be bounded. But also other restrictions can hold like a rule system or preference constraints for the places.
Of large interest is also the case of a bounded number of auxiliary places.

References

[1] Garey MR, Johnson DS (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco

[2] Hopcroft J, Karp RM (1973) An $n^{\frac{5}{2}}$ algorithm for maximum matching in bipartite graphs. SIAM J. Comp. 2:225–231

[3] Kämmerer L (1998) Mathematische Behandlung von Stapelproblemen. Dissertation, Bauhaus-Universität Weimar

[4] Kämmerer L, Hempel L (1997) The towers of hanoi and pileproblems. In: Operations Research Proceedings. Springer-Verlag, Berlin, Heidelberg, New York

[5] Vasilkov D, Hempel L (1996) On the computational aspects of the pile problem. Preprint, Bauhaus-Universität Weimar

[6] Giemsch P, Jellinghaus A (2003) Optimization models for the containership stowage problems. In: Operations Research Proceedings. Springer-Verlag, Berlin, Heidelberg, New York

Risk and Safety Stock Management in Production Planning and Inventory Control with Stochastic Demand and Yield

Karl Inderfurth
Chair for Production and Logistics
Otto-von-Guericke-University Magdeburg

Abstract

Production planning and inventory control is facing challenging risk management problems if it is confronted with uncertainties from both the demand and the process side. By analyzing the respective planning problems with methods of stochastic inventory control it is possible to gain remarkably deep insights into the way how optimal reorder and safety stock management responds to joint demand and yield risks. These insights can be exploited to assess and improve the simple type of risk management rules employed in MRP systems to cope with uncertainties in demand and production yield.

1 Demand and Yield Risk in Production Planning and Control

Risk protection in production planning and control is necessary if uncertainties in the planning environment are of such significance that they cannot be neglected. In production planning major uncertainties refer either to the demand or to the production side. Demand uncertainties are found in all make-to-stock systems and can be regarded as a prevalent source of risk in many production systems. Process uncertainties are found in all cases where production processes lack complete reliability. Despite of all efforts in quality improvement programs, this is found in many industries and results in more or less considerable yield losses. In semiconductor manufacturing these losses can even exceed 80 % (see Nahmias 2005, p. 385). This problem is also significant in the fast-growing remanufacturing industry where limited knowledge of the quality of used products causes high yield uncertainty in the disassembly processes (see Guide 2000). When yield randomness coincides with uncertainty from the demand side, we face quite complex production planning and inventory control problems which demand appropriate means of risk protection.

The traditional measure to protect against uncertainties in production planning and control systems is to incorporate safety stocks and/or safety lead times as

buffering concepts (see Vollmann et al. 2005, p. 203). However, safety lead times are only advocated for timing uncertainties which are usually found in supply processes. Demand uncertainties regularly appear as quantity uncertainties which are recommended to be coped with by safety stocks. If yield losses occur in production systems, a natural way to incorporate these losses in production planning is the usage of scrap allowances when calculating requirements. These risk protection measures can also easily be applied in advanced MRP systems.

Considering a standard MRP record like in Table 1, demand uncertainty is reflected by [1] deriving gross requirements from deterministic demand forecasts and by [2] incorporating a safety stock in the 'projected on hand' calculation to protect against forecasting errors. Yield uncertainty is dealt with by [3] calculating the scheduled receipts from mean production yield and by [4] inflating the net requirements in the MRP record by respective scrap allowances. Regarding the yield risk, there is no clear advise from literature or from practice how to incorporate it in the MRP concept. Recommendations are found to do it by adjusting the scrap allowance (see Nahmias 2005, p. 385) or the safety stock (see Silver et al. 1998, p. 613), respectively.

Table 1: Standard MRP record

Period		1	2	3	4
[1] Gross requirements		20	10	5	15
[3] Scheduled receipts		8			
[2] Projected on hand	14	2	0	0	0
[4] Net requirements		0	8	5	15
Planned order receipts			8	5	15
Planned order releases		8	5	15	

An additional way to take uncertainty into account when using MRP is to apply it in a rolling horizon framework with regular update of all relevant forecasts. Under these conditions MRP order release decisions behave just like reorder decisions which follow a stochastic inventory control rule (see Axsäter/Rosling 1994, Inderfurth/Jensen 1999). For the above situation this control rule can be described as

$$p = P(x) = \begin{cases} p(x) = YF \cdot (CS - x) & \text{for} \quad x < CS \\ 0 & \qquad\quad x \geq CS \end{cases} \tag{1}$$

with p as order release quantity and x as inventory position (net inventory plus all scheduled receipts). YF is a yield correction factor for scrap allowances

which compensates for yield losses and *CS* is a critical stock level which comprises the expected requirements within the risk period (production lead time plus control period) and a safety stock. Thus, it turns out that the MRP rule is a critical stock policy (i.e., no order is released if the inventory position exceeds a critical level *CS*) with a linear reorder function $p(x)$.

When using such a control rule the safety stock *ss*, defined as expected stock level at the end of a period, is given by

$$ss = x + \mu_Z \cdot P(x) - \mu_D \tag{2}$$

where μ_D denotes the expected demand and μ_Z stands for the expected yield rate. For the linear MRP-rule in (1) the safety stock turns out to be a linear function of the inventory position (in the regular case of $x \leq CS$)

$$ss = (1 - \mu_Z \cdot YF) \cdot x + [\mu_Z \cdot YF \cdot CS - \mu_D] \quad . \tag{3}$$

From (3) it is obvious that only for a yield correction factor that exactly compensates for the expected yield loss, i.e. $YF = 1/\mu_Z$, the safety stock will be independent of the inventory level, amounting to

$$ss = CS - \mu_D \quad . \tag{4}$$

This reveals that, if yield risk is completely taken into consideration by adapting the critical stock level *CS*, the safety stock is a constant like in the case without yield risk and with uncertainty only from the demand side (see Silver et al. 1998, p. 279).

Up to now there is only little knowledge if this type of linear MRP-based control rule is reasonable in case of random demand and yield and, if so, how the control parameters *YF* and *CS* should be determined in order to protect against the respective risks appropriately. Specifically, it is not clear if risk protection should be performed solely by a safety stock like in (4) or jointly by a safety stock and safety yield correction, relying on a risk adjustment concerning both parameters *CS* and *YF*. This is also due to the fact that there is no clear picture of how the yield risk will affect the safety stock in (2) when an optimal control is applied.

This paper aims to give theory-based answers to these questions by exploiting knowledge from respective stochastic inventory control models. Since the focus of this analysis is on risk protection, we will not take into account lotsizing as-

pects. Additionally, we assume that both demand and yield risk (described by the variances of respective probability distributions) do not depend on time and on demand/production volume. This allows to rely on stationary models of stochastic inventory control.

2 Insights from Stochastic Inventory Control

The inventory control model related to the advanced MRP model described in Section 1 is the periodic review model with proportional costs under random yield and random demand which first was analyzed in detail in Gerchak et al. (1988) and in Henig/Gerchak (1990). They consider a pure cost model with constant per unit cost for production (c), inventory holding (h) and backlogging unmet demand (v). Demand follows a stationary demand distribution function $F_D(.)$ with expectation μ_D and variance σ_D^2. The lead time is fixed to zero. The yield is assumed to be proportional to the order quantity p with a stochastic proportionality factor Z (with $0 \le Z \le 1$) which is distributed according to a function $F_Z(.)$ with expectation μ_Z and variance σ_Z^2 (see Yano/Lee 1995 for justification of this type of yield model). So μ_Z is the mean yield rate and σ_Z is a measure of the yield risk. From the above literature it is known that for the general multi-period case the optimal control rule has the following structure

$$P^*(x) = \begin{cases} p^*(x) > 0 & \text{if} \quad x < CS^* \\ 0 & \qquad\quad x \ge CS^* \end{cases} \quad . \tag{5}$$

Like the MRP-rule in (1) the optimal policy is a critical stock rule, but the optimal reorder function is in general non-linear with a first derivate for which $dp^*(x)/dx \le 1$ holds. In detail, the optimal control rule is implicitly given by

$$\int_0^1 z \cdot F_D(x + z \cdot p) \cdot dF_Z(z) = \mu_Z \cdot F_D(CS) \quad . \tag{6}$$

The critical stock level CS^* and the reorder function $p^*(x)$ will depend on both the demand and yield risk in a multi-period situation, but can only be evaluated numerically.

It should be noted that in the case of deterministic yield losses, i.e. for $\sigma_Z = 0$, the optimal control rule degenerates to the standard order-up-to rule which is well-known from inventory theory for stochastic demand. Under these circum-

stances the reorder function, here denoted by $p^*_{RF}(x)$ to indicate the (yield) risk-free case, is a linear one

$$p_{RF}(x) = \frac{1}{\mu_Z} \cdot \left(CS^* - x \right) \quad . \tag{7}$$

A special case which allows for analytical insights in the general control rule for stochastic yield in (5) is the single-period case. Here, it can be shown that the critical stock is an α-quantile of the demand distribution according to a newsvendor solution

$$CS^* = F_D^{-1}(\alpha) \qquad \text{with} \qquad \alpha = \frac{v - c/\mu_Z}{v + h} \quad . \tag{8}$$

As in the newsvendor situation, cost ratio α can be interpreted as a service measure. The larger α is given, the higher is the probability of not being out-of-stock which under stochastic yield also depends on the yield rate distribution $F_Z(.)$. It turns out that the critical stock level in (8) is not affected by the yield risk σ_Z. However, yield uncertainty will always have an impact on the $p^*(x)$ function. This function can only be evaluated analytically for specific demand and yield rate distribution functions like uniform or exponential ones (see Gerchak et al. 1988). If demand D and yield rate Z are both uniformly distributed with $D \in [0, D^+]$ and $Z \in [0, Z^+]$, the optimal reorder function turns out to be piecewise convex and linear as follows (see Inderfurth 2004):

$$p^*(x) = \begin{cases} \dfrac{1}{Z^+} \cdot \sqrt{\dfrac{\left(D^+ - x \right)^3}{3 \cdot \left(D^+ - CS^* \right)}} & \text{if} \quad 0 \le x \le (3\alpha - 2) \cdot D^+ \\ \dfrac{3}{2 \cdot Z^+} \cdot \left(CS^* - x \right) & \text{if} \quad (3\alpha - 2) \cdot D^+ \le x \le CS^* \end{cases} \tag{9}$$

with $CS^* = \alpha \cdot D^+$.

From these results we find an answer to the first research question referring to the structure of the optimal control rule. It turns out that a linear rule, as used in MRP-based approaches, is only optimal under very specific conditions with regard to the planning horizon (single period), demand and yield rate distribution (uniform) and the service value α. From (9) it can easily be seen that the optimal order release function $p^*(x)$ is only completely linear over the whole

range of $0 \le x \le CS^*$ if $\alpha \le 2/3$ holds. Since service values of less than 67 % are hardly desirable under general circumstances, it can be concluded that under practical conditions we have to be aware that the optimal control rule will almost never be a linear one. In order to learn more about the general relationship between optimal risk control and safety stock management, we will more deeply evaluate the analytical insights from the optimal control rule in (9).

3 Yield Risk and Safety Stock Under Uniform Yield and Demand

In exploiting the knowledge of the optimal reorder rule given in (9) we first consider the case of low service value $(\alpha \le 2/3)$ where the order release function $p^*(x)$ is completely linear. Thereafter the situation with a non-linear control rule (for $\alpha > 2/3$) will be analyzed.

Case I: $\alpha \le 2/3$

From (9) we can derive an optimal reorder rule which because of its linearity will be denoted by $p_L^*(x)$

$$p_L^*(x) = YF^* \cdot \left(CS^* - x\right) \tag{10}$$

with $YF^* = \frac{3}{2} \cdot \frac{1}{Z^+}$.

From the underlying uniform distributions we have $Z^+ = 2 \cdot \mu_Z$ and $D^+ = 2 \cdot \mu_D$ so that we find the optimal yield correction factor YF^* to be smaller than in the case without yield risk described in (7)

$$YF^* = \frac{3}{4} \cdot \frac{1}{\mu_Z} < \frac{1}{\mu_Z} \quad . \tag{11}$$

According to the safety stock definition in (2) this results in a safety stock which is not constant, but depends on the inventory level x

$$ss_L^*(x) = \frac{1}{4} \cdot x + \left(\frac{3}{2}\alpha - 1\right) \cdot \mu_D \quad . \tag{12}$$

From (12) we learn that, depending on stock level x, the optimal safety stock might be negative even if service value α is larger than 50 % , a result which is

not found in the yield risk-free case. In detail, there exists an α-dependent critical stock $\bar{x}$ with

$$ss_L^*(x) < 0 \qquad \text{for} \qquad x < \bar{x}(\alpha) = 2 \cdot \mu_D (2 - 3\alpha) \quad . \tag{13}$$

The reason behind this effect is that according to (11) the yield risk reduces the yield correction factor compared to the risk-free situation, a counter-intuitive response which will be discussed later. A further result in this context is that for small service values of $\alpha \leq 2/3$, independent of stock level x, the optimal safety stock from (12) will never be larger than the safety stock ss_{RF} from the risk-free rule in (7)

$$ss_L^*(x) \;\leq\; ss_{RF}^* \;=\; (2\alpha - 1) \cdot \mu_D \quad . \tag{14}$$

Case II: $\alpha > 2/3$

For sufficiently large service value α we face the situation of a non-linear reorder function $p^*(x)$ which, as described above, will be typical for practical situations. Figure 1 presents a picture of the optimal reorder function derived in (9) which is strictly convex for small and linear for large inventory levels x.

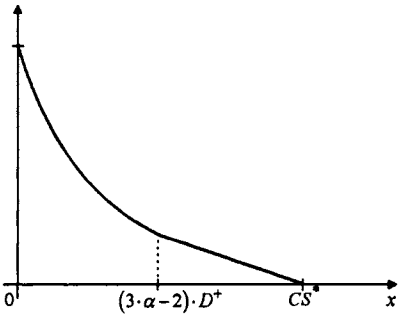


Figure 1: Optimal reorder function for $\alpha > 2/3$

In this case, the respective safety stock according to (3) can be formulated as

$$ss^*(x) = \begin{cases} \dfrac{1}{2}x + \left(\mu_D - \dfrac{1}{2}x\right) \cdot \left[\sqrt{\dfrac{2\mu_D - x}{6\mu_D(1-\alpha)}} - 1\right] & \text{if} \quad 0 \leq x \leq (6\alpha - 4) \cdot \mu_D \\ \dfrac{1}{4}x + \left(\dfrac{3}{2}\alpha - 1\right) \cdot \mu_D & \text{if} \quad (6\alpha - 4) \cdot \mu_D < x \leq CS^* \end{cases} \quad . \tag{15}$$

From (15) it can easily be derived that the optimal safety stock always will be positive ($ss^*(x) > 0$) irregardless of the size of inventory level x:

$$ss^*(x) > 0 \qquad \text{for} \qquad 0 \le x \le CS^* \quad . \tag{16}$$

Additionally, it can be shown from (15) that the safety stock is continuously decreasing with increasing inventory level, i.e. $d\,ss(x)/dx < 0$. That means that, different from the situation with small service values ($\alpha < 2/3$) in (12), the risk is perceived to increase the smaller the starting inventory is. This seems intuitive since a smaller inventory level is associated with a higher production order which itself bears more yield risk than a smaller one. The highest risk in this sense is connected with zero inventory ($x = 0$) for which the safety stock in (15) reaches its maximum $ss^*_{\max}$ and turns out to be

$$ss^*_{\max} = \mu_D \cdot \left[\sqrt{\frac{1}{3(1-\alpha)}} - 1 \right] \quad . \tag{17}$$

From (17) we see that the (maximum) safety stock is increasing with increasing service value α (as expected), starting with $ss^*_{\max} = 0$ at the lower bound for α in Case II ($\alpha = 2/3$) and tending to infinity ($ss^*_{\max} \to \infty$) as α is approaching its upper bound ($\alpha \to 1$). The latter effect is caused by the fact that due to the assumed yield rate minimum of zero the production order must tend to infinity in order to avoid any shortage as demanded by a 100 % service value. Thus, what we learn is that in case of $\alpha > 2/3$ the safety stock incorporates some yield risk adjustment which depends on service value α and inventory level x. The maximum adjustment is given for $x = 0$ and can be used for a comparison with the risk-free safety stock ss^*_{RF} to evaluate whether the risk adjustments tends to increase or decrease the safety stock. By comparing $ss^*_{\max}$ from (17) with ss^*_{RF} from (14) it is easy to see that $ss^*_{\max} < ss^*_{RF}$ for $\alpha = 2/3$ and $ss^*_{\max} > ss^*_{RF}$ for $\alpha \to 1$.

Thus, a critical service value $\hat{\alpha}$ must exists which divides a region ($\alpha < \hat{\alpha}$) where yield risk is responded by a decrease of the safety stock from a region ($\alpha > \hat{\alpha}$) where the opposite holds. This critical value $\hat{\alpha}$ is found from equating $ss^*_{\max}$ and ss^*_{RF} which leads to $\hat{\alpha} \cong 0.9$.

In Figure 2 this situation is visualized by presenting two examples of the (bold-line) reorder function $p^*(x)$ which refer to the case $\alpha < \hat{\alpha}$ and $\alpha > \hat{\alpha}$, respectively. For comparison the (dashed-line) yield risk-free function $p_{RF}(x)$ is also included.

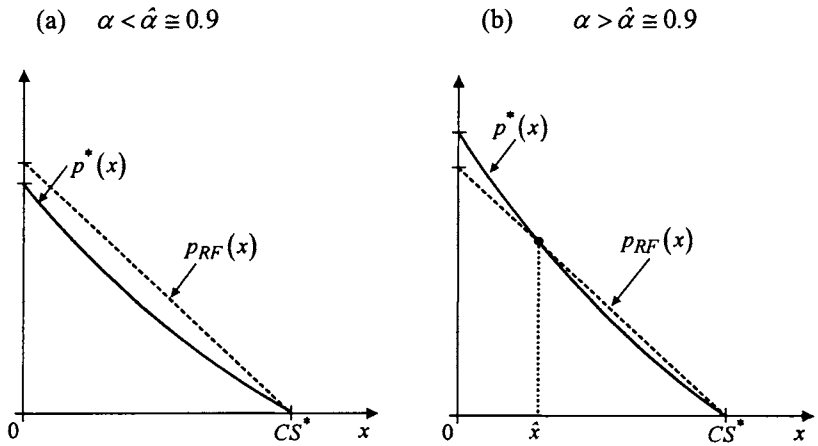


Figure 2: Optimal and risk-free reorder functions under uniform distributions

Apparently, for service values $\alpha < \hat{\alpha}$ a yield risk always diminishes the order release quantity compared to a risk-free situation. However, if service value α exceeds the critical $\hat{\alpha}$-level, the order release quantity can be larger than in the case without yield risk. This will always happen if the inventory level x is sufficiently small so that a relatively large production quantity is released. The critical inventory level $\hat{x}$ which divides the two regions of opposite risk adjustment is found by equating $p^*(x)$ and $p_{RF}(x)$. Obviously, this inventory level depends on the service value α so that $\hat{x} = \hat{x}(\alpha)$ holds.

Summarizing, the results from analyzing the yield risk impact in the case of a single-period problem with uniformly distributed demand and yield rate are presented in the following Table 2.

Table 2: Yield risk effects under uniform distributions

Service value	Safety stock	Yield risk effect
$\alpha \le 0.\overline{6}$	$ss^*(x) < ss_{RF} \quad \forall\ x$ with $ss^*(x) < 0 \quad$ if $x < \bar{x}(\alpha)$	*ss* reduction
$0.\overline{6} < \alpha \le 0.9$	$0 < ss^*(x) \le ss_{RF} \quad \forall\ x$	*ss* reduction
$\alpha > 0.9$	$ss^*(x) \le ss_{RF} \quad$ if $x \ge \hat{x}(\alpha)$ $ss^*(x) > ss_{RF} \quad$ if $x < \hat{x}(\alpha)$	*ss* reduction *ss* increase

The most remarkable result of this analysis is the fact we face ambiguity of yield risk impact. Depending on the desired level of service and on the initial inventory level, yield risk can either increase or decrease the optimal safety stock and reorder quantity.

4 General Insights into the Yield Risk Impact

From the analysis in Section 3 two major questions arise. First, can the specific results summarized in Table 2 be generalized to other planning situations with stochastic yield and demand? Second, how can the ambiguity of yield risk effects be explained?

Concerning the planning situation, one of the restrictive assumptions in the previous analysis is that of uniformly distributed yield rate and demand. This assumption causes the specific type of optimal control rule given in (9). However, there is no obvious reason why differently shaped distribution functions (like e.g. in case of normal or beta distribution) should result in $p^*(x)$ functions which are not able to generate yield risk ambiguity effects as shown in Figure 2. Several numerical experiments (see Transchel 2004) support this supposition.

A second restrictive assumption is that of a single-period planning situation. An extension to the multi-period case does not affect the policy structure, although it results in a shift of the critical stock level *CS*. Given that only demand risk exists it is well known from inventory theory (see Neumann 1993, p. 659) that in case of an infinite planning horizon and demand backlogging this shift represents a stock level increase from the α-quantile in (8) to a β-quantile given by

$$CS^* = F_D^{-1}(\beta) \qquad \text{with} \qquad \beta = \frac{v}{v+h} \quad . \tag{18}$$

If also stochastic yield has to be taken into account, the critical stock level in the multi-period case will additionally be influenced by the yield rate distribution (see Henig/Gerchak 1990). There is, however, no compelling argument why this influence should lead to disappearance of yield risk ambiguity.

Thus, there must be some general reason behind the ambiguous yield risk impact which specifically can explain the counter-intuitive effect that order quantity and safety stock can decrease when yield risk increases. For investigating yield risk effects we must consider the economic rational behind building up a safety stock for risk protection. In a pure demand risk situation the critical stock level CS^* (and thereby the safety stock ss^*) is determined in such a way that the expected overage and underage cost $L(CS)$ after demand realization (depending on the ordering, holding and shortage cost parameters) is minimized (see Silver et al. 1998, pp. 385-388). Under cost assumptions as made in Section 2 cost function $L(CS)$ turns out to convex with a minimum for $CS > 0$ as depicted in Figure 3.

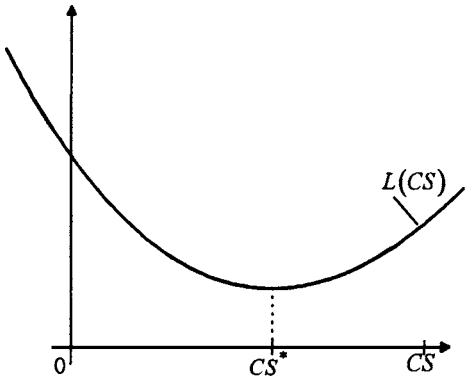


Figure 3: Function of expected overage and underage cost

The optimal cost trade-off results in a quantile solution as described in (8) or (14), respectively. If additionally the yield turns out to be stochastic, the desired critical stock CS^* cannot be reached with certainty by an ordering decision. Thus, an undershoot or overshoot of CS^* can occur which will lead to an increase in the expected costs $L(CS)$. The higher the yield risk, the larger the expected deviation from CS^* will be. The appropriate protection against this additional risk of deviation from CS^* now depends on the whether an undershoot causes higher expected costs $L(CS)$ than an equal-sized overshoot or if the opposite holds. In the former case which is found for high v/h-cost ratios (and equiva-

lently for high service values) the optimal reaction is to increase the order quantity while in the latter case yield risk should be responded by decreasing in the order size. So it is the slope of the $L(CS)$-function, specifically in the neighborhood of the critical level CS^*, which determines the yield rate reaction. The overshoot/undershoot is also affected by the order size necessary to bring the initial inventory up to the critical stock level, because a stochastically proportional yield model is assumed. So a smaller starting inventory causes a higher order level which in turn results in a higher yield risk. This effect contributes to the fact that an increase in safety stock is mainly observed in cases of high service values and low inventory levels.

Summarizing, we see that the optimal production policy in general systems with stochastic yield and demand is not only quite complex, but is also responding to yield risk in such a form that it is jointly covered with demand risk by both fixing the critical stock level CS^* and determining the shape of the order control rule $p^*(x)$. Additionally, there is no unique impact of yield risk on the safety stock which itself is variable because it depends an the starting inventory of each period. Obviously, these complications make it critical to implement a simple linear control rule as described in Section 1. On the other hand, for practical application (like in MRP systems) only such simple two-parameter rules seem to be employable. So it is important to know how the respective parameters YF and CS should be determined appropriately and what performance can be expected from such a linear reorder rule.

5 Yield Risk Management by Linear Control Rules

All theory-based approaches for developing linear approximations for the optimal control rule $p^*(x)$ in (5) (see Bollapragada/Morton 1999, Zipkin 2000, Henig/Gerchak 1990) refer to a stationary situation with infinite planning horizon for which a myopic policy is employed. According to the standard model without yield uncertainty a newsvendor-type solution for the critical stock level CS is advocated resulting in a β-fractile of the demand distribution as described in (18).

The linear approximation approaches primarily differ in the way how yield uncertainty is taken into consideration. Henig/Gerchak (1990) and Zipkin (2000) determine the critical stock CS without reflecting yield risk by just choosing it according to (18). Under normality assumption this results in a critical stock CS_1 and a safety stock ss_1 given by

$$CS_1 = \mu_D + ss_1 \qquad \text{and} \qquad ss_1 = N^{-1}(\beta) \cdot \sigma_D \tag{19}$$

where $N(.)$ denotes the standard normal distribution function.

The yield rate uncertainty then is completely taken into account by adjusting the yield correction factor for the yield variability using the following formula:

$$YF = \frac{1}{\mu_Z + \sigma_Z^2 / \mu_Z} \quad . \tag{20}$$

The mean yield rate adjustment in (20) is derived from specific approximation procedures in Henig/Gerchak (1990) and Zipkin (2000), respectively. The corresponding linear control rule in this case, denoted by $p_{L1}(x)$, is given by: $p_{L1}(x) = YF_1 \cdot (CS_1 - x)$.

A limited number of numerical tests indicate that this linear approximation performs quite unsatisfactorily and deteriorates with increasing yield risk (see Transchel 2004). From the analytical investigation of the single-period model for uniform distributions this result is not surprising. The yield factor $(2/3 \cdot Z^+)^{-1}$ in the linear part of the $p^*(x)$ function in (9) just coincides with YF_1 in (20) if the yield rate is uniformly distributed in $[0, Z^+]$. This reorder rule systematically underestimates the impact of yield risk, especially for high service parameters and low inventory levels. Therefore, it also turns out (see Transchel 2004) that this kind of linear approximation often performs worse than a respective linear rule without risk adjustment as it is given by the risk-free reorder function in (7).

An alternative approach for providing a linear approximation to $p^*(x)$ in (5) has been developed in Bollapragada/Morton (1999). Here, different from the former approach, the yield risk is not considered by adjusting the yield correction factor, which is chosen to be $YF_2 = 1/\mu_Z$, but by modifying the critical stock level *CS*.

Basically, Bollapragada/Morton (1999) develop an approximation to the non-linear $p^*(x)$ function and construct a linear function with slope $-1/\mu_Z$ such that it intersects the optimal reorder function at a stock level equal to the safety stock *ss*. Thus, the linear and the optimal reorder function result in the same planned-order decision when the inventory equals the expected stock level by the end of a period. From this analysis it can be derived that under normality assumption a critical stock CS_2 is given as follows:

$$CS_2 = \mu_D + ss_2 \qquad \text{and} \qquad ss_2 = N^{-1}(\beta) \cdot \sqrt{\sigma_D^2 + \mu_D^2 \cdot \frac{\sigma_Z^2}{\mu_Z^2}} \qquad (21)$$

From (21) it can be seen that the yield risk always results in an increase of the safety stock as it seems reasonable for fairly high service parameters according to the single-period analysis. The respective linear control rule with this approach, denoted by $p_{L2}(x)$, is given by $p_{L2}(x) = YF_2 \cdot (CS_2 - x)$.

Figure 4 displays the results of a numerical investigation (see Transchel 2004) and shows how the linear control rule $p_{L2}(x)$ behaves compared to the optimal reorder function $p^*(x)$ as well as to the linear rule $p_{RF}(x)$ with CS from (19) and without any yield risk adjustment (i.e. $YF = 1/\mu_Z$). The curves in Figure 2 refer to a problem with service parameter $\beta = 0.8$ and with normally distributed demand and yield rate whose coefficients of variation are equal to 0.2.

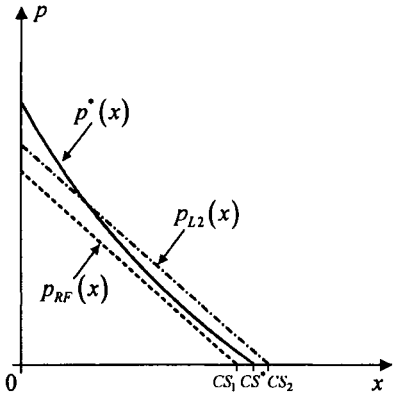


Figure 4: Reorder rules in the multi-period infinite-horizon case

From Figure 4 it can be seen that $CS_1 < CS^* < CS_2$ and that $p_{RF}(x) < p^*(x)$. This finding turns out to be typical for many situations so that it can be expected that the linear rule $p_{L2}(x)$ yields a better approximation than $p_{RF}(x)$. This is confirmed by a numerical performance study in Bollapragada/Morton (1999). In this study, it additionally is revealed that the linear reorder rule $p_{L2}(x)$ is near-optimal over a wide range of parameter values. Thus, the $p_{L2}(x)$ rule can be

considered to be a recommendable theory-based approach for being integrated in a MRP concept. However, in several cases of high β-values ($\beta \geq 0.95$) cost deviations of more than 10 % above optimum have been observed. This effect very likely is caused by the fact that, as shown in the analysis of service level impact in Section 3, the optimal reorder rule turns out to have an increasingly non-linear shape as the service parameter increases.

6 Conclusions

Risk management in production planning and inventory control is a major challenge when demand and yield uncertainties have to be taken into consideration simultaneously. From theory of stochastic inventory control we get many useful insights into the complex interaction of stochastic impacts from the demand and process side and appropriate measures to respond to the respective risks. Specifically the yield risk impact sometimes leads to counter-intuitive advice for safety stock management. Upon deeper examination, however, these policies can be seen as reasonable. Furthermore, general insights from the stochastic analysis can be used to assess different approaches of simple risk protection by applying a yield factor and safety stock parameter in extended MRP systems.

It can be shown that under these types of approaches linear reorder rules are generated. Within these policies reorder rule $p_{L2}(x)$ turns out to be superior, but still retains some deficiencies concerning its performance under certain circumstances. The lack of performance of the linear control rule $p_{L2}(x)$ in case of very high service levels may be caused by the fact that yield uncertainty is only taken into account by adjusting the safety stock, but not by correcting the mean yield factor. It may be presumed that by incorporating the yield risk in both safety stock and yield factor determination a better performance can be achieved. There is, at present, no sound theory from stochastic inventory control on how this should be done, leaving it can open matter for further research.

References

[1] Axsäter S, Rosling K (1994) Multi-level production-inventory control: Material requirements planning or reorder point policies? European Journal of Operational Research 75:405–412

[2] Bollapragada S, Morton T (1999) Myopic heuristics for the random yield problem. Operations Research 47:713–722

[3] Gerchak Y, Vickson R, Parlar M (1988) Periodic review production models with variable yield and uncertain demand. IEE Transactions 20:144–150

[4] Guide Jr. VDR (2000) Production planning and control for remanufacturing: industries practices and research needs. Journal of Operations Management 18:467–483

[5] Henig M, Gerchak Y (1990) The structure of periodic review policies in the presence of random yield. Operations Research 38:634–643

[6] Inderfurth K, Jensen T (1999) Analysis of MRP policies with recovery options. In: Leopold-Wilburger U, Feichtinger G, Kistner KP (eds.) Modelling and Decisions in Economics. Springer, Heidelberg/New York, pp. 189–228

[7] Inderfurth K (2004) Analytical solution for the single-period inventory problem with uniformly distributed yield and demand. Central European Journal of Operations Research 12:117–127

[8] Nahmias S (2005) Production and Operations Analysis, 5th edition. McGraw-Hill, New York

[9] Neumann K, Morlock M (1993) Operations Research. Hanser, München/Wien

[10] Silver EA, Pyke DF, Peterson R (1998) Inventory Management and Production Planning and Scheduling, 3rd edition. John Wiley & Sons, New York

[11] Transchel S (2004) Zur Güte linearer Dispositionsregeln für ein Produktions-Lagerhaltungs-Problem mit stochastischer Nachfrage und stochastischer Produktionsausbeute. Diploma Thesis, University of Magdeburg, Magdeburg

[12] Vollmann TE, Berry WL, Whybark DC, Jacobs FR (2005) Manufacturing Planning and Control Systems for Supply Chain Management, 5th edition. McGraw-Hill, New York

[13] Yano CA, Lee HL (1995) Lot sizing with random yields: A review. Operations Research 43:311–334

[14] Zipkin PH (2000) Foundations of inventory management. McGraw-Hill, New York

Economies of Scale in Hub & Spoke Network Design Models: We Have It All Wrong

Alf Kimms

Lehrstuhl für ABWL, insbes. Produktion und Logistik
Technische Universität Freiberg

1 Traditional Models for Hub & Spoke Network Design

The hub & spoke network design problem is a strategic logistics planning problem with applications for airlines, telecommunication companies, computer networks, postal services, and trucking companies, for example. Basically, the problem in all these applications is that for a given set $V = \{1, \ldots, n\}$ of nodes (airports, computers, post offices, depots, ...) goods must be transported between possibly every pair of nodes. Direct connections between every pair of nodes would result in $n(n-1)$ linkages which is impractically high and economically non–profitable. Consider, for instance, an airline that serves several airports worldwide. Offering non–stop flights between every pair of airports would require a huge amount of planes and crews and many empty seats on board could be observed for many connections. In such settings, it turns out to be reasonable to install one or more so–called hub locations where direct links are then available to hub nodes as indicated in figure 1 where nodes 3, 6, and 9 are assumed to be hubs. Transporting goods from, say, node 1 to node 11, can then be done via hubs 3 and 6.

Roughly speaking, the network design problem at hand can be couched as follows: Given a graph with node set V and edge set $E = V \times V$, select one or more nodes from V to become hub nodes and select some edges from E to become transportation links. For each pair of nodes $(i, j) \in V \times V$ we have a quantity $q_{ij} \in \boldsymbol{R}_{\geq 0}$ that is to be transported from node i to node j. Established models assume that the unit cost of transportation using an edge e is $c_e \in \boldsymbol{R}_{\geq 0}$ and that, if e connects two hub nodes, a discount can be gained such that the unit cost of transportation using edge e is $\alpha c_e \in \boldsymbol{R}_{\geq 0}$ with $0 \leq \alpha \leq 1$. We will question these cost assumptions in section 3 and discuss alternatives. Note that the costs may be asymmetric, i.e. $c_{ij} = c_{ji}$ may not be true.

For the hub & spoke network structure one can wish to have specific characteristics that define the design problem to be solved. Some usual and

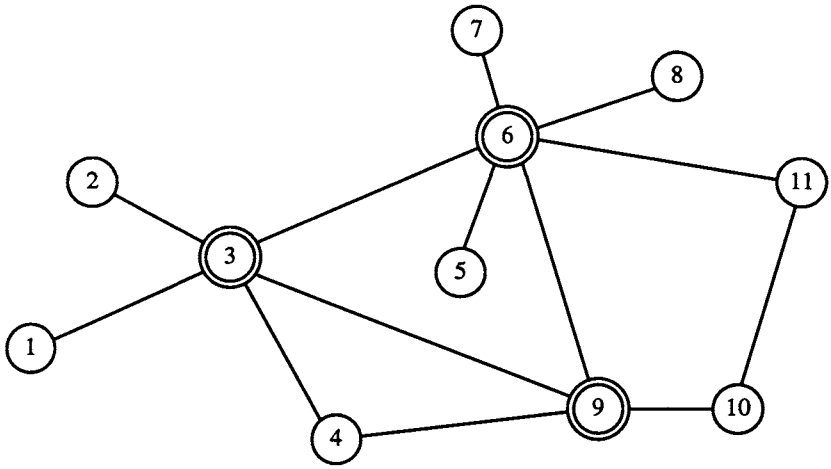


Figure 1: An Illustration of a Hub & Spoke System

basic features are the following:

- What determines the number of hubs?
  - Hub location problems with fixed hub costs: Installing a hub at node h incurs a fixed cost $f_h \in \mathbb{R}_{\geq 0}$. The number of hubs is a result of the planning process.
  - p–Hub median problems: The number p of hubs is predefined. Fixed hub costs are usually assumed to be the same for all nodes so that they can be ignored for the purpose of optimization.
- How are the non–hub nodes connected?
  - Single allocation: Each non–hub node is allocated to a unique hub. In a single allocation network, node 4 (figure 1) would not be allowed to have a direct link to two hubs.
  - Multiple allocation: Non–hub nodes (like node 4 in figure 1) may be connected to several hubs.
  - Direct services: Non–hub nodes may have a direct connection like nodes 10 and 11 in figure 1.

Throughout this paper, we assume that the set of hubs is fully meshed, i.e. the subgraph induced by the hub nodes is complete, and that the objective for designing the network is to minimize the sum of relevant costs.

Mathematically, typical hub & spoke network design problems can be stated as follows (see Campbell, 1994b, for several models in this area):

For the single allocation case without direct services, define a binary decision variable y_{ih} which is equal to one, if and only if node h is the hub that node i is allocated to, and zero, otherwise. A value $y_{hh} = 1$ indicates that node h is a hub. Additionally, a real–valued variable x_{ijhk} is used to model the fraction of flow that is routed from node i to node j via hubs h and k in that order.

$$\text{Min } F_1(\mathbf{x},\mathbf{y}) = \sum_{h=1}^{n} f_h y_{hh} + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{h=1}^{n}\sum_{k=1}^{n}(c_{ih} + \alpha c_{hk} + c_{kj})q_{ij}x_{ijhk} \tag{1}$$

s.t.

$$y_{ih} \le y_{hh} \qquad i,h = 1,\ldots,n;\ i \ne h \tag{2}$$

$$\sum_{h=1}^{n} y_{ih} = 1 \qquad i = 1,\ldots,n \tag{3}$$

$$\sum_{h=1}^{n}\sum_{k=1}^{n} x_{ijhk} = 1 \qquad i,j = 1,\ldots,n \tag{4}$$

$$\sum_{k=1}^{n} x_{ijhk} \le y_{ih} \qquad i,j,h = 1,\ldots,n \tag{5}$$

$$\sum_{h=1}^{n} x_{ijhk} \le y_{jk} \qquad i,j,k = 1,\ldots,n \tag{6}$$

$$x_{ijhk} \ge 0 \qquad i,j,h,k = 1,\ldots,n \tag{7}$$

$$y_{ih} \in \{0,1\} \qquad i,h = 1,\ldots,n \tag{8}$$

The objective (1) minimizes the sum of fixed costs for installing the hubs plus transportation costs. Due to (2), a non–hub node i can only be linked to a hub h, if node h is indeed a hub. (3) makes sure that every node is linked to exactly one hub. Because of (4), all quantities are shipped. And because of (5) and (6) a path from node i to node j via nodes h and k, respectively, can only be used, if h and k are hub nodes. It is noteworthy to say that it is a common (e.g., if it is assumed that costs fulfill $c_{ij}+c_{jk} \ge c_{ik}$) that while moving from a node i to a node j at most two other (hub) nodes are passed in-between. (7) and (8) define the decision variables. Note that there exists an optimal solution with x being integral.

By adding the constraint

$$\sum_{h=1}^{n} y_{hh} = p$$

and assuming the special case $f_h = 0$ for all h, we would get a model for the p–hub median problem.

For representing the multiple allocation case, a one–index binary variable y_h is sufficient. The variable y_h is one, if node h is a hub, and zero, otherwise.

$$\text{Min } F_2(\mathbf{x},\mathbf{y}) = \sum_{h=1}^{n} f_h y_h + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{h=1}^{n}\sum_{k=1}^{n}(c_{ih} + \alpha c_{hk} + c_{kj}) q_{ij} x_{ijhk} \tag{9}$$

s.t.

$$\sum_{h=1}^{n}\sum_{k=1}^{n} x_{ijhk} = 1 \qquad i,j = 1,\ldots,n \tag{10}$$

$$\sum_{k=1}^{n} x_{ijhk} + \sum_{k=1,k\neq h}^{n} x_{ijkh} \leq y_h \qquad i,j,h = 1,\ldots,n \tag{11}$$

$$x_{ijhk} \geq 0 \qquad i,j,h,k = 1,\ldots,n \tag{12}$$

$$y_h \in \{0,1\} \qquad h = 1,\ldots,n \tag{13}$$

It should be sufficient to explain (11). It makes sure that a flow from node i to node j can pass a node h only if h is a hub node. In this case either i or j (or both) would be linked to h.

By adding the constraint

$$\sum_{h=1}^{n} y_h = p \tag{14}$$

and assuming the special case $f_h = 0$ for all h, we would get a model for the p–hub median problem.

If direct services are allowed, an additional real–valued variable x'_{ij} for the fraction of flow from node i to node j leads to the following model:

$$\text{Min } F_3(\mathbf{x}',\mathbf{x},\mathbf{y}) = \sum_{h=1}^{n} f_h y_h + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{h=1}^{n}\sum_{k=1}^{n}(c_{ih} + \alpha c_{hk} + c_{kj}) q_{ij} x_{ijhk} + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} q_{ij} x'_{ij} \tag{15}$$

s.t.

$$x'_{ij} + \sum_{h=1}^{n}\sum_{k=1}^{n} x_{ijhk} = 1 \qquad i,j = 1,\ldots,n \qquad (16)$$

$$\sum_{k=1}^{n} x_{ijhk} + \sum_{k=1,k\neq h}^{n} x_{ijkh} \leq y_h \qquad i,j,h = 1,\ldots,n \qquad (17)$$

$$x'_{ij} \geq 0 \qquad i,j = 1,\ldots,n \qquad (18)$$

$$x_{ijhk} \geq 0 \qquad i,j,h,k = 1,\ldots,n \qquad (19)$$

$$y_h \in \{0,1\} \qquad h = 1,\ldots,n \qquad (20)$$

Here, the objective function (15) takes into account the costs for direct services as well. Due to (16), all quantities must be shipped where some of them may be shipped directly.

And again, by adding the constraint (14) and assuming the special case $f_h = 0$ for all h, we would get a model for the p–hub median problem.

The seminal work of O'Kelly (1986,1987) has launched a series of follow–up publications. To get an overview of the work that has been done, we refer to Bryan and O'Kelly (1999), Campbell (1994a), Campbell et al. (2002), Cánovas et al. (2004), Klincewicz (1998), Mayer (2001), O'Kelly and Miller (1994) and Wagner (2005). Note that the models presented here are not intended to define the most efficient model formulations for the problems described. The models should just describe the problems formally and should show what almost — see section 2 — all authors assume. Publications where efficient model formulations are discussed are, e.g., Cánovas et al. (2005), Ernst et al. (2002), Ernst and Krishnamoorthy (1999), Kara and Tansel (2000), Skorin–Kapov et al. (1996), and Wagner (2003) just to mention a few.

In what follows, we will concentrate on the modeling of the transportation costs in the presence of economies of scale. In section 2 we will review the (surprisingly few) work that has been devoted to economies of scale in the context of hub & spoke network design. Eventually, in section 3 we will present and discuss our point of view on this matter. Section 4 will illustrate the new models from section 3 by means of a few numerical examples. A short conclusion will summarize the work in the final section.

2 Existing Approaches to Represent Economies of Scale

Surprisingly enough, although the economies of scale phenomenon is one of the main motivations for installing hub & spoke systems, the way costs are modeled has not really been questioned by many authors in this area.

At least some authors have noted that applying a discount factor α to the costs on arcs between hubs while disregarding the flow on these arcs contradicts the motivation of this discount factor. Consequently, Podnar et al. (2002) introduced flow tresholds that must be reached in order to gain the discount α. Campbell et al. (2004a, 2004b) state that "...the basic assumption in hub median models that flow costs are discounted on hub arcs to reflect high volumes leads to a possible mismatch between the abstracted model and the underlying motivations of the model". So, they consider models where so–called hub arcs, i.e. arcs that link two hubs and on which costs being discounted by a factor α, are to be selected explicitly which means that the cost for a flow between two hubs may or may not be discounted by a factor α depending on the selection. The total number of such hub arcs to be chosen is prespecified in their models, because otherwise every arc that links two hubs would be selected as a hub arc.

Three notable exceptions that address economies of scale are Bryan (1998), O'Kelly and Bryan (1998) and Klincewicz (2002). They argue — and we agree — that "By simplifying interhub travel costs and assuming that these costs are independent of flows, the current model not only miscalculates the total network cost, but also erroneously selects optimal hub locations and allocations".

As a consequence, O'Kelly and Bryan (1998) suggest to replace the interhub cost expression

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha c_{hk} q_{ij} x_{ijhk}$$

for hubs h and k in the objective function(s) by a concave function

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \left(1 - \theta \left(\frac{\sum_{i'=1}^{n}\sum_{j'=1}^{n} q_{i'j'} x_{i'j'hk}}{\sum_{i'=1}^{n}\sum_{j'=1}^{n} q_{i'j'}} \right)^{\beta} \right) c_{hk} q_{ij} x_{ijhk}$$

where the parameters $1 \geq \theta > 0$ and $\beta > 0$ are to be specified in advance. This function is monotonically increasing with the flow across the link between node h and node k. It can reasonably be used to model economies of scale: the marginal cost per unit is decreasing and the average cost per unit is decreasing as well. The latter property is what is called economies of scale.

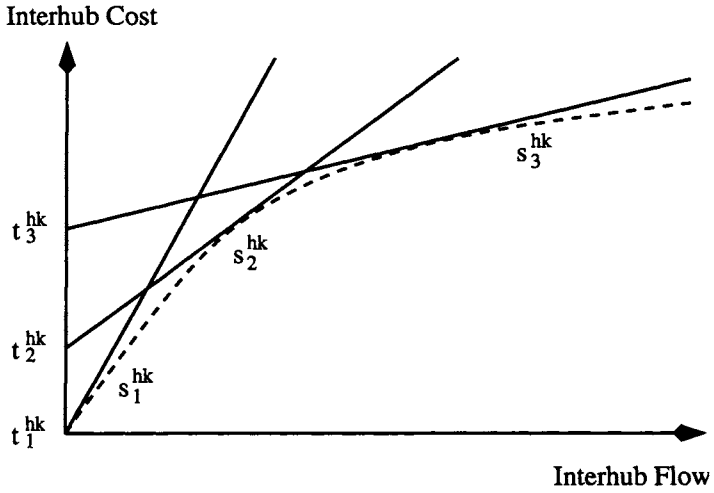


Figure 2: Interhub Cost Structure due to O'Kelly and Bryan (1998)

As illustrated in figure 2, the non–linear function

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left(1-\theta\left(\frac{\sum_{i'=1}^{n}\sum_{j'=1}^{n}q_{i'j'}x_{i'j'hk}}{\sum_{i'=1}^{n}\sum_{j'=1}^{n}q_{i'j'}}\right)^{\beta}\right)q_{ij}x_{ijhk}$$

can be approximated by a piecewise linear function in such a way that the lower envelope of this piecewise linear function approximates the function from above. This is exactly what Bryan (1998), O'Kelly and Bryan (1998) and Klincewicz (2002) do. Each piece r of these o linear pieces can be specified by two parameters: the intercept $t_r^{hk} \geq 0$ and the slope $s_r^{hk} > 0$. This allows to provide a linear mixed–integer formulation where the real–valued decision variable f_{rhk} is the total flow between hubs h and k to which the linear piece r is applied and the binary variable z_{rhk} indicates whether or not the piece r is applied at all. For the sake of brevity, we will provide here a model for the multiple allocation p–hub median problem without direct services. It should be easy for the reader to write down

other problem variants in a similar fashion.

$$\text{Min } F_4(\mathbf{f},\mathbf{x},\mathbf{y},\mathbf{z}) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{h=1}^{n}\sum_{k=1}^{n}(c_{ih}+c_{kj})q_{ij}x_{ijhk} + \sum_{r=1}^{o}\sum_{h=1}^{n}\sum_{k=1}^{n}c_{hk}(t_r^{hk}z_{rhk}+s_r^{hk}f_{rhk}) \tag{21}$$

s.t.

$$\sum_{r=1}^{o} f_{rhk} = \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}x_{ijhk} \qquad h,k=1,\ldots,n \tag{22}$$

$$f_{rhk} \le \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}z_{rhk} \qquad \begin{matrix} r=1,\ldots,o \\ h,k=1,\ldots,n \end{matrix} \tag{23}$$

$$\sum_{h=1}^{n}\sum_{k=1}^{n} x_{ijhk} = 1 \qquad i,j=1,\ldots,n \tag{24}$$

$$\sum_{k=1}^{n} x_{ijhk} + \sum_{k=1,k\neq h}^{n} x_{ijkh} \le y_h \qquad i,j,h=1,\ldots,n \tag{25}$$

$$\sum_{h=1}^{n} y_h = p \tag{26}$$

$$f_{rhk} \ge 0 \qquad \begin{matrix} r=1,\ldots,o \\ h,k=1,\ldots,n \end{matrix} \tag{27}$$

$$x_{ijhk} \ge 0 \qquad i,j,h,k=1,\ldots,n \tag{28}$$

$$y_h \in \{0,1\} \qquad h=1,\ldots,n \tag{29}$$

$$z_{rhk} \in \{0,1\} \qquad \begin{matrix} r=1,\ldots,o; \\ h,k=1,\ldots,n \end{matrix} \tag{30}$$

The objective function (21) measures interhub traffic by means of the cost approximation described above. The total flow between two hubs is represented by f as well as by x. (22) links these two decision variables in the correct manner. (23) guarantees that the flag to indicate whether or not a certain piece of the piecewise linear approximation is used is properly set. Note that due to the minimization objective, the lower envelope of the piecewise linear approximation is indeed used.

Bryan (1998) extends this model for economies of scale on all connections, not only the interhub ones. The technique with which this is done is the same piecewise linear approximation that is described above.

3 Alternative Model Formulations

From a cost accounting point of view, the model for economies of scale described in the previous section is hard to motivate. If economies of scale are due to quantity discounts, then we face a piecewise linear, monotone, quasiconcave cost function. We refer to Stadtler (2004) for a discussion of modeling quantity discounts. In a hub & spoke network design setting, however, economies of scale due to quantity discounts may appear only if the transportation is done by a third party. We will not discuss this case here.

Consequently we will try to derive an alternative model and reveal other sources for economies of scale. For doing this, let us consider two nodes i and h. These nodes can be hubs or not. And indeed we will see that economies of scale occur not only on interhub links but on all links. In general, if we provide a service from node i to node h, i.e. we have a positive flow from i to h, we face a (flow independent) fixed cost $c^f_{ih} \in \boldsymbol{R}_{\geq 0}$ and a unit (handling) cost $c^v_{ih} \in \boldsymbol{R}_{\geq 0}$. If f_{ih} is the flow between i and h then

$$\begin{array}{ll} c^f_{ih} + c^v_{ih} f_{ih}, & \text{if } f_{ih} > 0 \\ 0, & \text{otherwise} \end{array}$$

is the total cost for the service. Imagine, for example, the airline situation for passenger transport. If there is a flight from airport i to airport h, a (large) fixed cost c^f_{ih} for using the plane, employing the cabin crew etc. is incurred. In addition to that, a (relatively small) unit handling cost c^v_{ih} per passenger is incurred (mainly for serving an additional meal and a few drinks on board). Note that this simple cost model describes economies of scale already: The unit total cost is

$$\begin{array}{ll} \frac{c^f_{ih}}{f_{ih}} + c^v_{ih} & \text{if } f_{ih} > 0 \\ 0, & \text{otherwise} \end{array}$$

which means that the unit total cost decreases if the flow increases. Note that such a model, a model with fixed costs on active links, has already been provided by Campbell (1994b) who did not mention the economies of scale aspect. Another work with fixed costs on active arcs is the one by Garfinkel et al. (1996) which is confined to a very special situation (no handling costs, two hubs), but again, economies of scale have not been discussed. Gavish (1992) considers fixed costs on links for a computer network design problem.

It is important to note that flow from a node i to a node j may be routed through several — possibly more than two — other hub nodes to save fixed costs. We will just provide a model formulation for the multiple allocation p–hub median problem with direct services where the decision variable f_{ij} denotes the direct flow between nodes i and j and the binary decision variable z_{ij} indicates whether or not there is a positive flow between nodes i and j. The variable x_{ijhk} denotes the fraction of flow that originates from i and is sent to node j using the link between h and k:

$$\text{Min } F_5(\mathbf{f},\mathbf{x},\mathbf{y},\mathbf{z}) = \sum_{i=1}^{n}\sum_{j=1}^{n}(c_{ij}^f z_{ij} + c_{ij}^v f_{ij}) \tag{31}$$

s.t.

$$\sum_{k=1}^{n} x_{ijik} = 1 \qquad i,j = 1,\ldots,n \tag{32}$$

$$\sum_{k=1,k\neq h}^{n} x_{ijkh} = \sum_{k=1,k\neq h}^{n} x_{ijhk} \qquad \begin{array}{l} i,j,h = 1,\ldots,n \\ h \neq i,j \end{array} \tag{33}$$

$$\sum_{h=1}^{n} x_{ijhi} = 0 \qquad \begin{array}{l} i,j = 1,\ldots,n \\ i \neq j \end{array} \tag{34}$$

$$\sum_{k=1}^{n} x_{ijhk} \leq y_h \qquad \begin{array}{l} i,j,h = 1,\ldots,n \\ h \neq i,j \end{array} \tag{35}$$

$$\sum_{k=1}^{n} x_{ijkh} \leq y_h \qquad \begin{array}{l} i,j,h = 1,\ldots,n \\ h \neq i,j \end{array} \tag{36}$$

$$\sum_{h=1}^{n} y_h = p \tag{37}$$

$$f_{ij} = \sum_{h=1}^{n}\sum_{k=1}^{n} q_{hk} x_{hkij} \qquad i,j = 1,\ldots,n \tag{38}$$

$$f_{ij} \leq \sum_{h=1}^{n}\sum_{k=1}^{n} q_{hk} z_{ij} \qquad i,j = 1,\ldots,n \tag{39}$$

$$f_{ij} \geq 0 \qquad i,j = 1,\ldots,n \tag{40}$$

$$x_{ijhk} \geq 0 \qquad i,j,h,k = 1,\ldots,n \tag{41}$$

$$y_h \in \{0,1\} \qquad h = 1,\ldots,n \tag{42}$$

$$z_{ij} \in \{0,1\} \qquad i,j = 1,\ldots,n \tag{43}$$

The objective function (31) is defined as described above. (32) and (33) are flow constraints. For each pair of nodes i and j the complete quantity must

leave node i and reach node j. Each flow that reaches a node in–between must leave that node completely. (34) prevents short cycles. (35) and (36) make sure that only hubs nodes can be used in–between to go from a node i to a node j. (38) calculates the total flow over a link from node i to node j. Of course, because of this equality constraint, the variable f could be completely eliminated from the model by substitution. (39) makes sure that a positive flow on a link between i and j is correctly indicated. It is noteworthy to mention that there always exists an optimum solution with all x–variables being integer valued.

Note that if we replace $\sum_h \sum_k q_{hk}$ by some number L_{ij} in (39), we could easily model a constrained capacity on that link. Then, an optimum solution with integral x–values may not exist.

Should direct services not be allowed, the constraint

$$x_{ijij} \leq y_i + y_j \qquad i,j = 1,\ldots,n$$

would forbid such service.

The fixed cost c^f_{ij} is charged if the directed link between node i and node j is active. Using that link in opposite direction incurs an additional fixed cost c^f_{ji}. If we consider two nodes i and j where $i \leq j$ and if we would like to model a situation where a fixed cost c^f_{ij} is charged no matter in what direction the link between these two nodes is used, then we could simply add the constraint

$$z_{ij} = z_{ji} \qquad \begin{array}{l} i = 1,\ldots,n-1 \\ j = i+1,\ldots,n \end{array}$$

and set $c^f_{ji} = 0$ for all $i < j$. Of course, we could eliminate almost half of the z–variables instead to keep only those variables z_{ij} with $i \leq j$ which requires a slight modification of the model formulation especially in (31) and (39).

There may be situations where the just presented model is still too simple. Imagine, for instance, a trucking company. If they provide a transportation service from town i to town h, they may have a fixed cost c^f_{ij}, they may have a unit handling cost c^v_{ij}, but they may also figure out that the number of trucks to serve that link depends on the flow, because each truck has a limited capacity Q, where each truck incurs a fixed cost $c^t_{ij} \in \boldsymbol{R}_{\geq 0}$ as well. If f_{ih} is the flow on that connection, the total cost adds to

$$\begin{array}{ll} c^f_{ih} + c^t_{ih} \left\lceil \dfrac{f_{ih}}{Q} \right\rceil + c^v_{ih} f_{ih}, & \text{if } f_{ih} > 0 \\ 0, & \text{otherwise} \end{array}$$

and figure 3 illustrates the situation. Similar situations have been mentioned by Ebner (1997) and Wlček (1998).

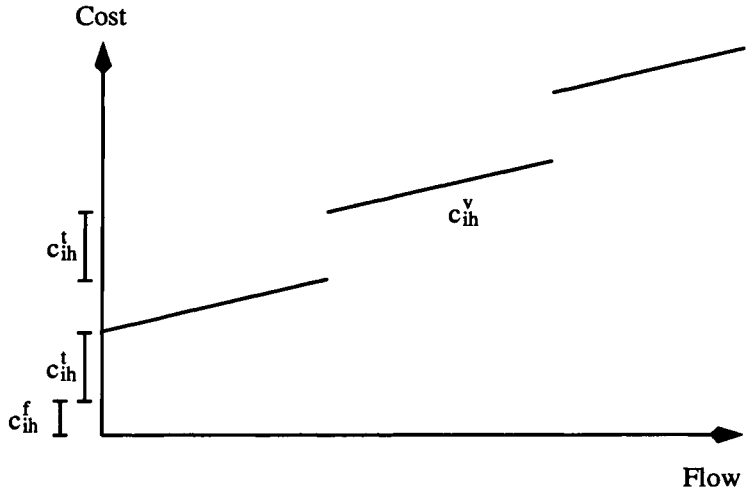


Figure 3: Cost Structure with Fixed Costs

The corresponding model formulation with an integer–valued decision variable t_{ih} for the number of vehicles needed looks as follows:

$$\text{Min } F_6(\mathbf{f},\mathbf{t},\mathbf{x},\mathbf{y},\mathbf{z}) = \sum_{i=1}^{n}\sum_{j=1}^{n}(c^f_{ij}z_{ij} + c^t_{ij}t_{ij} + c^v_{ij}f_{ij}) \tag{44}$$

s.t.

$$\sum_{k=1}^{n} x_{ijik} = 1 \qquad i,j = 1,\ldots,n \tag{45}$$

$$\sum_{k=1,k\neq h}^{n} x_{ijkh} = \sum_{k=1,k\neq h}^{n} x_{ijhk} \qquad \begin{array}{l} i,j,h = 1,\ldots,n \\ h \neq i,j \end{array} \tag{46}$$

$$\sum_{h=1}^{n} x_{ijhi} = 0 \qquad \begin{array}{l} i,j = 1,\ldots,n \\ i \neq j \end{array} \tag{47}$$

$$\sum_{k=1}^{n} x_{ijhk} \leq y_h \qquad \begin{array}{l} i,j,h = 1,\ldots,n \\ h \neq i,j \end{array} \tag{48}$$

$$\sum_{k=1}^{n} x_{ijkh} \leq y_h \qquad \begin{array}{l} i,j,h = 1,\ldots,n \\ h \neq i,j \end{array} \tag{49}$$

$$\sum_{h=1}^{n} y_h = p \tag{50}$$

$$f_{ij} = \sum_{h=1}^{n}\sum_{k=1}^{n} q_{hk} x_{hkij} \qquad i,j = 1,\ldots,n \tag{51}$$

$$f_{ij} \leq \sum_{h=1}^{n}\sum_{k=1}^{n} q_{hk} z_{ij} \qquad i,j = 1,\ldots,n \tag{52}$$

$$f_{ij} \leq Q \cdot t_{ij} \qquad i,j = 1,\ldots,n \tag{53}$$

$$f_{ij} \geq 0 \qquad i,j = 1,\ldots,n \tag{54}$$

$$t_{ij} \in N_0 \qquad i,j = 1,\ldots,n \tag{55}$$

$$x_{ijhk} \geq 0 \qquad i,j,h,k = 1,\ldots,n \tag{56}$$

$$y_h \in \{0,1\} \qquad h = 1,\ldots,n \tag{57}$$

$$z_{ij} \in \{0,1\} \qquad i,j = 1,\ldots,n \tag{58}$$

The objective function (44) has already been defined above. The main point that is new here is restriction (53). Because of this constraint, the capacity offered on a certain link must be sufficient to transport the calculated flow on that link. Because of the minimization objective, the offered capacity will be as small as possible. It should be remarked that an optimum solution with all x–variables being integral may not exist.

Additional constraints of the form

$$t_{ij} \leq T_{ij} \qquad i,j = 1,\ldots,n$$

could be used to model a restricted number of T_{ij} vehicles on a particular link. In a similar fashion

$$\sum_{i=1}^{n}\sum_{j=1}^{n} t_{ij} \leq T$$

could constrain the total number of vehicles to a limit T in the whole network.

This model can be adapted to even more general situations. Imagine situations where the type of transportation vehicle can also be decided. For instance, if one can use large trucks instead of small ones, trains instead of trucks, planes instead of trucks or trains and so on. Or imagine the telecommunication industry where the bandwidth of connected nodes can be decided upon. Let us assume that each link has a set of M so–called modes in which that link can be established. For a particular mode m, let Q^m be the capacity of a vehicle that corresponds to that mode and let c_{ih}^{fm},

c_{ih}^{tm} and c_{ih}^{vm} be the cost coefficients for this mode for a given link (i,h). Note that several modes may cause economies of scale if $c_{ih}^{tm}/Q^m > c_{ih}^{tm'}/Q^{m'}$ for modes m and m'. But this is not a sufficient criterion and depends especially on the values of c_{ih}^{fm} and c_{ih}^{vm}. The adapted decision variables z_{ih}^m, t_{ih}^m and f_{ih}^m have a straightforward interpretation:

$$\text{Min } F_7(\mathbf{f},\mathbf{t},\mathbf{x},\mathbf{y},\mathbf{z}) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(c_{ij}^{f} z_{ij} + \left(\sum_{m=1}^{M}\left(c_{ij}^{fm} z_{ij}^m + c_{ij}^{tm} t_{ij}^m + c_{ij}^{vm} f_{ij}^m\right)\right)\right) \tag{59}$$

s.t.

$$\sum_{k=1}^{n} x_{ijik} = 1 \qquad i,j = 1,\ldots,n \tag{60}$$

$$\sum_{k=1,k\neq h}^{n} x_{ijkh} = \sum_{k=1,k\neq h}^{n} x_{ijhk} \qquad \begin{array}{l} i,j,h = 1,\ldots,n \\ h \neq i,j \end{array} \tag{61}$$

$$\sum_{h=1}^{n} x_{ijhi} = 0 \qquad \begin{array}{l} i,j = 1,\ldots,n \\ i \neq j \end{array} \tag{62}$$

$$\sum_{k=1}^{n} x_{ijhk} \leq y_h \qquad \begin{array}{l} i,j,h = 1,\ldots,n \\ h \neq i,j \end{array} \tag{63}$$

$$\sum_{k=1}^{n} x_{ijkh} \leq y_h \qquad \begin{array}{l} i,j,h = 1,\ldots,n \\ h \neq i,j \end{array} \tag{64}$$

$$\sum_{h=1}^{n} y_h = p \tag{65}$$

$$\sum_{m=1}^{M} f_{ij}^m = \sum_{h=1}^{n}\sum_{k=1}^{n} q_{hk} x_{hkij} \qquad i,j = 1,\ldots,n \tag{66}$$

$$\sum_{m=1}^{M} f_{ij}^m \leq \sum_{h=1}^{n}\sum_{k=1}^{n} q_{hk} z_{ij} \qquad i,j = 1,\ldots,n \tag{67}$$

$$f_{ij}^m \leq \sum_{h=1}^{n}\sum_{k=1}^{n} q_{hk} z_{ij}^m \qquad \begin{array}{l} m = 1,\ldots,M \\ i,j = 1,\ldots,n \end{array} \tag{68}$$

$$f_{ij}^m \leq Q^m \cdot t_{ij}^m \qquad \begin{array}{l} m = 1,\ldots,M \\ i,j = 1,\ldots,n \end{array} \tag{69}$$

$$f_{ij}^m \geq 0 \qquad \begin{array}{l} m = 1,\ldots,M \\ i,j = 1,\ldots,n \end{array} \tag{70}$$

$$t_{ij}^m \in N_0 \qquad m = 1, \ldots, M;\ i, j = 1, \ldots, n \tag{71}$$

$$x_{ijhk} \geq 0 \qquad i, j, h, k = 1, \ldots, n \tag{72}$$

$$y_h \in \{0, 1\} \qquad h = 1, \ldots, n \tag{73}$$

$$z_{ij} \in \{0, 1\} \qquad i, j = 1, \ldots, n \tag{74}$$

$$z_{ij}^m \in \{0, 1\} \qquad m = 1, \ldots, M;\ i, j = 1, \ldots, n \tag{75}$$

Again, the objective function (59) has already been explained above. The interesting aspect here is (66). One should note that due to this formulation, the mode for a link between i and h need not be unique. Indeed, one could use a mix of different modes. If a certain mode m should be not available on a link from node i to node j, then one can simply set $z_{ij}^m = 0$ so that due to (68) this mode will not be active on that link. Eventually, one should note that an optimum solution with all x–variables being integer–valued may not exist.

An extended model where at most one mode per link can be chosen is easy to formulate by adding:

$$\sum_{m=1}^{M} z_{ij}^m \leq 1 \qquad i, j = 1, \ldots, n$$

4 Numerical Examples

To illustrate the above models, we provide here the optimum results for three random examples computed with the commercial mathematical programming software package AMPL/CPLEX. Every example consists of $n = 7$ nodes. The number of hubs is always $p = 3$. The quantities q_{ij} to be transported in all examples are given by table 1, and the fixed cost coefficients c_{ij}^f are provided in table 2.

Example 1 corresponds to the model formulation (31)–(43). In addition to the parameters already introduced, the variable cost coefficients c_{ij}^v as defined in table 3 are used.

Figure 4 shows an optimum solution for example 1 (what is shown are the positive flows f_{ij}). The optimum objective function value is 2403. It is remarkable to note that in this solution some quantities must indeed be transported via more than two hubs. The five units to be shipped from node $i = 5$ to node $j = 2$, for example, flow through hubs 1, 3, and 4 in that order.

Table 1: Transportation Quantities q_{ij} for the Examples

q_{ij}	1	2	3	4	5	6	7
1	8	3	7	7	4	3	4
2	4	1	5	5	6	5	7
3	5	4	2	4	9	4	8
4	8	8	1	8	8	7	5
5	5	5	4	2	5	8	1
6	8	2	8	1	6	1	2
7	7	3	9	3	2	2	3

Table 2: Fixed Cost Coefficients c^f_{ij} for the Examples

c^f_{ij}	1	2	3	4	5	6	7
1	60	176	50	175	130	134	70
2	80	184	40	120	190	60	64
3	90	132	46	149	168	57	59
4	200	154	47	138	147	180	80
5	150	156	48	168	123	147	190
6	140	189	168	147	150	153	130
7	160	147	137	159	130	164	180

Table 3: Variable Cost Coefficients c^v_{ij} for the Examples

c^v_{ij}	1	2	3	4	5	6	7
1	7	2	3	8	1	2	4
2	2	3	3	6	5	8	2
3	7	4	5	1	3	9	9
4	2	1	3	5	7	4	4
5	2	5	9	5	8	4	2
6	7	5	5	3	1	1	2
7	3	3	1	5	9	8	7

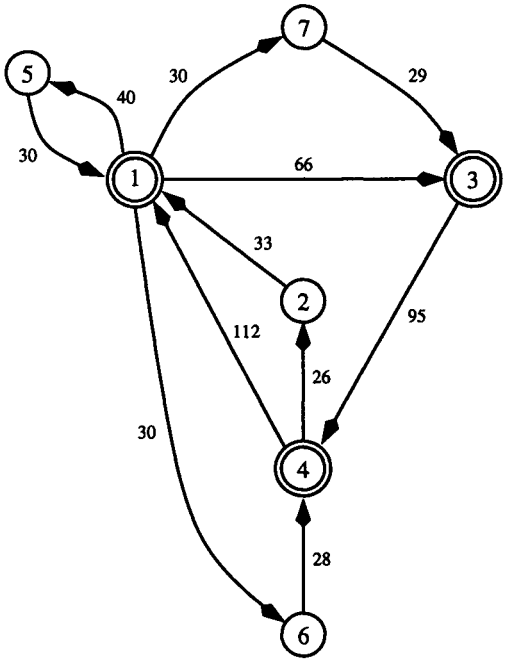


Figure 4: An Optimum Solution for Example 1

Example 2 corresponds to the model formulation (44)–(58). We use $Q = 10$. The cost coefficients c_{ij}^f and c_{ij}^v are the same as in example 1. The cost coefficients c_{ij}^t can be found in table 4.

Table 4: Cost Coefficients c_{ij}^t for the Examples

c_{ij}^t	1	2	3	4	5	6	7
1	45	49	35	18	36	14	90
2	42	62	58	46	35	58	75
3	38	15	42	58	64	75	31
4	10	20	30	57	18	45	46
5	75	15	46	35	26	14	70
6	50	60	10	80	66	55	40
7	80	60	20	25	16	34	71

Figure 5 shows an optimum solution for example 2 (again, the positive flows f_{ij} are shown). The optimum objective function value is 4327. Note that just by introducing a fixed cost c_{ij}^t per vehicle to be used on a link, the solution looks quite different than the one for example 1.

Example 3 corresponds to the model formulation (59)–(75). $M = 2$ modes are used with $Q^1 = 10$ and $Q^2 = 20$. The fixed cost coefficients c_{ij}^f are defined in table 2. The mode dependent fixed cost coefficients c_{ij}^{fm} are provided in table 5. For mode $m = 1$ we use the cost coefficients c_{ij}^{v1} and c_{ij}^{t1} like they were defined in tables 3 and 4, respectively. For mode $m = 2$ the values of the parameters c_{ij}^{v2} and c_{ij}^{t2} are specified in tables 6 and 7, respectively.

Figure 6 shows an optimum solution for example 3 (here, the flows f_{ij}^1/f_{ij}^2 are shown). The optimum objective function value is 3736. Note again that just by introducing new aspects (a second mode) and keeping all other parameters as they were before in example 2, the solution looks completely different. It should be noted in example 3 that one link indeed uses a mix of modes (see the link from node 2 to node 1). Some nodes (see, e.g., node 6) have different ingoing and outgoing modes. In practice this could mean, for instance, that small trucks are used to go to node 6 and large trucks are required to come from node 6. As a consequence one would face a number of empty moves of the vehicles. Additional constraints and terms in the objective function that take into account such empty moves may become relevant in some applications. This example also reveals the

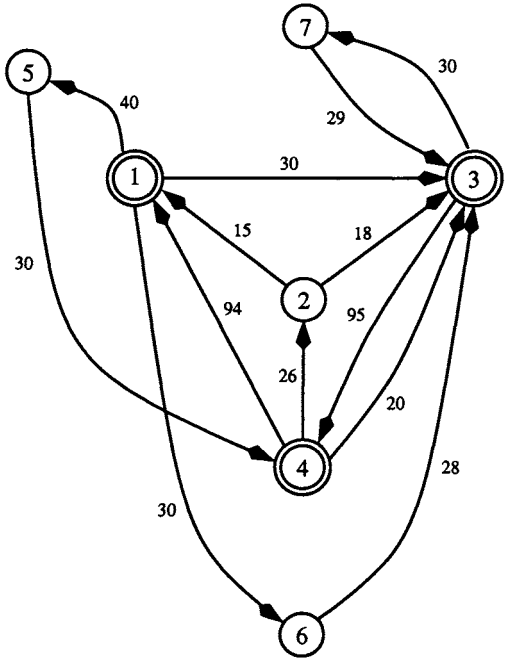


Figure 5: An Optimum Solution for Example 2

Table 5: Mode Dependent Fixed Cost Coefficients c_{ij}^{f1}/c_{ij}^{f2} for the Examples

c_{ij}^{f1}/c_{ij}^{f2}	1	2	3	4	5	6	7
1	3/9	5/5	4/7	8/8	3/4	1/6	5/5
2	6/6	4/2	7/3	8/3	9/4	6/8	4/7
3	6/3	8/1	5/4	5/5	3/6	6/8	4/6
4	3/3	5/2	7/4	9/7	3/9	5/9	4/5
5	3/1	5/4	7/7	8/8	9/5	5/4	2/3
6	4/5	8/7	9/8	3/4	4/2	4/1	7/3
7	8/2	5/4	4/5	6/7	2/8	1/9	3/3

Table 6: Variable Cost Coefficients c_{ij}^{v2} for the Examples

c_{ij}^{v2}	1	2	3	4	5	6	7
1	8	7	5	6	1	3	2
2	2	3	2	2	7	9	6
3	5	3	5	1	3	1	7
4	2	6	3	1	5	6	4
5	5	6	3	2	1	4	7
6	5	6	3	2	4	5	7
7	1	4	5	6	3	2	1

Table 7: Cost Coefficients c_{ij}^{t2} for the Examples

c_{ij}^{t2}	1	2	3	4	5	6	7
1	90	60	50	30	40	20	100
2	50	70	75	60	55	69	140
3	70	22	60	90	100	133	50
4	18	33	53	67	19	80	70
5	80	17	66	55	50	27	110
6	99	69	15	140	77	99	50
7	90	100	35	30	30	60	80

phenomenon of split quantities. It is a little hard to see in figure 6, but if we would examine in more detail which quantities q_{ij} go what what ways — which can easily be done by inspecting the values of the decision variables x — it would turn out in this example that the five units to be shipped from node $i = 2$ to node $j = 3$ are split: Four units go directly from node 2 to node 3, but one unit goes from node 2 to node 1 and from there it goes to node 3.

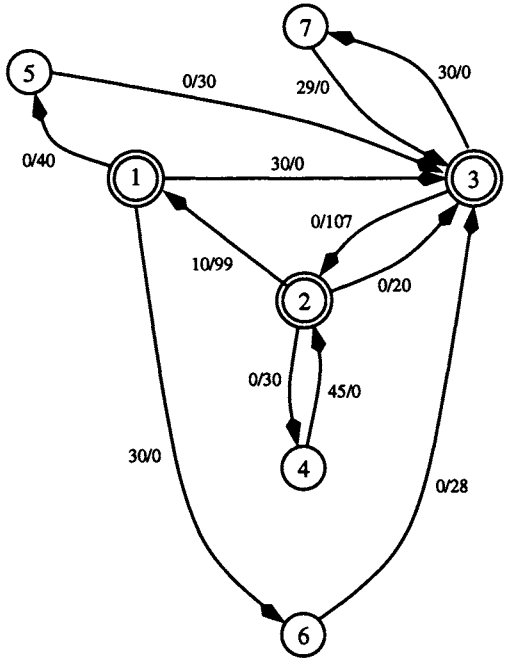


Figure 6: An Optimum Solution for Example 3

5 Conclusion

This paper is devoted to the modeling of economies of scale within hub & spoke network design problems. First we have shown how economies of scale are modeled in traditional hub & spoke network design models and it turned out that all flows have a constant unit cost. Interhub connections allow to reduce the unit cost of flow by a factor α. Hence, all traditional models do not represent flow dependent unit costs and therefore do not reflect the issue of economies of scale in a correct manner. This is rather

surprising, because economies of scale are a key driving force for installing hub & spoke systems. Given that (almost) all researchers up to today have used this (incorrect) way of modeling economies of scale, it is time to question this, because obviously wrong models lead to wrong decisions no matter how good the procedures to solve these models are.

It seems that only very few authors, namely O'Kelly and Bryan (1998) and Klincewicz (2002), have noticed this serious defect before. To be fair, Podnar et al. (2002) and Campbell et al. (2004a, 2004b) also question traditional models and discuss approaches where the discount factor α is applied only under certain, flow dependent circumstances, but they still rely on that factor α which is not a convincing approach for modeling economies of scale. O'Kelly and Bryan (1998) and Klincewicz (2002) have suggested a non–linear, concave cost function to model economies of scale between interhub connections in a much more appropriate way. Using a piecewise linear approximation, these authors provide a linear mixed–integer model formulation (and solution procedures not discussed here) where the unit cost is flow dependent. However, the motivation for using this non–linear, concave cost function for modeling economies of scale is somewhat weak in our opinion, because there is no cost accounting argument that justifies such a cost function.

Given this, we contribute the following: First of all, we allow that economies of scale do not only occur on interhub connections only, but they can occur on all kinds of connections, a point that has been made by Bryan (1998) already before. So, we propose models to reflect this. Furthermore, we give an economic explanation for the occurrence of economies of scale which allows us to derive a cost function that is less artificial and can be explained better than the one proposed in the literature. The sources for economies of scale mentioned in this paper are (i) quantity discounts (if a third party is employed), (ii) fixed costs, and (iii) multiple modes. Models and examples are provided in this paper to illustrate the latter two aspects.

Future work should be dedicated to develop and test exact and heuristic solution procedures for these models. Also, an emphasis should be on investigating real–world applications to explain by cost accounting arguments which costs really matter and how these costs are calculated to get much more reliable solutions.

Acknowledgement

We thank James Campbell, Sergio García Quilles, John Klincewicz, Jadranka Skorin–Kapov, and Bernd Wagner for their comments on a draft version of this paper. Maria Merker has implemented the models from section 3 in AMPL.

References

[1] Bryan DL (1998) Extensions to the hub location problem: Formulations and numerical examples. Geographical Analysis 30:315–330

[2] Bryan DL, O'Kelly ME (1999) Hub–and–spoke networks in air transportation: An analytical review. Journal of Regional Science 39:275–295

[3] Campbell JF (1994a) A survey of network hub location. Studies in Locational Analysis 6:31–49

[4] Campbell JF (1994b) Integer programming formulations of discrete hub location problems. European Journal of Operational Research 72:387–405

[5] Campbell JF, Ernst AT, Krishnamoorthy M (2002) Hub location problems. In: Drezner Z, Hamacher HW (eds.) Facility Location – Applications and Theory. Springer, Berlin, pp. 373–407

[6] Campbell JF, Ernst AT, Krishnamoorthy M (2004a) Hub arc location problems: Part I – introduction and results. Management Science, to appear

[7] Campbell JF, Ernst AT, Krishnamoorthy M (2004b) Hub arc location problems: Part II – formulations and optimal algorithms. Management Science, to appear

[8] Cánovas L, García S, Marín A (2004) Localización de Concentradores. In: Pelegrín B (ed.) Avances en Localización de Servicios y sus Aplicaciones. Servicio de Publicaciones de la Universidad de Murcia, pp. 35–58

[9] Cánovas L, Landete M, Marín A (2005) New formulations for the uncapacitated multiple allocation hub location problem. European Journal of Operational Research, to appear

[10] Ebner G (1997) Controlling komplexer Logistiknetzwerke. GVB Schriftenreihe 34, Nürnberg

[11] Ernst AT, Hamacher H, Jiang H, Krishnamoorthy M, Woeginger G (2002) Uncapacitated single and multiple allocation p–hub center problems. Working Paper, CSIRO Mathematical and Information Sciences, Clayton

[12] Ernst AT, Krishnamoorthy M (1999) Solution algorithms for the capacitated single allocation hub location problem. Annals of Operations Research 86:141–159

[13] Garfinkel RS, Sundararaghavan PS, Noon C, Smith DR (1996) Optimal use of hub facilities: A two–hub model with fixed arc costs. Sociedad de Estadística e Investigación Operativa, Top 4:331–343

[14] Gavish B (1992) Topological design of computer communication networks – the overall design problem. European Journal of Operational Research 58:149–172

[15] Kara BY, Tansel BÇ (2000) On the single–assignment p–hub center problem. European Journal of Operational Research 125:648–655

[16] Klincewicz JG (1998) Hub location in backbone/tributary network design: A review. Location Science 6:307–335

[17] Klincewicz JG (2002) Enumeration and search procedures for a hub location problem with economies of scale. Annals of Operations Research 110:107–122

[18] Mayer G (2001) Strategische Logistikplanung von Hub&Spoke–Systemen. Deutscher Universitäts–Verlag, Wiesbaden

[19] O'Kelly ME (1986) The location of interacting hub facilities. Transportation Science 20:92–106

[20] O'Kelly ME (1987) A quadratic integer program for the location of interacting hub facilities. European Journal of Operational Research 32:393–404

[21] O'Kelly ME, Bryan DL (1998) Hub location with flow economies of scale. Transportation Research Part B 32:605–616

[22] O'Kelly ME, Miller HJ (1994) The hub network design problem – a review and synthesis. Journal of Transport Geography 2:31–40

[23] Podnar H, Skorin–Kapov J, Skorin–Kapov D (2002) Network cost minimization using threshold based discounting. European Journal of Operational Research 137:371–386

[24] Skorin–Kapov D, Skorin–Kapov J, O'Kelly M (1996) Tight linear programming relaxations of uncapacitated p–hub median problems. European Journal of Operational Research 94:582–593

[25] Stadtler H (2004) A general quantity discount and supplier selection mixed integer programming model. Working Paper, University of Hamburg

[26] Wagner B (2003) An exact solution procedure for a cluster hub location problem. Working Paper, TU Darmstadt

[27] Wagner B (2005) Design von Hub&Spoke-Netzwerken in der Logistik – modellbasierte Lösungsansätze. Ph.D. dissertation, TU Darmstadt, in preparation

[28] Wlček H (1998) Gestaltung der Güterverkehrsnetze von Sammelgutspeditionen. GVB Schriftenreihe 37, Nürnberg

A Stackelberg Equilibrium Model for Supply Chain Inventory Management*

Yeong-Cheng Liou

Department of Information Management,
Cheng Shiu University

Siegfried Schaible (corresponding author)

A. G. Anderson Graduate School of Management,
University of California

Jen-Chih Yao

Department of Applied Mathematics,
National Sun Yat-sen University

Abstract

In this paper we consider one-buyer, one-seller, finite horizon, multi-period inventory models in which the economic order quantity is integrated with the economic production quantity (EOQ-EPQ in short). We introduce the Stackelberg equilibrium framework in which the objective is to maximize the vendor's total benefit subject to the minimum total cost that the buyer is willing to incur. Some existence results, optimality conditions and the optimal replenishment policy under the Stackelberg equilibrium concept are obtained and a numerical algorithm is presented to find the optimal replenishment policy in practice.

Keywords. Inventory models, Stackelberg equilibrium, economic order quantity-economic production quantity.

1 Introduction

It is important to reduce the logistical cost including the transportation/ distribution cost and warehousing cost in a modern competitive enterprise given the cycle of market/product lifetime and the challenge of shortening lead-times. Supply chain management (SCM in short) is a key to solve the problems above. In the literature many scholars [6, 10-11] focus on

*This research was partially supported by a grant from the National Science Council of Taiwan R.O.C.

the joint total costs of both parties, the buyer's and the seller's. But that cannot provide the main idea for a strategy in SCM. The concept of joint total costs only reduces the cost of a single party, the buyer's or the seller's, instead of integrating both parties. Therefore we propose an SCM inventory model with market mechanisms to fit the goal of SCM, "benefit-sharing with contracts".

Stackelberg games play an important role in economics, design of mechanical structures, transportation, resource allocation, decomposition reformulations of large-scale mathematical programming, minimax mathematical programming and decision science. This problem is a special type of bilevel programming problems (BP in short) that have been introduced in the optimization area in the seventies (see [2, 4-5, 8]). Nash equilibria for static games make consistent predictions of the behavior of all players; that is, no player has an incentive to play differently. However sometimes one of the players has the ability to enforce his strategy on the other players. In 1934 von Stackelberg first proposed the solution concept of a Stackelberg equilibrium for dynamic games. In the framework of a Stackelberg game there is the leader-level and follower-level party. Each has his own decision rule. The leader declares his strategy first and enforces it on the follower. It is a hierarchical equilibrium solution concept. The framework can be described as follows

$$\text{(BP)} \quad \begin{array}{ll} \min & f(x,y) \\ \text{s.t.} & y \in Sol(x), x \in X \end{array}$$

where the decision variable on the leader-level is $x \in X$. The decision variable on the lower-level is $y \in Y$, $f : X \times Y \to \mathbb{R}$ is the leader-level objective function, $Sol : X \to 2^Y$ is a set-valued mapping such that for each $x \in X$, $Sol(x)$ denotes the solution/reaction set in the lower-level problem for a given x. It is characterized by

$$Sol(x) = \{y : g(x,y) \leq g(x,z) \ \forall \ z \in \Omega(x)\}$$

where $g : X \times Y \to \mathbb{R}$ is the lower-level objective function and $\Omega : X \to 2^Y$ is a constraint mapping. That is, the lower-level problem for a given x can be formulated as

$$\begin{array}{ll} \min & g(x,y) \\ \text{s.t.} & y \in \Omega(x). \end{array}$$

We observe that if the lower-level objective function is differentiable, then the lower-level problem can be characterized by a variational inequality problem, i.e.,

$$Sol(x) = \{y^* \in \Omega(x) : \langle \nabla_y g(x,y^*), y - y^* \rangle \geq 0 \text{ for all } y \in \Omega(x)\}. \quad (1)$$

In [8] Liou and Yao establish some existence results for (BP) in a reflexive Banach space setting which are based on $Sol(x)$ as given by (1) for each x.

2 The Model and Preliminaries

The notation and the assumptions used here to develop the mathematical model are as follows:

Notation	Definition
V	Ordering cost, \$/per unit
Ch_B	Holding cost for the buyer, \$/per unit/per order/per year
Cs_B	Shortage cost for the buyer, \$/per unit/per order/per year
H	Finite horizon in planning cycle
n	The number of distributions, #/per order
q	The amount of distribution, #/per order
D	The total demand during the finite horizon in a planning cycle
P	The total production during the finite horizon in a planning cycle
t_s	The stopping time of production per order
C	The contract transaction cost, \$/per unit
Ch_V	Holding cost for the vendor, \$/per unit/per order/per year
Cm_V	Production cost for the vendor, \$/per unit
Ct	Transportation cost of inventory, \$/per distribution/per order
Cp	Processing cost of inventory, \$/per order
α	The ratio of transportation fee shared by the buyer, $0 \leq \alpha \leq 1$
s_i	The ith reorder point, $0 \leq i \leq n$, with $s_0 = 0$
t_i	The ith replenishment/backlogged point, $1 \leq i \leq n$
u_n	The replenishment policy, that is, $u_n = \{(t_i, s_i)_{i=1}^n, q\}$

For simplicity, the assumptions used here to develop the mathematical model are as follows:

(i) The assumptions of EOQ/EPQ with a single product, one-vendor, one-buyer, full backordering and $P > D$, will be satisfied and multi-order multi-distribution in a finite horizon H.

(ii) The amount of distribution per order, q is a constant per distribution and the number of distributions per order n is an integer constant with $(D \bmod nq) = 0$. The stopping time of production of the vendor is corresponding to some backlogging point of the buyer.

(iii) The backlogging has a delay in time when an order is placed, that is, there exists lead-times in each ordering.

(iv) Six types of cost functions are considered, including the ordering cost, holding cost, shortage cost, processing cost, transportation cost and contract transaction cost. They are continuous in time. Moreover, the transportation cost will be shared via the ratio α and the motivation of contract is that the holding cost are less than the shortage cost.

Figure 1 shows the behavior of the continuous-time inventory system $I_B(t)$ without contracts at time t.

(1) The buyer's model

Let $\{s_k\}_{k=0}^{n}$ be a strictly increasing sequence with $\lim_{k\to\infty} s_k = \frac{Hnq}{D}$ and $s_0 = 0$. Let the sequence $\{t_k\}_{k=1}^{n}$ be strictly increasing with $I(t_k+) - I(t_k) = q > 0$ for all k. We make an assumption that the number of procurements $1 \le \frac{D}{nq}$ during the finite horizon $[0, H]$ is an integer value. Then $u_n = ((s_k, t_k)_{k=1}^{n}, q) \in U(n)$ is called an admissible strategy where $U(n) = \left\{0 \le t_1 \le s_1 \le \cdots \le t_n \le s_n = \frac{Hnq}{D}, q > 0\right\}$ is the set of admissible strategies for a given n. The inventory trajectory with $s_0 = 0$ based on Figure 1 is described by

$$\frac{\mathrm{d}I_B(t)}{\mathrm{d}t} = -\frac{D}{H} \text{ with } I(s_{i-1}+) = I(s_i+) = 0,\ s_{i-1} \le t \le s_i\ (i = 1, \cdots, n),$$

where $I_B(s_i+) = \lim_{h\downarrow 0} I_B(t+h)$, that is $I_B(\cdot)$ is right-continuous. Then we have the inventory level at time t

$$I_B(t) = -\frac{D}{H}t + \sum_{k=1}^{N(t)} q \text{ where } N(t) = \sup\{k : t_k \le t\}.$$

Hence we can obtain the related costs with respect to the total cost as follows:

(i) Holding cost $\mathcal{H}_i$: the amount of inventory carried during the part of the ith cycle, $t_i \le t \le s_i$, multiplied by the unit holding cost in each procurement, given by

$$\mathcal{H}_i = Ch_B \int_{t_i}^{s_i} I_B(t)\ \mathrm{d}t = \frac{Ch_B D}{2H}(s_i - t_i)^2.$$

(ii) Shortage cost $\mathcal{B}_i$: the amount of lost sales during the part of the ith cycle, $s_{i-1} \le t \le t_i$, multiplied by the unit backorder cost in each procurement, given by

$$\mathcal{B}_i = Cs_B \int_{s_{i-1}}^{t_i} |I_B(t)|\ \mathrm{d}t = \frac{Cs_B D}{2H}(t_i - s_{i-1})^2.$$

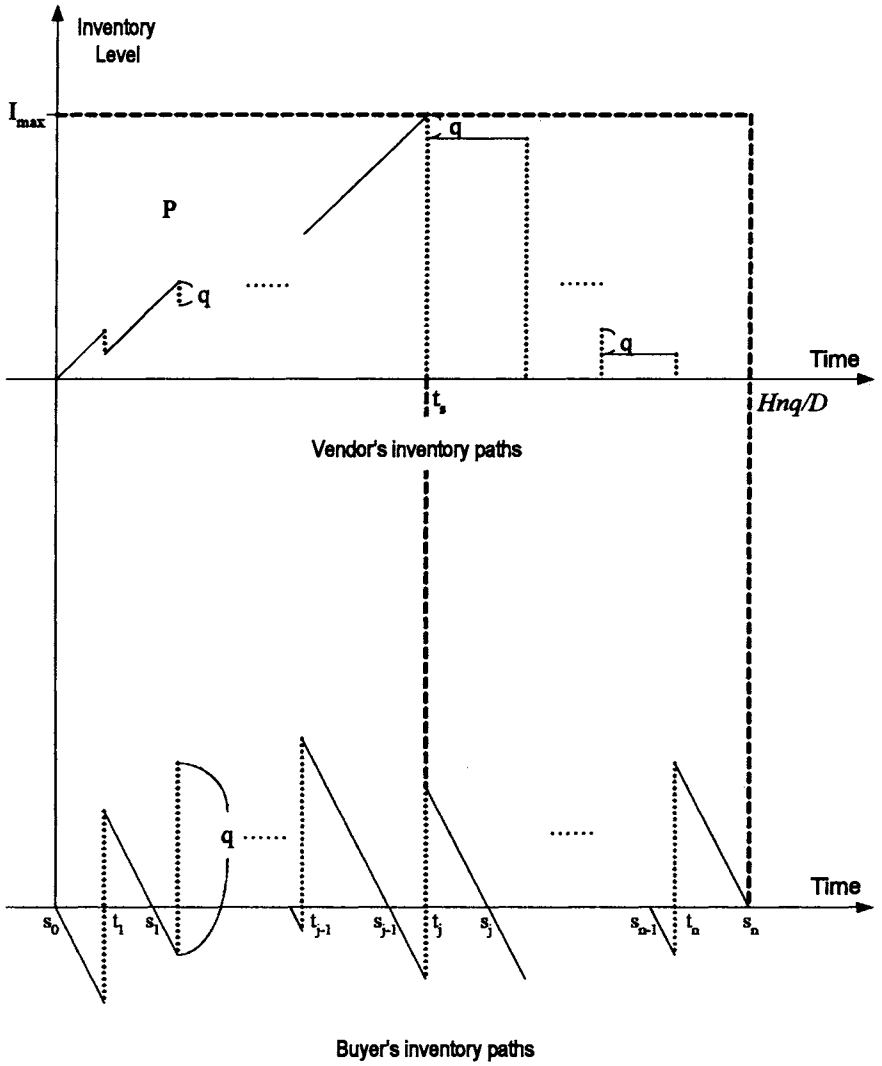


Figure 1: The trajectory of inventory without contracts

The total cost of the buyer with a continuous policy u_n is given by

$$\begin{aligned} J(n,u_n) &= J\left((s_k,t_k)_{k=1}^{n},q\right) \\ &= VD+\frac{D}{nq}\left[Cp+\sum_{i=1}^{n}\mathcal{H}_i+\sum_{i=0}^{n-1}\mathcal{B}_i+\alpha nCt\right]. \end{aligned} \tag{2}$$

(2) The vendor's model

According to Figure 1, the inventory level at time t in each procurement can be characterized by

$$I_V(t)=\begin{cases} \frac{P}{H}t-(i-1)q & \text{, if } t_{i-1}\le t<t_i \ (i=2,\cdots,s) \\ \frac{P}{H}t_s-iq & \text{, if } t_{i-1}\le t<t_i \ (i\ge s+1) \end{cases}$$

where $I_V(\cdot)$ is right-continuous with $I_V(\frac{Hnq}{D})=0$. Hence we can obtain the related holding cost $\mathcal{H}_V$ with respect to the total cost, the amount of inventory carried during the procurement cycle $\left[0,\frac{Hnq}{D}\right]$ multiplied by the unit holding cost in each procurement, given by

$$\mathcal{H}_V=Ch_V\left[\frac{P}{2H}t_s^2-\sum_{i=1}^{s-1}i(t_{i+1}-t_i)q-\sum_{i=s}^{n-1}i(t_{i+1}-t_i)q\right].$$

The total profit of the seller with an integer n is given by

$$\begin{aligned} \pi(n,u_n) &= \pi\left((s_k,t_k)_{k=1}^{n},q\right) \\ &= VD-Cm_VD+\frac{D}{nq}\left[Cp+\mathcal{H}_V+(1-\alpha)nCt\right]. \end{aligned} \tag{3}$$

2.1 Contracts of Supply Chains

Considering the strategy of supply chains, the inventory path during period $\left[0,\frac{Hnq}{D}\right]$ in each procurement can be described by the following.

Contracts of supply chains serve the purpose of shortening the time span of backlogging so that the seller has a higher service level with extra payments and the buyer reduces lost sales and gains the competition in the market with a pooling risk. We make a contract of supply chains that the supplier and buyer share the transportation fee at the ratio α with $0\le\alpha\le 1$. Under the effect of supply chains, the buyer can acquire the vendor's assurance on no lost sales with extra transaction costs. With the assumption of deterministic demand D, the cycle of distribution becomes a constant, denoted by T as shown in Figure 2, and hence we have $nT=\frac{Hnq}{D}$,

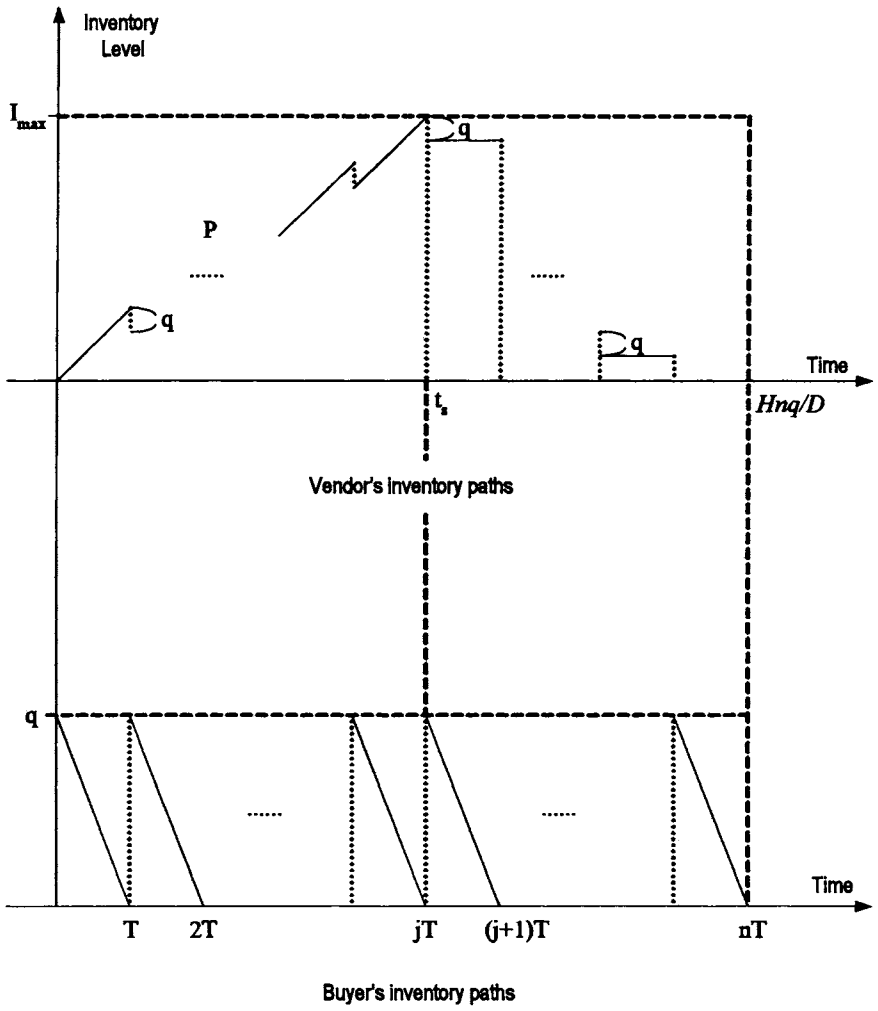


Figure 2: The trajectory of inventory with contracts

that is, $T = \frac{Hq}{D}$. Therefore, the total cost (2) can be obtained in the same process.

$$\begin{aligned} J'(n, u_n) &= J'(n, q) \\ &= VD + CD + \frac{D}{nq}\left[Cp + \sum_{i=1}^{n} \mathcal{H}_i + \alpha nCt\right] \end{aligned} \tag{4}$$

where C is the unit transaction cost and $\mathcal{H}_i$ is the holding cost per distribution under the contracts of supply chains, that is,

$$\mathcal{H}_i = Ch_B \frac{qT}{2} = Ch_B \frac{Hq^2}{2D}.$$

Hence we can obtain the related holding cost $\mathcal{H}_V$ with respect to the total cost, the amount of inventory carried during the procurement cycle $\left[0, \frac{Hnq}{D}\right]$ multiplied by the unit holding cost in each procurement, given by

$$\begin{aligned} \mathcal{H}_V &= Ch_V \left[\frac{P}{2H} t_s^2 - \sum_{i=1}^{s-1} iTq - \sum_{i=s}^{n-1} iTq\right] \\ &= Ch_V \left[\frac{P}{2H} t_s^2 - \sum_{i=1}^{s-1} i\frac{Hq^2}{D} - \sum_{i=s}^{n-1} i\frac{Hq^2}{D}\right]. \end{aligned}$$

The total profit of the seller is given by

$$\begin{aligned} \pi'(n, u_n) &= \pi'(n, q) \\ &= VD + CD - Cm_V D + \frac{D}{nq}\left[Cp + \mathcal{H}_V + (1-\alpha) nCt\right]. \end{aligned} \tag{5}$$

3 Main Results

In this section we will state and prove some existence results for a Stackelberg problem. Making use of the results, we derive the optimal replenishment policy and present a numerical procedure. Let M be a nonempty subset of $\mathbb{R}^n$ and $h : M \to [-\infty, \infty]$. We say that h is proper if h is not identically equal to $+\infty$, or h is not identically equal to $-\infty$. A real-valued function $h : M \to \mathbb{R}$ is called upper semicontinuous on M if for each $x \in M$ and for each sequence $\{x_n\}$ of M converging to x,

$$h(x) \geq \limsup_{n \to \infty} h(x_n).$$

The graph of a mapping S is denoted by

$$\mathrm{Gr}(S) = \{(x, y) \in X \times Y : y \in S(x)\}.$$

Let $\mathbb{Z}_+$ be the collection of positive integers. According to the concept of a Stackelberg equilibrium, problem (SP) can be formulated as follows

$$(\text{SP}) \quad \begin{array}{ll} \max & \pi(n, u_n) \\ \text{s.t.} & u_n \in Sol(n), n \in \mathbb{Z}_+ \end{array}$$

where the decision variable of the seller is $n \in \mathbb{Z}_+$, the decision variables of the buyer are $u_n = \{(s_k, t_k)_{k=1}^n, q\} \in U(n)$, π is the seller's profit function, $Sol : \mathbb{Z}_+ \to 2^{U(n)}$ is a set-valued mapping such that for each $n \in \mathbb{Z}_+$, $Sol(n)$ is the solution/reaction set of the buyer's problem with a given n. This is formulated as

$$\begin{array}{ll} \min & J(n, u_n) \\ \text{s.t.} & u_n \in U(n) \end{array} \tag{6}$$

where J is the buyer's cost function. According to [5, Chapter 8], the problem is called the (mixed-) integer bilevel optimization problem. Even in continuous bilevel optimization the numerical algorithm is complicated and expensive, and so it is for the (mixed-) integer case.

To prove the existence of a Stackelberg equilibrium for the problem, we begin with several properties in discrete convex analysis (see [10]). A partially ordered set Y (poset in short) is a set on which there is defined a binary relation $\leq$ which is reflexive, antisymmetric and transitive. For any $a, b \in Y$, $a \vee b = \sup\{a, b\}$ and $a \wedge b = \inf\{a, b\}$ denote the least upper bound and the greatest lower bound in a component-wise sense, respectively. If the least upper bound and the greatest lower bound of each pair belongs to Y, then Y is a lattice, that is, a lattice is defined as a poset $(Y, \leq)$ whose least upper bound and greatest lower bound are given by $a \vee b$ and $a \wedge b$ are in Y. If $Z \subset Y$ contains the least upper bound and the greatest lower bound of each pair in Z, then Z is a sublattice of Y. Like with convexity, the closure of a sublattice of $\mathbb{R}^n$ is also a sublattice. A nonempty compact sublattice of $\mathbb{R}^n$ has a greatest element and a least element. Let $\theta : X \to \mathbb{R}$ be real-valued on a nonempty lattice X. θ is discretely concave in x if the first-order difference at x, denoted by $\Delta_x \theta(x) = \theta(x+1) - \theta(x)$, is monotone decreasing in x. The function θ is called supermodular (submodular) in x if θ has increasing (decreasing) differences on X, that is,

$$\theta(x \vee x') + \theta(x \wedge x') \geq (\leq)\ \theta(x) + \theta(x') \text{ for any } x, x' \in X.$$

Furthermore, a twice differentiable function θ is supermodular (submodular) if and only if for all $x \in X$,

$$\frac{\partial^2 \theta(x)}{\partial x_j \partial x_k} \geq (\leq)\ 0 \text{ for all } j \neq k \text{ and } x_j \in X_j, x_k \in X_k.$$

We summarize some results about supermodularity (submodularity) in the following Proposition 1 (see [10]).

Proposition 1. Let $\theta : X \to \mathbb{R}$ be real-valued on a nonempty lattice X. Let $X = X_i \times X_{-i}$ be a partitioning of X. Assume θ is supermodular and proper on X. For any $x = (x_i, x_{-i}) \in X$, if $\theta(x_i, \cdot)$ is upper semicontinuous on X_{-i} and X_{-i} is a compact sublattice for each $x_i \in X_i$, then the solution set

$$s(x_i) = \arg\max \{\theta(x_i, x_{-i}) : x_{-i} \in X_{-i}\}$$

is nonempty compact and is a sublattice of X for each $x_i \in X_i$.

Theorem 1. Let $\mathcal{U} = \cup_{n \in \mathbb{Z}_+} U(n)$ and $\pi : \mathbb{Z}_+ \times \mathcal{U} \to \mathbb{R}$. Assume $nq \leq D < \infty$.

(a) π is supermodular on $\mathrm{Gr}(Sol)$;

(b) J is submodular and proper on $\mathbb{Z}_+ \times \mathcal{U}$ and $J(n, \cdot)$ is lower semicontinuous on $U(n)$ with $U(n)$ being a compact sublattice for each $n \in \mathbb{Z}_+$.

Then problem (SP) has a sublattice solution.

Proof. From $nq \leq D < \infty$ we see that $\mathbb{Z}_+ \times U(n)$ is finite for each n. By Proposition 1 and the fact that J is proper, we have the follower's solution set

$$Sol(n) = \arg\min \{J(n, u_n) : u_n \in U(n)\} \text{ for each } n \in \mathbb{Z}_+$$

which is nonempty compact and a sublattice of X. By Proposition 1 again, the leader's problem

$$\sup_{n \in \mathbb{Z}_+} \{\pi(n, u_n) : (n, u_n) \in \mathrm{Gr}(Sol)\},$$

has a sublattice solution since $\mathrm{Gr}(Sol)$ is finite. Consequently, there exist $(n^*, u_n^*) \in \mathbb{Z}_+ \times U(n^*)$ such that $\pi(n^*, u_n^*) = \sup_{n \in \mathbb{Z}_+} \inf_{u_n \in Sol(n)} \{\pi(n, u_n)\}$, and the proof is complete. □

By the concept of a Stackelberg equilibrium, we first analyze the follower's problem (6) for a given n. We find the continuous policy $\{(s_k^*, t_k^*)\}_{k=1}^n$ for the follower as a best response to any decision $n \in \mathbb{Z}_+$ made by the leader via the following necessary conditions:

$$\frac{\partial J(n, u_n)}{\partial s_k} = 0 \text{ and } \frac{\partial J(n, u_n)}{\partial t_k} = 0 \; \forall \; k = 1, \cdots, n.$$

Based on the assumptions (i) and (ii) of supply-demand balance, we understand the continuous policy (the amount of distribution q for each distribution in every procurement cycle) is to fit the requirements from its previous reorder point s_{k-1} to its next reorder point s_k for each period k. Thus we have

$$q = \frac{D}{H}(s_i - s_{i-1}),\ i = 1, \cdots, n.$$

Therefore we have

$$s_i = \frac{s_{i+1} - s_{i-1}}{2} \text{ and } s^*_{i+1} - s^*_{i-1} = \frac{Hq}{D},\ i = 1, \cdots, n. \tag{7}$$

Thus we want to solve for the replenishment point t^*_i. By dynamic programming concepts and (7), minimizing the total cost $J(n, u_n)$ is equivalent to solving n subproblems, and for each $i = 1, \cdots, n$, the ith subproblem (P_i) is given by

$$(\mathrm{P}_i) \quad \inf_{s^*_{i-1} \le t_i \le s^*_i} f(t_i) = \frac{D}{2}\left[Ch_B\left(s^*_i - t_i\right)^2 + Cs_B\left(t_i - s^*_{i-1}\right)^2\right]. \tag{8}$$

Subproblem (P_i) is a convex programming problem, and by the first-order necessary optimality conditions we have

$$\frac{\mathrm{d}f}{\mathrm{d}t_i} = -Ch_B D\left(s^*_i - t_i\right) + Cs_B D\left(t_i - s^*_{i-1}\right) = 0$$

with $\sum_{i=1}^{n}\left(s^*_i - s^*_{i-1}\right) = \frac{Hnq}{D}$, we have

$$s^*_i - t^*_i = \frac{Cs_B}{Ch_B + Cs_B}\left(\frac{Hq}{D}\right) \text{ and } t^*_i - s^*_{i-1} = \frac{Ch_B}{Ch_B + Cs_B}\left(\frac{Hq}{D}\right). \tag{9}$$

Substituting (9) into (8), we have

$$f(t^*_i) = \frac{Ch_B Cs_B\left(Ch_B + Cs_B\right)H^2 q^2}{2D\left(Ch_B + Cs_B\right)^2}.$$

Now we reformulate (2) as a function of n and q as follows:

$$J\left(\left(s^*_k, t^*_k\right)_{k=1}^n, q\right) = VD + \frac{D}{nq}\left[Cp_B + nzq^2 + \alpha nCt_B\right]$$

where

$$z = \frac{Ch_B Cs_B\left(Ch_B + Cs_B\right)H^2}{2D\left(Ch_B + Cs_B\right)^2}.$$

Next we can find the solution $q^*(n)$ for the buyer as a response to any given $n \in \mathbb{Z}_+$ by the vendor, satisfying:

$$\frac{\partial J\left((s_k^*, t_k^*)_{k=1}^n, q\right)}{\partial q} = 0,$$

that is,

$$2nqz - \frac{D}{nq^2}\left(Cp + \alpha nCt\right) = 0.$$

Therefore we have

$$q^*(n) = \left(\frac{D\left(Cp + \alpha nCt\right)}{2n^2 z}\right)^{1/3}. \tag{10}$$

Based on Figure 1 we can see that the stopping-production point t_s, the jth replenishment point t_j, can be found by the assumption of supply-demand balance that the amount of production in $[0, t_s]$ is equal to the amount of requirement in the whole horizon $[0, H]$

$$\frac{P}{H} t_s = nq \Rightarrow t_s = \frac{Hnq}{P}.$$

Due to $t_i^* - s_{i-1}^* = \frac{Ch_B}{Ch_B + Cs_B}\left(\frac{Hq}{D}\right)$ and $t_s = t_j^*$, we have

$$\sum_{i=1}^{j-1}\left(s_i^* - s_{i-1}^*\right) + t_j^* - s_{j-1}^* = t_s,$$

that is, the stopping time of production during a cycle of procurement is

$$j^* = j(n) = \left(\frac{nD}{P} - \frac{Ch_B}{Ch_B + Cs_B}\right) + 1. \tag{11}$$

Hence we obtain the optimal replenishment policy $u_n^* = \left((s_k^*, t_k^*)_{k=1}^n, q^*(n)\right) \in Sol(n)$ for a given n for the buyer's problem.

Remark 1. Note that $(s_k^*, t_k^*)_{k=1}^n = O\left(\frac{1}{n^{2/3}}\right)$ in view of (9), $q^*(n) = O\left(\frac{1}{n^{2/3}}\right)$ in view of (10) and $j(n) = O(n)$ according to (11) where $w_n = O(v_n)$ implies that (w_n/v_n) is bounded. Hence we see that $q^*(n)$ and $(s_k^*, t_k^*)_{k=1}^n$ are monotone decreasing functions of n. We note that the assumption (ii) for the stopping time of production $j(n)$ must be satisfied, which limits the scope of some parameters varying. This shows that the leader has priority over the follower in strategy, that is, the leader's strategy affects the variations in the follower's strategy.

Thus we go to the vendor's problem and find the solution n^* for the vendor who is anticipating the response by the follower. The total profit of the vendor is seen as the implicit function of n, i.e.,

$$\begin{aligned}\pi(n,u_n^*) &= \pi\left((s_k^*,t_k^*)_{k=1}^n, q^*(n)\right)\\ &= VD - Cm_V D + \tfrac{D}{nq(n)}\left[Cp + \mathcal{H}_V(n) + (1-\alpha)\,nCt\right].\end{aligned} \tag{12}$$

Since $n \in \mathbb{Z}_+$, we take the first-order difference of π with respect to n, denoted by

$$\begin{aligned}\Delta_n\pi(n,u_n^*) = &\ \pi\left(n+1,u_{n+1}^*\right) - \pi\left(n,u_n^*\right)\\ = &\ Cp\cdot D\left(\tfrac{1}{(n+1)q^*(n+1)} - \tfrac{1}{nq^*(n)}\right)\\ &+D\left(\tfrac{\mathcal{H}_V^*(n+1)}{(n+1)q^*(n+1)} - \tfrac{\mathcal{H}_V^*(n)}{nq^*(n)}\right)\\ &+(1-\alpha)\,Cp\cdot D\left(\tfrac{1}{q^*(n+1)} - \tfrac{1}{q^*(n)}\right)\end{aligned} \tag{13}$$

where $\mathcal{H}_V^*(n) = Ch_V\left[\frac{H}{2P}n^2 - \frac{H}{D}\left(\frac{n(n-1)}{2}\right)\right](q^*(n))^2$.

Moreover, the Stackelberg equilibrium of the vendor and the follower in SP can be characterized by the next theorem.

Theorem 2. Let the coefficients $(P, D, V, \alpha, H, Ch_B, Cs_B, Ch_V, Cm_V, Cp, Ct)$ be the cost structure of problem (SP). Let $\bar{\pi}$ be an extension of π such that $\Delta_n\bar{\pi}(n,u_n^*)\,|_{n=\tau} = 0$ for some $\tau \in \mathbb{R}$ with $u_n^* \in U(n)$ where $\Delta_n\bar{\pi}(n,u_n^*)\,|_{n=\tau}$ means substituting $n=\tau$ into the function $\Delta_n\bar{\pi}(n,u_n^*)$. Let $\mathcal{U} = \cup_{n\in\mathbb{Z}_+}U(n)$. Suppose the cost structure of problem (SP) supports that $\Delta_n\bar{\pi}(n,u_n^*)$ is monotone decreasing in n and $j(k) \in \mathbb{Z}_+$ for some $n_1 \le k \le n_2$ so that $n_1 \le \tau^* \le n_2 - 1\ \forall\ k, n_1, n_2 \in \mathbb{Z}_+$ where $\Delta_n\bar{\pi}(n,u_n^*)\,|_{n=\tau^*}=0$. Then there exists $(n^*,u_{n^*}^*) \in \mathbb{Z}_+ \times \mathcal{U}$ which is a solution in problem (SP) where

$$n^* \in \arg\max\left\{\pi(n,u_n^*) : j(n)\in\mathbb{Z}_+,\ n_1 \le n \le n_2\right\}. \tag{14}$$

Proof. Based on (14) and the fact that $\Delta_n\bar{\pi}(n,u_n^*)$ is monotone decreasing in n with $\Delta_n\bar{\pi}(n,u_n^*)\,|_{n=\tau} = 0$, we can choose $(n_1,n_2) \in \mathbb{Z}_+ \times \mathbb{Z}_+$ with $n_1 \le \tau \le n_2 - 1$ such that $j(n) \in \mathbb{Z}_+$ for some integer $n \in [n_1,n_2]$. Then we have

$$\pi\left(n_1,u_{n_1}^*\right) \le \bar{\pi}\left(\tau,u_\tau^*\right) \text{ and } \pi\left(n_2,u_{n_2}^*\right) \le \bar{\pi}\left(\tau,u_\tau^*\right).$$

Therefore, we have

$$n^* \in \arg\max\left\{\pi(n,u_n^*) : j(n)\in\mathbb{Z}_+,\ n_1 \le n \le n_2\right\}$$

is well-defined. Consequently, we obtain the solution of problem (SP), $(n^*,u_{n^*}^*) \in \mathbb{Z}_+ \times \mathcal{U}$ such that $\pi(n^*,u_{n^*}^*) = \sup_{n\in\mathbb{Z}_+}\inf_{u_n\in Sol(n)}\{\pi(n,u_n)\}$, and the proof is complete. □

3.1 Optimal Contracts of Supply Chains

Based on the discussion above, problem (SP) under contracts of supply chains can be formulated as

$$(\text{SP'}) \quad \begin{array}{ll} \max & \pi'(n, u_n) \\ \text{s.t.} & u_n \in Sol'(n), n \in \mathbb{Z}_+ \end{array}$$

where $Sol'(n)$ is the solution/reaction set of the buyer's problem under contracts for a given n. The buyer's problem for a given n, is formulated as

$$\begin{array}{ll} \min & J'(n, u_n) \\ \text{s.t.} & u_n \in U(n). \end{array} \tag{15}$$

We can find the solution $q^{**}(n)$ of (15) for the buyer as a response to any decision $n \in \mathbb{Z}_+$ by the vendor under contracts of supply chains, satisfying:

$$\frac{\partial J'(n,q)}{\partial q} = 0,$$

that is,

$$q^{**}(n) = \left(\frac{D(Cp + \alpha nCt)}{nH(Ch_B/2)} \right)^{1/2}. \tag{16}$$

From Figure 2, we can see that in vendor's view the stopping-production point t_s, that is the jth replenishment point t_j, can be determined by the assumption of supply-demand balance that the amount of production in $[0, t_s]$ is equal to the amount of requirement in the whole horizon $[0, H]$

$$\frac{P}{H} t_s = nq \Rightarrow t_s = \frac{Hnq}{P}.$$

Due to $t_s = jT$ and $H = nT$, we see that the stopping time of production during a cycle of procurement is

$$j^{**} = j(n) = \frac{nD}{P} \tag{17}$$

and $\mathcal{H}_V^*(n) = Ch_V \left[\frac{n^3}{2P} - \frac{n^2(n-1)}{2D} \right] (q^{**}(n))^2 T$. By a similar way as in Theorem 2, we can find the Stackelberg equilibrium $(n^{**}, u_{n^{**}}^*) \in \mathbb{Z}_+ \times \mathcal{U}$ such that

$$\pi'(n^{**}, u_{n^{**}}^*) = \sup_{n \in \mathbb{Z}_+} \inf_{u_n \in Sol'(n)} \{\pi'(n, u_n)\}$$

where $Sol'(n) = \arg\min_{u_n \in U(n)} J'(n, u_n)$.

Remark 2. Note that $q^{**}(n) = O\left(\frac{1}{n^{1/2}}\right)$ according to (16) and $j^{**} = O(n)$ in view of (17). The stopping time of production j^{**} must be an integer. Moreover, for a given $n \in \mathbb{Z}_+$, we can compare $q^*(n)$ with $q^{**}(n)$ as follows:

$$q^*(n) < q^{**}(n) \text{ if } \left(\frac{nH}{D^2 Ch_B}\right)^{1/6} > 1.$$

Since $q^*(n) > q^{**}(n)$ is equivalent to

$$\left(\frac{D(Cp + \alpha nCt)}{n^2H^2\left(Ch_B/\left[D\left(1+\frac{Ch_B}{Cs_B}\right)\right]\right)}\right)^{1/3} < \left(\frac{D(Cp+\alpha nCt)}{nH(Ch_B/2)}\right)^{1/2},$$

that is,

$$n^{2/3}H^{2/3}\left(Ch_B/\left[D\left(1+\frac{Ch_B}{Cs_B}\right)\right]\right)^{1/3} > n^{1/2}H^{1/2}(Ch_B/2)^{1/2},$$

that is,

$$\begin{aligned} &\frac{n^{2/3}H^{2/3}\left(Ch_B/\left[D\left(1+\frac{Ch_B}{Cs_B}\right)\right]\right)^{1/3}}{n^{1/2}H^{1/2}(Ch_B/2)^{1/2}} > \frac{\sqrt{2}(nH)^{1/6}}{D^{1/3}(Ch_B)^{1/6}2^{1/3}} \\ &> \left(\frac{nH}{D^2Ch_B}\right)^{1/6} > 1, \end{aligned} \tag{18}$$

since $Ch_B < Cs_B$ by assumption (iv) in the first inequality of (18).

3.2 Algorithm

Since the optimal solution n of the vendor's problem has to be an integer, it is difficult to seek solutions satisfying the mild conditions of Theorem 2. We now propose an algorithm to find them.

Algorithm. Suppose that $Sol(n) \neq \emptyset$ for some $n \in \mathbb{Z}_+$.

Step 1. (Initialization) Select the cost structure satisfying the conditions: $\Delta_n \bar{\pi}(n, u_n^*)$ is monotone decreasing in n and $j(k) \in \mathbb{Z}_+$ for some integer k.

Step 2. Calculate $\Delta_n \bar{\pi}(n, u_n^*)$. If there exists $\tau \in \mathbb{R}$ such that

$$\Delta_n \bar{\pi}(n, u_n^*)\,|_{n=\tau} = 0,$$

then we consider two cases:

(2.1) If $\tau \geq k$, choose n^* by taking $n_2 \in \mathbb{Z}_+$ and $n_2 \geq \tau + 1$,

$$n^* \in \arg\max\{\pi(n, u_n^*) : j(n) \in \mathbb{Z}_+ \ \forall \text{ integers } n \in [k, n_2]\},$$

we obtain the solution $(n^*, u_{n^*}^*)$; STOP.

(2.2) Else, if $\tau < k$, choosing n^* from n_1 to k where $n_1 \geq \tau$ and $n_1 \in \mathbb{Z}_+$,

$$n^* \in \arg\max\{\pi(n, u_n^*) : j(n) \in \mathbb{Z}_+ \ \forall \text{ integer } n \in [n_1, k]\}$$

and we obtain the solution $(n^*, u_{n^*}^*)$; STOP.

Otherwise, for all $\omega \in \mathbb{Z}_+$ we have

$$\Delta_n \pi(n, u_n^*)\,|_{n=\omega} < 0.$$

Then GO TO Step 3.

Step 3. Choose

$$n^* \in \max\{n : j(n) \in \mathbb{Z}_+ \ \forall \text{integers } n\},$$

and we obtain the solution $(n^*, u_{n^*}^*)$; STOP.

Remark 3. Note that the cases of a solution in Step 2 and Step 3 are well-defined in view of Theorem 2 and Step 1.

3.3 Numerical Example

We follow Step 1 to set up the cost structure with different *Cps* for both buyer and vendor as follows:

$H = 1$ year	$D = 100$ units		
$P = 250$ units	$V = \$30$		
$Cp_B = \$5$	$Ch_B = \$2$	$Cs_B = \$3$	$Ct = \$3,$
$Cp_V = \$15$	$Ch_S = \$1$	$C = \$0.5V$	$\alpha = 0.8.$

Then we see that $\Delta_n \bar{\pi}(n, u_n^*)$ and $\Delta_n \bar{\pi}'(n, u_n^*)$ are monotone decreasing in n according to Figures 3 and 4. We can calculate the Stackelberg equilibrium based on the algorithm via Mathematica 4.0 as follows:

Table 1: Summary of results

Without the contract	With the contract
$n^* = 6$	$n^{**} = 5$
$q^* = 16.498304$	$q^{**} = 18.439008$
$t_s^* = 0.395959$	$t_s^{**} = 0.368782$
$j^* = 3$	$j^{**} = 2$
$\left(t_{i-1}^*, s_i^*\right)_{i=1}^n$ shown in Table 2	$T = 0.184391$
$I_V(t_s^*) = 49.494914$	$I_V(t_s^{**}) = 55.317226$
$J\left(n^*, u_n^*\right) = 3034.35$	$J\left(n^{**}, q^{**}\right) = 4541.45$
$\pi\left(n^*, u_n^*\right) = 2446.64$	$\pi\left(n^{**}, q^{**}\right) = 3940.93$

Table 2: Summary of results for reorder and replenishment points

Reorder points	Replenishment points
$s_1^* = 0.164983$	$t_1^* = 0.065993$
$s_2^* = 0.329966$	$t_2^* = 0.230976$
$s_3^* = 0.494949$	$t_3^* = 0.395959$
$s_4^* = 0.659932$	$t_4^* = 0.560942$
$s_5^* = 0.824915$	$t_5^* = 0.725925$
$s_6^* = 1$	$t_6^* = 0.890908$

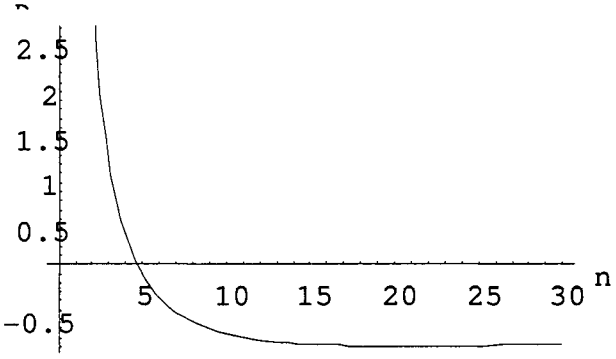


Figure 3: The graph of $\Delta_n \pi\left(n, u_n^*\right)$ without contracts

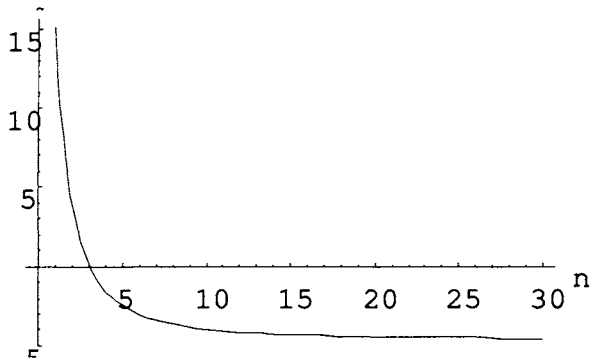


Figure 4: The graph of $\Delta_n \pi'(n, u_n^*)$ *with contracts*

4 Conclusions

In this paper, we consider one-buyer, one-seller, finite horizon, multi-period EOQ/EPQ integrated inventory models. We establish some existence results under some mild assumptions on the cost structure and present a numerical procedure to determine the optimal replenishment equilibrium policy in practice. Based on our example, we find that contracts reduce the number of distributions and the amount of each distribution is higher than without contracts.

References

[1] Aubin JP (1979) Mathematical Methods of Game and Economic Theory, Revised Edition. North-Holland, Amsterdam

[2] Basar T, Olsder GJ (1982) Dynamic Noncooperative Game Theory. Academic Press, New York

[3] Bensoussan A, Crouhy M, Proth JM (1983) Mathematical Theory of Production Planning. North-Holland, Amsterdam

[4] Cachon G, Netessine S (2004) Game theoretic applications in supply chain analysis. In: Simchi-Levi D, Wu SD, Shen Z-J (eds.) Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era, Chapter 2, Kluwer Academic Publishers, Boston, pp. 10–65

[5] Dempe S (2002) Foundations of Bilevel Programming. Kluwer Academic Publishers, Boston

[6] Goyal SK (1976) An integrated inventory model for a single supplier-single customer problem. International Journal of Production Research 15:107–111

[7] Li SX, Huang Z, Ashley A (1996) Improving buyer-seller system cooperation through inventory control. International Journal of Production Economics 43:37–46

[8] Liou YC, Yao JC (2005) Bilevel decision via variational inequalities. Computers and Mathematics with Applications 49:1243–1253

[9] Lu L (1995) A one-vendor multi-buyer inventory model. European Journal of Operational Research 81:312–323

[10] Topkis DM (1998) Supermodularity and Complementarity. Princeton University Press, New Jersey

[11] Yang PC, Wee HM (2001) A single-vendor multi-buyers integrated inventory policy for a deteriorating item. Journal of the Chinese Institute of Industrial Engineers 18(5):23–30

[12] Yang PC, Wee HM (2002) A single-vendor and multiple-buyers production- inventory policy for a deteriorating item. European Journal of Operational Research 143:570–581

Das Potenzial von Operations Research in Transport und Verkehr

Joachim Schmidt
PTV Planung Transport Verkehr AG, Karlsruhe
Vorstand

1 Einführung

Die wenigsten Autofahrer werden einen Bezug zwischen Operations Research – oder kurz OR – und ihrem Navigationssystem herstellen. Genauso wenig denkt man beim Studium der Reiseroute, die man sich gerade im Internet errechnet hat, an Begriffe wie „Minimalgerüst" und „kürzeste Wege". Und was hat die pünktliche Auslieferung der bestellten Möbel mit dem Knoten-orientierten Tourenplanungsproblem zu tun?

Wir, die wir uns mit OR beschäftigen, wissen das natürlich. Viele Erleichterungen des Alltags gäbe es ohne OR gar nicht: Routenplaner auf CD oder im Internet, pünktliche Lieferungen auf Basis professioneller Tourenplanung oder auch immer neue Optimierungen in der Verkehrsplanung. Bei PTV arbeiten wir täglich damit und tragen mit unseren Produkten die positiven Resultate der OR in die ganze Welt.

2 OR von Anfang an

Im November 1979 wurde die PTV AG als Planungsbüro Transport und Verkehr aus der Universität Karlsruhe heraus gegründet. Von Anfang an waren bei der Projektarbeit die Vorlesungen von Professor Neumann, die die „Graphentheorie Transportproblem und Tourenplanungsprobleme" zum Gegenstand hatten, allgegenwärtig. Denn die Arbeitsschwerpunkte der neu gegründeten Firma lagen darauf, Verkehrsabläufe zu simulieren und OR in die Praxis umzusetzen.

Die ersten Projekte waren die Entwicklung eines Dispositionsverfahren für Rufbus am Bodensee, eines der ersten Projekte für nicht liniengebundenen Nahverkehr im ländlichen Raum, dazu die Standort- und Distributionsplanung für die Raiffeisengesellschaft Schleswig-Holstein, eine Liniennetzplanung für Mannheim und die Tourenplanung für die Spaten-Brauerei in München. Diese Projekte stellten gleichzeitig die Grundlage für die ersten Softwareprodukte der jungen PTV dar: INTERTOUR (interaktive Tourenplanung) und INTERNETZ (interaktive Liniennetzplanung).

Die Vision der PTV-Gründer, OR-Kenntnisse in Transport und Verkehr technisch-wissenschaftlich umzusetzen, wurde wahr – getreu dem Motto „Wir machen alles, was man nicht mit Cobol machen kann."

Der Bezug zu mathematischen Optimierungsverfahren ging auch in den folgenden Jahren nicht verloren: 1982 bot PTV die erste Koordinatendatenbank für Transport- und Tourenplanung an. Dieses Produkt lebt immer noch! 1990 war PTV das erste Unternehmen, das Verkehrssimulation auf einem PC offerierte. Heute ist das Unternehmen auf diesem Gebiet weltweit Marktführer. 1999 folgte das erste kommerzielle System für eine Online-Verkehrsprognose. Seit 2000 profitieren die PTV-Kunden von integrierten Softwarelösungen und darauf aufbauenden Informationsdiensten aus einer Hand. Viele moderne Verkehrsmanagementzentralen nutzen PTV-Technologie.

Heute bietet die PTV AG Software, Consulting und Forschung für die Reise-, Verkehrs- und Transportplanung im B2B-Bereich. Europaweit marktführend sind PTV-Produkte wie map&guide zur professionellen Routenplanung, VISUM für die Verkehrsplanung und PTV Intertour für die Tourenplanung. Mittlerweile arbeiten weltweit rund 500 Mitarbeiterinnen und Mitarbeiter für das Unternehmen.

3 Enge Kontakte zu Hochschulen

Die Produkte und Beratungsleistungen der PTV erfordern ein hohes Maß an Know-how, das immer an Personen, die Mitarbeiterinnen und Mitarbeiter der PTV, geknüpft ist. Dass sie außerdem bereit sind, sich ständig weiteres Wissen zu erarbeiten und dieses auch weiterzugeben, ist sicherlich ein wesentlicher Erfolgsfaktor. Durch die sehr engen Kontakte zu den Hochschulen ließen sich viele neue Kollegen und Freunde direkt von dort für das Unternehmen gewinnen. Die PTV betreut einerseits viele Praktikanten und Diplomanden gerade aus den Karlsruher Hochschulen, andererseits kommen viele Absolventen nach ihrem Studium zur PTV.

Die Ausbildung an den Karlsruher Hochschulen, die Anwendung mathematischer Methoden, und damit letztlich das Lehrgebiet, welches auch Professor Neumann am Lehrstuhl für Wirtschaftstheorie und Operations Research vertrat, hat die Entwicklung der PTV nachhaltig beeinflusst.

4 Traffic, Mobility und Logistics: OR als starke Basis

Gemeinsame Basis über alle Geschäftsfelder ist die Anwendung mathematischer Methoden zur Lösung geografischer Planungsaufgaben. Im Geschäftsfeld Traffic

wird simuliert, wie Staus entstehen, oder die optimale Auslastung eines Verkehrsnetzes optimiert. Durch PTV-Software wird ein maximaler Verkehrsfluss an einer Kreuzung durch eine optimale Ampelsteuerung gewährleistet. Die Ansätze basieren auf der Verkehrsforschung an der Universität Karlsruhe, wo in den 70er Jahren erstmals beschrieben wurde, wie die Bewegung von Fahrzeugen und das Verhalten der Fahrer in einem mathematischen Modell abgebildet werden können. Parallel wurde von der PTV ein makroskopisches Modell entwickelt, welches nicht das einzelne Fahrzeug in den Mittelpunkt stellt, sondern die Verkehrsströme. So lassen sich die großräumigen und langfristigen Änderungen von Verkehrsprojekten simulieren.

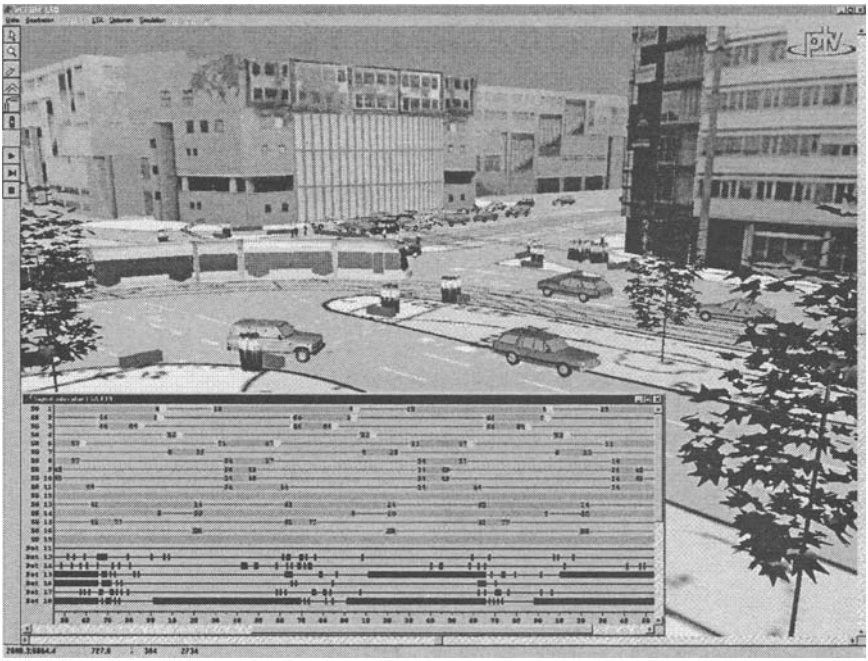


Abbildung 1: 3D-Simulation einer Kreuzung

Die Softwarelösungen und Services im Geschäftsfeld Mobility decken das Gesamtspektrum des individuellen Mobilitätsmanagements ab. Die Berechnung der optimalen Route, aktuelle Verkehrsinformationen oder personalisierte Informationen bezogen auf den geografischen Standort des Nutzers bilden den Schwerpunkt der Arbeiten in diesem Geschäftsfeld. Eine besondere Herausforderung stellt hier die Verarbeitung von zahlreichen dynamisch veränderlichen Informationen bei der Online-Berechnung von optimalen Routen dar.

Das Geschäftsfeld Logistics entwickelt Software und ASP-Dienste für die Tourenplanung und das Flottenmanagement. Professionelle Geomanagement-Lösungen für die Planung und Visualisierung von Liefer- und Vertriebsstrukturen komplettieren das Angebot.

Die Methoden, auf denen die Entwicklungen in den drei Geschäftsfeldern basieren, sind mathematische Optimierungsverfahren, angepasst und erweitert um das Fachwissen des jeweiligen Anwendungsgebiets. So kann z.B. eine aus der modellorientierten Sicht als optimal geltende Tour- oder Fahrtroute aus logistischer Sicht unbefriedigend sein, da sich wichtige Aspekte der „real life" Probleme nicht modellhaft abbilden lassen. Das Kerndilemma dabei ist, dass die Güte einer Lösung erst im Kontext beurteilt werden kann, modellorientierte OR-Probleme hingegen kontextfrei das Optimum anstreben und ausweisen. Deshalb war für die PTV die Kombination aus Problemvisualisierung, Anwendungswissen und einer leistungsfähigen Benutzerinteraktion immer eine entscheidende Produkteigenschaft. Neben der Methodenbereitstellung ist die Verfügbarkeit digitaler Karten und Geodaten ein weiterer Baustein der PTV-Produkte. Das Kombinieren und Verdichten von digitalisierten Netzen und Daten mit Georeferenz erfordert ebenfalls einen methodischen Kern, der sich der OR-Verfahren bedient.

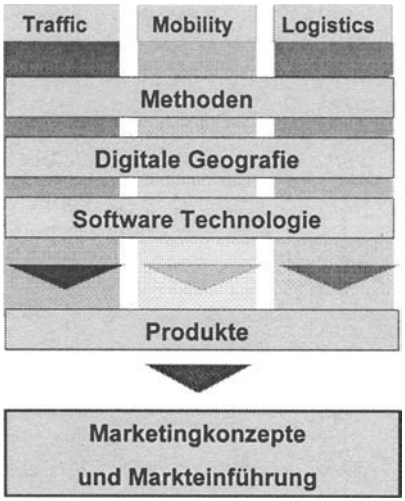


Abbildung 2: Gemeinsame Basis der drei Geschäftsfelder von PTV

Optimierung im Verkehrs- und Transportwesen bedeutet, immer alle Möglichkeiten der Technik auszuloten. Der erste Versuch einer Tourenplanung lief auf einem Taschenrechner TI 41, die primäre seriöse Implementierung auf einem

Hewlett-Packard Grafikterminal. Die erstmalige Implementierung der Verkehrssimulation auf einem PC konnte städtische Netze mit wenigen hundert Strecken berechnen. Heute errechnet VISUM landesweite Verkehrsnetze mit bis zu einer Million Strecken. Aber auch bei wachsender Rechnerleistung werden schnelle, hochperformante Verfahren benötigt, die selbst bei Praxisproblemgrößen von z.B. 5.000 Transportaufträgen und mehreren hundert Fahrzeugen in Sekunden erste, für den Benutzer brauchbare Ergebnisse liefern.

5 Neue Chancen gegen Staus

Als typisches Beispiel für einen stark methodischen und modellorientierten Ansatz kann das Projekt PTV Validate gelten: Das größte Verkehrsmodell der Welt ermittelt Verkehrsbelastungen durch Simulationsrechnungen und setzt neue Maßstäbe mit dem hochaufgelösten digitalen Straßennetz für ganz Deutschland. Um eine sinnvolle Abbildung der Verkehrsströme in Deutschland zu erreichen, ist das Untersuchungsgebiet in etwa 80.000 Marktzellen eingeteilt, die wiederum in circa 7.000 Verkehrszellen zusammengefasst sind. Eine Verkehrszelle repräsentiert ungefähr 12.000 Einwohner, eine Marktzelle etwa 1.000 Einwohner in Deutschland.

Aufgrund dieser Daten berechnet das System, wie viel Verkehr sich zu welcher Zeit auf den Straßen bewegt, und zeigt die Abhängigkeit der Reisezeit von äußeren Einflüssen wie Strukturdaten und nationalen Verkehrsverhaltensmustern (Pendlerdaten, Fahrzwecke etc.). Die Daten zur Verkehrsbelastung erlauben stimmige Rückschlüsse auf die benötigten Reisezeiten.

Damit liegen nun für alle deutschen Hauptstraßen Verkehrsmengenangaben auf einem tiefendigitalisierten Netz mit etwa 1,3 Millionen nach Richtungen getrennten Strecken vor.

Um eine Modellrechnung auf dem Netz durchführen zu können, werden die Strecken mit verkehrlichen Daten angereichert. Insbesondere müssen für alle Strecken Kapazität und Geschwindigkeit bei freier Fahrt bekannt sein. Diese Daten werden zu weiten Teilen direkt aus den Navigationsdaten hochgerechnet, nachrecherchiert und durch Net-Matching mit regionalen Verkehrsmodellen verfeinert. Das bedeutet, sie werden durch die Angaben aus regionalen Verkehrsmodellen ergänzt.

Die Ergebnisse lassen sich wieder auf das Straßennetz zurückführen und somit für die Navigation nutzen.

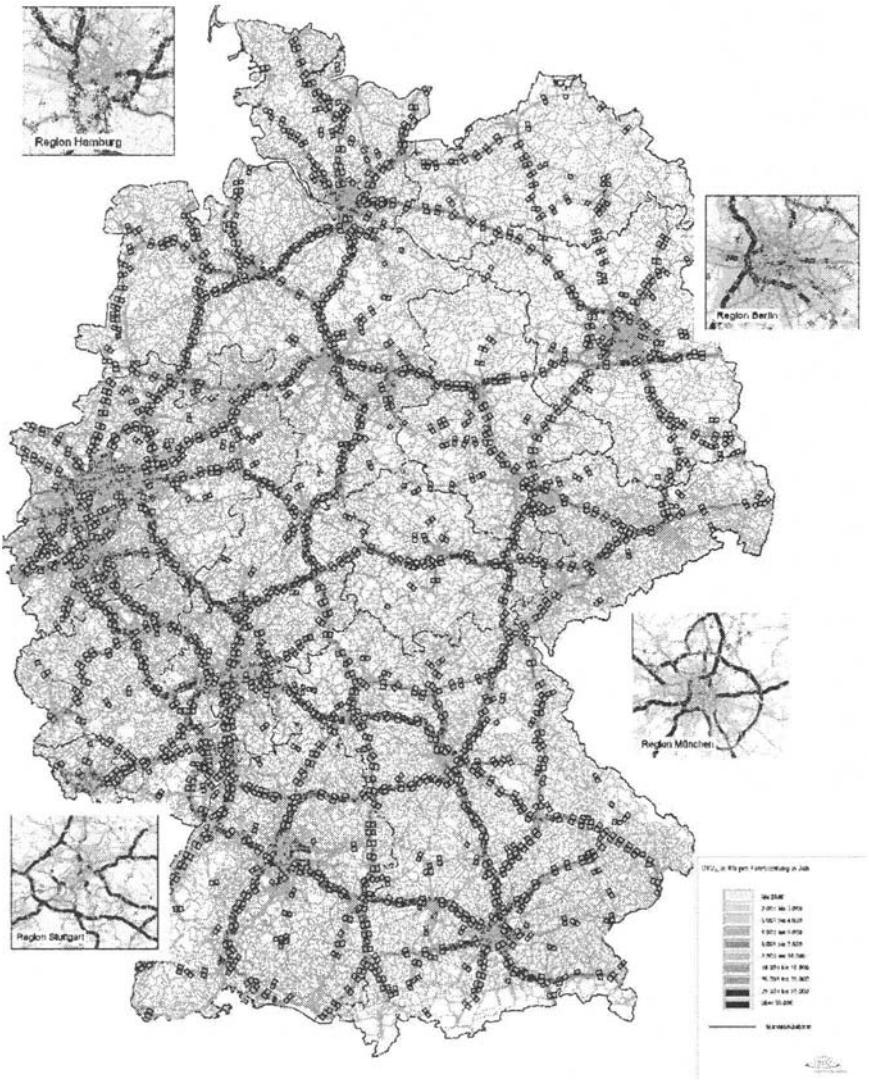


Abbildung 3: PTV Validate, das weltgrößte Verkehrsmodell für den Individualverkehr

6 Integrierte Systeme halten uns mobil

Die Daten aus dem oben beschriebenen Projekt PTV Validate sorgen außerdem für eine realistische Angabe von Reisezeiten bei der Routenplanung. Das ist vor

allem interessant für Navigationssysteme, für die professionelle Tourenplanung oder für Routenplaner. Bei den bisher verfügbaren Ansätzen entspricht die real zu erwartende Ankunftszeit oft nicht der Realität. Grund dafür ist, dass die vorgegebenen Geschwindigkeiten für verschiedene Straßenklassen, auf denen die ermittelten Reisezeiten basieren, unabhängig von Tageszeit und Wochentag und deren typischen oder zu erwartenden Verkehrsbelastungen sind.

Um verlässliche Reise- und Ankunftszeiten unter Berücksichtigung der Verkehrsverhältnisse zu ermitteln, werden die Daten aus PTV Validate ergänzt um Mess- und historische Daten aus Detektoren sowie um die Auswertung von Verkehrsmeldungen. Hinzu kommen Daten, welche die im Verkehr befindlichen Fahrzeuge mittels Telematikeinrichtungen selbst generieren, sogenannte Floating Car Data (FCD). Das Ergebnis: Fahrzeiten abhängig vom Wochentag, der Tageszeit oder auch von Ferienzeiten oder dem Wetter.

Vor allem professionelle Nutzer profitieren davon. Denn werden neuralgische Staupunkte gezielt umfahren, sorgt das beispielsweise in der Logistik für eine effizientere Auftragsabwicklung. Etwa, indem die Auslieferstrecke entgegen der Hauptverkehrsströme geplant wird, da die prognostizierte Verkehrsbelastung der Stecken dem Optimierungsmodell der Tourenplanung zur Verfügung steht.

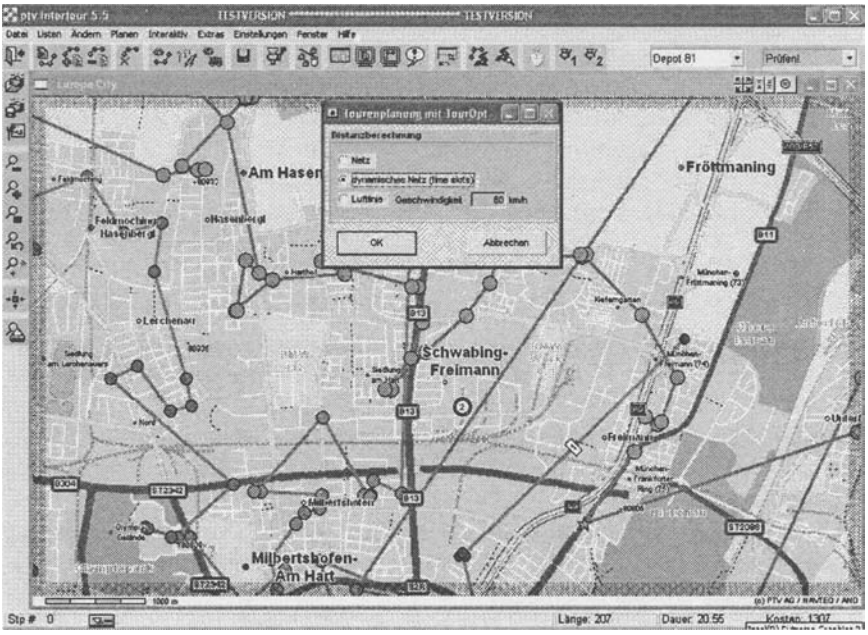


Abbildung 4: Verlässlichere Tourenplanung für die Stadtlogistik durch realistischere Fahrtzeiten

PTV Intertour „weiß" so bereits bei der Vorplanung, wo Staus sein werden. Das ermöglicht, Straßen, auf denen zu bestimmten Zeiten Staus und Verkehrsbehinderungen herrschen, einfach zu umfahren. Damit lässt sich die für die Auslieferung im City-Bereich benötigte Fahrzeit reduzieren und mehr Aufträge können innerhalb einer Tour abgewickelt werden.

Zudem bietet eine Routenempfehlung, die verkehrliche Engpässe ausweist, flexibleren Verkehrsteilnehmern die Chance, zu einer günstigeren Zeit zu reisen. Wer terminlich gebunden ist, kann immerhin Staus umfahren, wodurch sich die Verkehrslast gleichmäßiger verteilt.

7 Verkehrsmeldungen aus dem fließenden Verkehr – damit der Verkehr im Fluss bleibt

Ziel der weiteren Forschung und Entwicklung bei PTV ist es, automatisch Verkehrsmeldungen zu erzeugen. Die Ausgangsbasis sind Verkehrsdaten bzw. -informationen, die aus unterschiedlichen Quellen stammen können. Beispielsweise aus den Verkehrzentralen der Bundesländer, der Landesmeldestelle, vom ADAC oder auch direkt aus dem fließenden Verkehr, z.B. über Handys, die in Fahrzeugen mitgeführt werden und damit Floating Car Data generieren.

Die Prozesskette von den Daten zu den Verkehrsinformationen geht über ihre Erhebung, Veredelung, Prüfung und ihren Versand. Die Quelldaten und -informationen werden bei PTV in einem speziell dafür entwickelten Verfahren kontinuierlich überlagert und miteinander verschnitten. Unter anderem werden Daten aus in der Fahrbahn bereits vorhandenen Meldeschleifen verarbeitet. Ergebnis dieser „Meldungsfusion" der einzelnen Datenquellen sind hochwertige Verkehrsinformationen, die aktuell, zuverlässig und korrekt sind und das Straßennetz optimal abdecken.

Die auf diese Art erzeugten PTV-Meldungen werden kontinuierlich an eine Verkehrsredaktion übermittelt, wodurch die Möglichkeit gegeben ist, jede einzelne Meldung nochmals redaktionell zu bearbeiten. Danach werden die Meldungen über encrypted Traffic Message Channel (eTMC) – einem verschlüsselten Datenkanal – ausgestrahlt und den Verkehrsteilnehmern zu Verfügung gestellt. Der Weg der Information bis zum Fahrer dauert dabei höchstens zehn Minuten.

Der verschlüsselte Datenkanal ist eine der Möglichkeiten, die Radios und moderne Navigationssysteme bieten, um Verkehrsmeldungen auf einem Datenkanal per RDS/TMC zu empfangen. Der Fahrer erhält die Information entweder in Form einer Liste von Verkehrsmeldungen oder symbolisch dargestellt auf einer Straßenkarte. Die Daten können so auch für das dynamische Routing bei

der Offboard Navigation verwendet werden. So erhält der Fahrer eine Zielführung, die die aktuelle Verkehrslage berücksichtigt.

PTV fördert durch innovative Technik- und Softwareentwicklungen die Verfügbarkeit besserer Verkehrsmeldungen. Das schafft die Voraussetzung, den Kunden unterwegs mit allen gewünschten Informationen über seine Route zu versorgen.

Das Projekt DIANA (Dynamic Information And Navigation Assistance) befindet sich derzeit noch im Pilotbetrieb und durchläuft gerade die Phase der Qualitätssicherung. Darin werden die von PTV generierten Verkehrsinformationen getestet und optimiert. DIANA ist Teil des Modellprojekts Staufreies Hessen 2015. Die Verkehrsdatenplattform der PTV ist schon heute eine Quelle für hochwertige Verkehrsinformationen für das Bundesgebiet sowie für das benachbarte Ausland.

8 Neue Kartentechnologie teilt die Welt in Kacheln ein

Um digitale Straßenkarten auf den neuesten Stand zu bringen, mussten früher die kompletten Daten ausgetauscht und neu installiert werden. Das ist ab sofort überflüssig dank einer neuen Kartentechnologie von PTV. Die Scalable Map Architecture ermöglicht, digitale Kartendaten dynamisch und selektiv zu aktualisieren und zu ergänzen – eine ausgezeichnete Grundlage, um alle Arten von maßgeschneiderten Mobilitätsdiensten einfach umzusetzen.

Das neue Kartenformat teilt die ganze Welt in fest definierte, weltweit einheitliche, quadratische Kacheln ein. Allein Deutschland ist in 550.000 Kacheln gegliedert. Diese Kachelung erlaubt die zentrale, partielle Aktualisierung der Geodaten – genau und nur in den Kacheln, in denen sich etwas geändert hat. Die Lösung zeigt eine wesentlich höhere Performance, da die zu übermittelnden Daten beim Update deutlich kleiner sind. So werden aktualisierte Kartenausschnitte ohne Zeitverluste angezeigt.

Die Anforderungen an digitale Kartendaten hängen von der Art der Anwendung ab: Für eine grobe Orientierung kann eine Ortsangabe auf 10 km genau genügen, doch zukünftige Fahrerassistenzsysteme möchten auf 10 cm die Position der Leitlinie kennen. Manchen Anwendungen reichen Karten, die 10 km^2 abdecken, andere brauchen weltweit detaillierte Informationen. Und wie häufig sollen die Daten aktualisiert werden? Einmal im Jahr oder täglich, um beispielsweise Verkehrsbehinderungen aktuell abzubilden? Nicht zuletzt ist die Kompatibilität wichtig, damit die Kartendaten auf allen Endgeräten verfügbar sind.

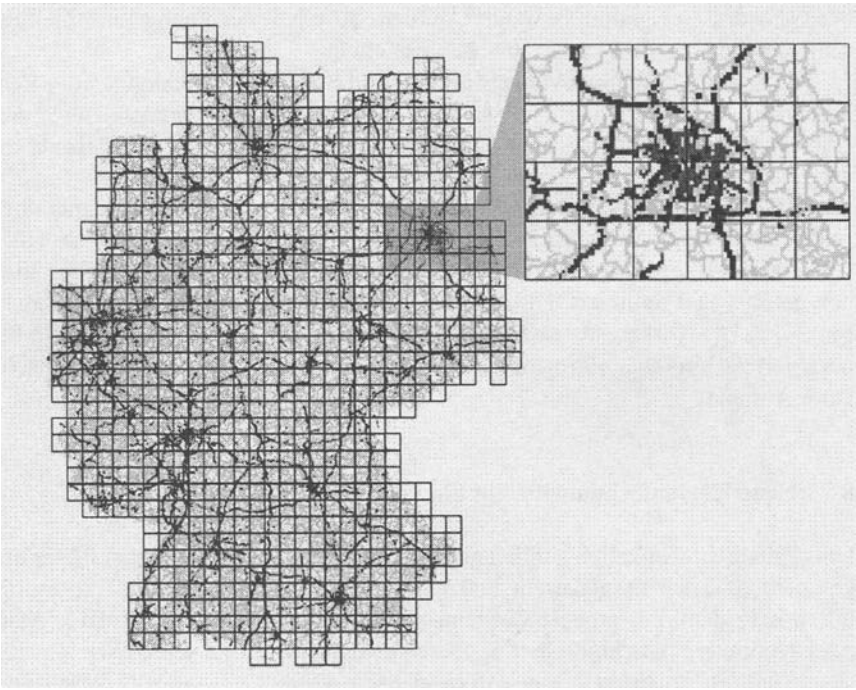


Abbildung 5: Berlin - kachelweise schnell aktuell dank Kartentechnologie von PTV

Das neue Kartenformat ist eine der Grundlagen für die hybride Navigation und ist daher besonders für Automobilunternehmen und Telekommunikationsanbieter interessant. Die Kartenansicht kann jeweils an die spezifischen Bedürfnisse in der Anwendung ausgerichtet werden. Mehrere Kacheln lassen sich beispielsweise zu größeren Kacheln zusammenfassen: in verschiedenen Auflösungen, von $1m^2$ bis zur ganzen Welt. Zudem gibt es die Karten in unterschiedlichen Informationstiefen: vom Bundesautobahnnetz bis zu detaillierten Nebenstraßen-Informationen.

9 OR – heute und in Zukunft

OR gehört bei PTV immer zu den aktuellen Themen. Methodenorientierung ist weiterhin Markenzeichen des Unternehmens. Die verfahrensorientierte Ausbildung an der Karlsruher Universität bildet die Grundlage für den international erfolgreichen Konzern.

Ein wichtiges Betätigungsfeld der PTV ist weiterhin der Brückenschlag zwischen transportlogistischen oder verkehrlichen Aufgabenstellungen und den Verfahren der Operations Research.

In Zukunft bleibt die Wahl des methodischen Ansatzes weiterhin wichtig, damit Verkehrsprognosen, integrierte Modelle und selbststeuernde Systeme im Verkehr helfen, auch bei steigendem Verkehr weiter Mobilität gewährleisten zu können.

Part IV
Interdisciplinary Dimensions

Evolution of Conventions in Populations with Local Interaction Structures

Siegfried K. Berninghaus

Institut für Wirtschaftstheorie und Operations Research
Universität Karlsruhe

1 Introduction

It is almost common knowledge in modern societies that *conventions* regulate many important economic and social processes. Nevertheless, this topic has not received adequate attention in the past neither from empirically oriented nor from theoretical economists. It is the main purpose of this paper to analyze this problem using an analytical model of strategy adaptation in populations with a given social structure (network). In most studies on evolutionary strategy adaptation it is assumed that members of a population are randomly matched with each other member of the population. Seemingly, this does not give a realistic scenario of strategy adaptation in most modern societies where members of a population are matched only with members of their reference groups like families, colleagues at work, etc. However, these groups are typically not isolated from each other. They are interrelated by individual connections, which makes the strategy adaptation problem of a single individual in the population a non-trivial strategic decision problem.

Before we start our analysis we give some familiar examples of conventions[1] which indicate the relevance of this field for strategic interaction in large populations:

- Suppose two people want to meet each other in a foreign city. What are the favorite meeting points for the people in this city? If the people we consider are artists, then they will probably try to find out which bars or restaurants belong to the artists' scene.

- Suppose a telephone call between two people is unexpectedly cut off after one minute. If both people are interested in the connection being restored, who will be the first one to call back?

[1] All examples are taken from Lewis (1969), which is an inspiring source of examples of conventions in economics and in social life.

- Consider people driving in opposite directions on the same winding two-lane road. It is not important for any of them on which side of the lane they drive, however it is extremely dangerous if two of them drive on the same lane. In order to escape from such terrible accidents everybody should use the same convention, that is either driving on the left or on the right lane.

- Consider an invitation to a party. In principle, it does not matter how everybody is dressed. But one would probably be embarrassed if one finds all guests almost dressed alike and is dressed differently oneself. Everybody must dress according to their expectations of how the others might be dressed. If everyone is dressed like a clown you should also be dressed like a clown, but it would be ridiculous to wear a clown suit on workdays in one's office. Again, how to be dressed for a particular party is a problem of dressing conventions.

- Suppose two tradesmen want to exchange different goods. In most societies a particular commodity is taken in exchange for goods. It does not matter to most of us which particular commodity is taken in exchange for goods, it may be wampum, it may be goats. However, it is very important that each tradesman takes what the other one does not refuse. Otherwise, he has to return without his good being sold. We learn from this example that currencies help to solve serious commodity exchange problems which are essentially coordination problems. Later on, we will see how conventions and coordination problems are tied together.

- Finally, we consider Rousseau's example (from *Discours sur l'inegalite*) of the *stag hunt* which recently stimulated many experimental investigations in the evolution of conventions. In this highly stylized example we consider a tribe of hunters and gatherers living in the wilderness without food. Each man can hunt rabbits separately and eat badly. Together, the hunters can catch stags and eat well. However, if at least one man deserts the stag hunt to catch a rabbit the hunt will not be successful. Moreover, there is a strong incentive for the remaining hunters to desert the stag hunt in order to at least eat a rabbit. Both types of behavior (stag hunt or rabbit hunt) can be regarded as conventions. A hunter will stay with the stag hunt if he believes that the others stay, too. And he believes that the others will stay when it has become a convention to hunt stags in this society.

Extracting the common character of these examples, we see that conventions are basically used to solve a *coordination problem.* Let us explain

this in more detail. We consider two ore more agents who have to choose one of several alternative actions in a strategic decision problem. In many situations, the set of alternative actions is the same for each decision maker involved. However, this is not necessary. If agents choose the same action every time they interact with other agents in the population a *regularity of behavior* will result. If everybody expects the other agents to conform to this regularity we call this a convention, provided no individual decision maker has an incentive to deviate from this regularity. We make this concept more precise by the following definition (see Lewis, 1969, p. 42).

Definition 1 *A regularity R in the behavior of the members of a population P when they are agents in a recurrent situation S is a convention if and only if, in any instance of S among members of P,*

1. *everyone conforms to R;*
2. *everyone expects everyone else to conform to R;*
3. *everyone prefers to conform to R on the condition that the others do,too, since S is a coordination problem and uniform conformity to R is a coordination equilibrium in S.*

It is essential for our introductory examples that there is no unique solution for the underlying strategic decision problem. Quite on the contrary, in each of these recurrent situations different regularities in behavior could evolve. For example, in the money exchange problem different commodities could be used for exchange money which is accepted on a market place. Only some basic requirements for this commodity need to be satisfied. It has to be storable for a longer time period, and it should not have too much weight so that tradesmen can carry it easily from one marketplace to the other. Or let us consider the problem of driving on one particular lane of the road. It is completely unimportant which particular lane is used. Both conventions, either driving on the left side or on the right side of the road could evolve. Only coordination failure (both drivers use different lanes) would induce negative results for both drivers.

As a further common characteristic of our examples the regularities in behavior, which are regarded as conventions, have an *equilibrium property.* That is, it is not profitable for an individual agent to deviate from the convention when the other agents still conform to it. For example, let us consider the stag hunt game. If the stag hunt evolved as a convention in a group of hunters a single agent would certainly not benefit at all from switching to rabbit hunting. He would have to eat rabbits instead of stag meat. On the other hand, if hunting rabbits became the hunting convention

in the group it would not make sense for a single agent to turn to stag hunting. Because then, he would neither eat rabbits nor stag meat. The same implication holds for the car driving example. If all agents conform to the rule to drive on the right lane of the road, obviously, it will not pay for a single agent to deviate from this convention.

From a game-theoretical perspective one can say that conventions help to solve the problem of *equilibrium selection* in coordination games recurrent in a population of players. Coordination games typically have multiple equilibria which sometimes may even be Pareto-ranked by their payoffs (symmetric coordination games).[2] If such games are played recurrently by players who are matched with members of their social reference groups to choose their strategy it will not be a trivial task for a single agent to forecast which equilibrium strategy his opponent will choose. In such situations, it may help the players to conform to a convention and choose a particular equilibrium strategy of the coordination game which is played by all agents in the population.

There are many different ways in which conventions become established in a society. We mention the two most important ones:

- Conventions may be imposed by a central authority. For example, during the French Revolution it was decreed that carriages in Paris should keep to the right lane (for details see Hopper, 1982), whereas the previous convention was for carriages to keep to the left lane and for pedestrians to use the right lane. Establishing this "opposite" convention at that time would have to be regarded as a symbol for the new order which had been established by the revolutionary regime. "Using the right lane" would then be the habit of the common man after the French Revolution.

- Conventions in a society may evolve through an "evolutionary process". The idea behind that is that as an interaction between agents in a population is repeated over and over by many individuals a regularity in behavior may be established at least in some parts of the population. These regularities may "infect" other subgroups so that after some time it becomes a custom in the whole population to use a certain action for solving a particular coordination problem. For example, driving on the left lane in Britain started out being a local custom which then spread over the regions of the country.

[2] In asymmetric coordination games (for example, the battle-of-the-sexes game), players may value equilibrium payoffs differently.

Of course, both procedures are not incompatible. In many cases, conventions which have already evolved through an evolutionary process in a society were later on codified into law.

In this paper, we are exclusively interested in the evolution of conventions as a result of individual choices of agents in a population playing a non-cooperative coordination game. In particular, we want to address two problems:

a) How will conventions evolve? Which pieces of information do we need to predict which conventions will finally evolve in a society?

b) From our examples we know that there may exist "good" or "bad" conventions, i.e. "good" or "bad" equilibria in the base coordination game. How can a society break out from an inferior convention?

In contrast to stochastic models of strategy adaptation which have been developed during the past decade (see, for example, Kandori, Mailath, and Rob, 1993, Young, 1993) we use a purely deterministic strategic decision approach. Using a deterministic approach we believe to be able to explain more complex phenomena of convention formation.

We will focus on the evolution of conventions in populations with fixed social networks. In Berninghaus and Schwalbe (1996a, 1996b) the evolution of conventions on two different fixed local interaction structures (circular structure and lattice) in a population with an odd number of individuals were studied. In this paper, we relax some of these restrictions and gain better insight into the process of strategy adaptation.

In the second part of the paper, we link the theoretical results to experiments on the evolution of conventions which have been conducted in our experimental lab at the University of Karlsruhe. Many experiments have been conducted on the evolution of conventions in an evolutionary framework (for a survey of recent results see Berninghaus and Ehrhart, 2001), but we do not know of many in which the social network aspect has been considered as in Berninghaus et al. (2002).

2 The Model

2.1 Basic Aspects

Almost all work on the evolution of conventions has been based on a highly stylized example of a 2-person coordination game the payoff table of which is given below.

	X	Y
X	(a,a)	(b,c)
Y	(c,b)	(d,d)

It is assumed that $a,b,c,d \in \mathbb{R}$ and that X and Y are the only alternatives the agents in the population can choose from when they are matched to play this symmetrical 2×2 game. Formally, this game is a normal form game $G = \{\Sigma_1, \Sigma_2;\ H_1(\cdot), H_2(\cdot)\}$ with identical strategy sets $\Sigma_1 = \Sigma_2 = \{X, Y\}$ and (symmetric) payoff functions $H_i(\cdot)$, that is, we have $H_1(X,Y) = H_2(Y,X)$. Because of symmetry this simple game can be completely characterized by its payoff matrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

In order to characterize game G as a (symmetric) coordination game we have to assume

$$a > c, \quad d > b.$$

This game has two pure strategy[3] Nash-equilibria (X,X) and (Y,Y) which can be Pareto-ranked according to their payoff level. If $a < d$ then the equilibrium (Y,Y) is called payoff-dominant. If the rationality of the players is common knowledge, each player would expect the opponent player to select the payoff superior equilibrium strategy, i.e. Y whenever they are matched and the convention "to choose Y" would immediately be established in the population. In this paper, we exclude common knowledge of rationality from our considerations. The populations we have in mind typically may be rather large and may be composed of people with different degree of rationality. In such a setting *bounded rational behavior* (characterized by myopic optimization) is the appropriate behavioral assumption.

Because of bounded rationality it is not quite clear which equilibrium strategy may be played if an agent is matched with a member in his reference group. If a player has absolutely no idea about his partner's strategy choice he might expect that X and Y are played with a probability $\frac{1}{2}$ each. Under this assumption it is more profitable to choose X when inequality

$$\frac{1}{2}a + \frac{1}{2}b > \frac{1}{2}c + \frac{1}{2}d$$

holds which is equivalent to

$$a - c > d - b. \tag{1}$$

[3]In our paper we do not consider mixed strategy equilibria.

If inequality 1 holds, strategy combination (X, X) is called *risk-dominant equilibrium* (for more detailed information see Harsanyi and Selten, 1988).

It is easy to see that there exist coordination games in which the risk-dominant equilibrium is payoff inferior.[4] Although risk-dominant equilibria may be payoff-inferior, a society in which perfect rationality is not common knowledge may get stuck with such unfavorable conventions. Even worse, in recent theoretical and experimental investigations (e.g., Kandori, Mailath, and Rob, 1993, Amir and Berninghaus, 1996, Berninghaus et al, 2002) it was shown that risk-dominant equilibria prove to be very attractive for strategy selection. In the following, we will further elaborate on this point.

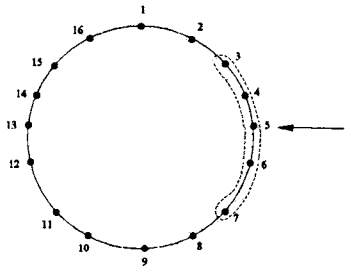
2.2 A Binary Formal Framework for the Evolution of Conventions

We consider a finite population $I = \{1, \ldots, n\}$ of players. A local interaction structure imposed on the population I is represented by the symmetric interaction matrix $A = \{a_{ij}\}_{i,j}$ with

$$a_{ij} = \begin{cases} 1 & \ldots\ldots \quad j \text{ is neighbor of } i \\ 0 & \ldots\ldots \quad \text{no connection between } i \text{ and } j \end{cases}$$

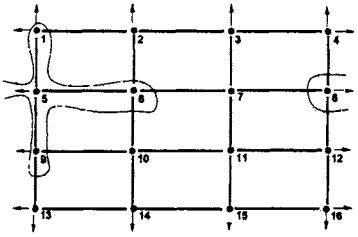
for $i \neq j$ and $a_{ii} = 0$. Two different local interaction structures have been utilized in the literature up to now (see Ellison, 1993, Berninghaus and Schwalbe, 1996a):

- The symmetric circular local interaction structure (with k right and left neighbors each). We illustrate this by a circle with population size $n = 16$ and neighborhood size $N(i) = 4$



- the lattice local interaction structure (with 4 neighbors each, where one is situated to the right, one to the left, one on the top, and one on the bottom), see, for example, a lattice with $n = 16$

[4] Consider, for example, a 2×2 game with $a = 3, b = 2, c = 0$, and $d = 4$. This implies $a - c > d - b$ but $H(X, X) < H(Y, Y)$.



We will generalize the results in Berninghaus and Schwalbe (1996a) by relaxing some restrictive assumptions and to get better insight into the dynamics of strategy choice. However, we will keep the behavioral assumption on *myopic local best reply*, i.e. we assume

Assumption 1 *Given a strategy configuration* $\sigma^{t-1} \in \{X, Y\}^n$ *in period* $t-1$, *then agent* i *will choose*[5] *in period* t

$$\sigma_i^t = \begin{cases} X & \ldots\ldots \ \frac{1}{N(i)} \sum_{j \in N(i)} H_i(X, \sigma_j^{t-1}) > \frac{1}{N(i)} \sum_{j \in N(i)} H_i(Y, \sigma_j^{t-1}) \\ Y & \ldots\ldots \ \frac{1}{N(i)} \sum_{j \in N(i)} H_i(X, \sigma_j^{t-1}) < \frac{1}{N(i)} \sum_{j \in N(i)} H_i(Y, \sigma_j^{t-1}) \end{cases}$$

where $N(i)$ *denotes the set of* i*'s neighbors.*

Assumption 1 is commonly used in theoretical investigations on strategy choice of locally interacting agents. It simply says that agent i in period t will choose the strategy giving the larger expected payoff when played against his neighbor's strategies of the previous period.

In order to simplify notation, in this section we use the convention to denote strategy X in the coordination base game by "1" and strategy Y by "0". Further, let us denote by $N(i)$ the set of neighbors of agent i and let us denote by n_i the number of players in $N(i)$ choosing strategy 1 in a given period $t-1$ then according to assumption 1 player i will choose 1 in period t if

$$\frac{n_i}{|N(i)|} a + (1 - \frac{n_i}{|N(i)|} b > \frac{n_i}{|N(i)|} c + (1 - \frac{n_i}{|N(i)|} d \tag{2}$$

holds which is equivalent to

$$n_i > |N(i)| \frac{d-b}{a-c+d-b}. \tag{3}$$

[5]Note, that we do not consider tie breaking rules. This can be justified by determining the payoffs values appropriately.

By defining $b_i := |N(i)|\frac{d-b}{a-c+d-b}$ and expressing the prevailing local interaction structure in the population by the matrix A the condition 3 can be expressed as follows

$$\sum_j a_{ij}\sigma_j^{t-1} > b_i. \tag{4}$$

By defining the *threshold function* of player i by

$$f_i := \begin{cases} 1 & \dots \quad \sum_j a_{ij}\sigma_j^{t-1} > b_i \\ 0 & \dots \quad \sum_j a_{ij}\sigma_j^{t-1} < b_i \end{cases}$$

we can express the myopic behavior of an individual player by his threshold function

$$f_i : \sigma \longmapsto \{0,1\}$$

which associates with each strategy configuration σ in the previous period a strategy $\sigma_i \in \{0,1\}$ in the coordination game that is played against all neighbors in the present period.

We are interested in strategy configurations σ which are stable with respect to myopic best reply. By adapting the Nash equilibrium concept to our particular situation we define a new equilibrium concept below.

Definition 2 *A strategy configuration σ^* is local Nash iff*

$$\forall i : \frac{1}{|N(i)|}\sum_{j\in N(i)} H_i(f_i(\sigma^*), \sigma_j^*) \geq \frac{1}{|N(i)|}\sum_{j\in N(i)} H_i(\sigma_i, \sigma_j^*)$$

for any $\sigma_i \in \{0,1\}$.

Local Nash configurations are the natural candidates for **conventions.**[6]

Supposing that all players adapt their strategy choice in each period simultaneously[7], the global strategy adaptation can be formally described by the adaptation function

$$F := (f_1, \dots, f_n) : \{0,1\}^n \longrightarrow \{0,1\}^n.$$

[6]Some authors use the concept of a convention in a more restrictive sense. They require the dynamic stability of local Nash configuration to be called convention.

[7]The term "simultaneous" should not be interpreted too literally. We only want to model a situation in which no player knows the strategy choice of his opponents in the same period.

We reduce the problem of simultaneous strategy adaptation to a well-known problem in discrete dynamics: Given a starting strategy configuration σ^0, how will the iterates of the global strategy adaptation $F^t(\sigma^0) = F(F(....(\sigma^0))$ behave? Will the sequence of iterates converge to a fixed point or to a limit cycle?

It can easily be seen that a fixed point of $F(\cdot)$ is a local Nash configuration resp. a convention in the population. Some definitive answers to the questions can be found in the literature on *threshold automata* (e.g., Goles, 1985, Fogelman, Goles and Weisbuch, 1983, Robert, 1986) and in the literature on *Evolutionary Games* (e.g., Berninghaus and Schwalbe, 1996a, 1996b). Unfortunately, there is no general theory on the behavior of the iterates $\{F^t\}$ for arbitrary local interaction structures. However, for particular local interaction structures we completely know the behavior of global strategy adaptation which will be illustrated by the results below. We will consider three different types regular of local interaction structures A:

α) The circular interaction structure with neighborhood size $|N(i)| = 2$,

β) the circular interaction structure with neighborhood size $|N(i)| = 4$,

γ) the lattice interaction structure with von-Neumann neighborhoods, i.e. $|N(i)| = 4$ (see the drawing above).

For local interaction structures b) and c) the following assumption will turn out to be useful to determine strategy adaptation over time.

Assumption 2 *Given the 2-person coordination game (introduced in subsection 2.1) with payoffs a, b, c, and d. Then, the following inequality*

$$\frac{1}{4} > \frac{d-b}{a-c+d-b}.$$

holds.

Essentially, assumption 2 sharpens the risk dominance of strategy 1. According to this assumption strategy 1 could be called *strictly risk dominant.* The following result contains the main implications of this assumption.

Result 1 *Suppose assumption 2 holds, then player i will choose strategy* 1 *if at least* one *player in his reference group (neighborhood) selects* 1.

Proof: a) According to inequality 3 the statement is true for local interaction structure α) irrespectively of assumption 2 since risk dominance of strategy 1 implies

$$\frac{d-b}{a-c+d-b} < \frac{1}{2}.$$

b) For local interaction structures β) and γ) the statement immediately follows from assumption 2 combined with inequality 3.

q.e.d.

Based on result 1 we conclude the following main result on dynamic strategy adaptation in our model.

Result 2 *a) Given any of the local interaction structures* $\alpha) - \gamma)$. *There exist only two fixed points* $\sigma^* = (1, 1, \ldots, 1)$ *and* $\sigma^{**} = (0, 0, \ldots, 0)$.

b) In local interaction structures $\alpha)$ *and* $\gamma)$ *with* n *odd and in local interaction structure* $\beta)$ *there do not exist limit cycles.*

c) For local interaction structures $\alpha)$ *and* $\gamma)$ *with* n *odd and for local interaction structure* $\beta)$ *the basins of attraction are*

$$B(\sigma^*) = \{0,1\}^n - (0, 0, \ldots, 0), \quad B(\sigma^{**}) = (0, 0, \ldots, 0).$$

Proof: a) Suppose there exists a fixed point $\hat{\sigma}$ of $F(\cdot)$ that is different from σ^* and σ^{**}. Then, there exist at least two members i, j with $j \in N(i)$ and $i \in N(j)$ such that $\hat{\sigma}_i = 1$ and $\hat{\sigma}_j = 0$. As myopic (local) best reply j will choose 1 in the following period which implies $F(\hat{\sigma}) \neq \hat{\sigma}$, i.e. $\hat{\sigma}$ cannot be a fixed point of $F(\cdot)$.

b) It is easy to see that the local interaction matrix A is *primitive* for the local interaction structures proposed in b), that is, there exists a $t < \infty$ such that A^t consists only of 1's. Thus, we have $A^t \sigma^0 = (1, 1, \ldots, 1)$ for any $\sigma^0 \neq (0, 0, \ldots, 0)$. There cannot exist a cycle.

c) Follows directly from the primitivity of A and part b).

q.e.d.

Remarks: 1) In result 2 we omitted the case of local interaction structures $\alpha)$ and *gamma*) with n even because of space restriction. This case beside the fixed points σ^* and σ^{**} admits limit cycles with a particular basin of attraction.

2) We should keep in mind that most results in our paper hold under assumption 2. Results may change drastically if this assumption is violated.

Economic Interpretation: *We conclude that risk-dominant conventions play a dominant role in strategy adaptation. This implies that there will be a strong tendency in a population to approach the inferior convention*

even when risk dominant and payoff dominant conventions do not coincide. This may also explain why societies sometimes get stuck with an inferior solution of a coordination problem although it is completely clear that the situation might be improved by changing conventions.[8]

3 Experiments

No experiments have been conducted on the evolution of conventions in the more literal sense of the word. However, among experimentally working social scientists the fact is agreed that conclusions about "evolution of conventions" can be drawn from performing *experiments with coordination games.* Van Huyck, Battalio, and Beil (1990) initiated experiments with symmetric n-person coordination games ("weakest-link games"). During the past decade their experiments have been repeated at various places with varying experimental design (see, for example, Berninghaus and Ehrhart, 1998, 2001, Van Huyck, Battalio, and Beil, 1993). Another type of coordination experiments has been initiated by Cooper, De Jong, Forsythe, and Ross (1992a, 1992b). All these experiments were focused on populations with global interaction, i.e. each agent in the population either played against all other members of the population or at least had a chance to play against another population member who was selected randomly. It was a common feature of the results of all these experiments that risk dominance plays an important role in strategy selection. This was also supported by theoretical investigations on convention selection, in which it was shown that risk dominant equilibrium configurations have the largest basin of attraction (see for example, Berninghaus, 2003, Amir and Berninghaus, 1996).

In this section, we will present experimental results on coordination games that are closely related to our theoretical considerations in the previous section. More precisely, we will present material on experiments on coordination games in populations with a given local interaction structure (see Berninghaus, Ehrhart, and Keser 2002). Again risk dominance turns out to be an important criterion for strategy selection.

3.1 Experimental Results

In this subsection we assume that each player is matched with the members of his reference group to play the bilateral 2×2 coordination game sequentially. We here want to report on some experiments which were conducted at the University of Karlsruhe from 1996-1998. More detailed information

[8] For an illustrative example consider a well known big software company producing operating systems.

on these experiments can be found in Berninghaus, Ehrhart, and Keser (1998, 2002).

These experiments were organized as follows: The basic 2×2 coordination game was characterized by the following payoff table.

	X	Y
X	$(80, 80)$	$(60, 10)$
Y	$(10, 60)$	$(90, 90)$

This game has two Nash-equilibria in pure strategies. Strategy combination (X, X) is the risk-dominant equilibrium and (Y, Y) is the payoff-dominant equilibrium.

We assume that a player who announces a particular strategy (X or Y) at the beginning of a period has to play this strategy sequentially against all members of his reference groups in the 2×2 one-shot coordination game. His per-period payoff is calculated as the average payoff gained by playing against each of his neighbors. In an experimental session, the respective game is played 20 times by the same population of players. After each repetition, each player is informed about the distribution of his neighbors' decisions in the previous round. A player's total payoff in a session is the sum of his period payoffs over all 20 rounds. Players know that their neighbors also interact with other neighbors and that the game ends after 20 rounds. However, they are not explicitly told the particular local interaction structure they live in.

Interaction structures were supposed to be symmetric. More precisely, in the experiments three types of local interaction structures were studied.

- Closed neighborhoods: In a closed neighborhood each player is supposed to interact with each of the other players in the population. Thus, $|N(i)| = n - 1$. For closed neighborhoods only populations of size $n = 3$ were considered. We introduced the closed neighborhood design to imitate global interaction in a small population. We chose the particular population size $n = 3$ to compare the results in populations with global interaction with the results of so-called *nearest neighbor interaction* which in our design is characterized by a circular local interaction structure with neighborhood size $|N(i)| = 2$.

- Local interaction along a circle: n players were assigned around a circle. Each player only interacted with his $|N(i)|$ (local) neighbors on the circle, where $\frac{|N(i)|}{2}$ were located to the right and $\frac{|N(i)|}{2}$ were located to the left. In the experiments with circular local interaction

structure only neighborhood sizes of $|N(i)| = 2$ and $|N(i)| = 4$ were considered.

- Local interaction on a lattice: To play a four-neighbors-game ($|N(i)| = 4$), sixteen players ($n = 16$) were distributed on a lattice. Each player interacted with his four (local) neighbors on the lattice, that is he interacted with his left, right, top and bottom neighbor. The players were assigned as illustrated below. Player 6, for example, had player 5 as his left neighbor, player 7 as his right neighbor, player 2 as his top neighbor and player 10 as his bottom neighbor. Player 1, for example, had player 4 as his left neighbor, player 2 as his right neighbor, player 13 as his top neighbor and player 5 as his bottom neighbor.

Experimental design: Lattice neighborhood structure

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

In this paper, we report on the experimental results of 4 different treatments given in the table below (for more details see Berninghaus, Ehrhart and Keser, 2002).

Four different neighborhood games (treatments)

treatment	$\|N(i)\|$	interaction structure	n	# sessions (players/session)
AVG2GROUP	2	closed	3	2 (12)
AVG2CIRCLE	2	circle	8	4 (16)
AVG4CIRCLE	4	circle	16	8 (16)
AVG4LATTICE	4	lattice	16	8 (16)

Treatments AVG2GROUP and AVG2CIRCLE were designed in a way so that one could test whether subjects in populations with local interaction behave differently from subjects in global interaction populations.

Treatments AVG4CIRCLE and AVG4LATTICE were designed to investigate differences in strategy selection which may be induced by different local interaction structures. By conducting our experiments we wanted to answer the following questions:

1.) Does local interaction really matter?

To answer this question we compare the results of treatments AVG2GROUP and AVG2CIRCLE. We remind the reader that players in AVG2CIRCLE were distributed around a circle with a nearest-neighbor interaction structure. Locally, each player plays sequentially against 2 opponents (members of his reference group). However, these opponents themselves play against other players in their reference groups. That is, globally each player plays against the whole population but only in an indirect way. In the AVG2GROUP treatment each player interacts globally and locally as well as with 2 opponent players. In this treatment there is no indirect strategic information transmission.

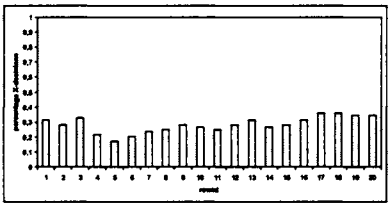


Figure 1: Percentage of X-decisions in AVG2CIRCLE ($n = 8, |N(i)| = 2$)

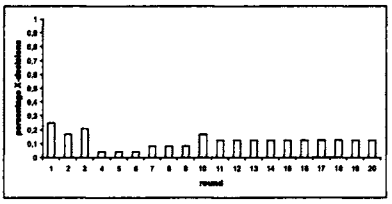


Figure 2: Percentage of X-decisions in AVG2GROUP ($n = 3$)

Comparing the percentage of X-decisions in closed neighborhoods with the percentage of X-decisions in circular neighborhood structures we see that the percentage of X-decisions in the circular neighborhood structure

is higher than in the closed neighborhoods.[9] In treatment AVG2CIRCLE in each round we observe a higher percentage of X-decisions than in treatment AVG2GROUP, although this difference is not statistically significant (two-sided U-test, p-value of 0.194). We conclude that in the long run local interaction matters. Moreover, risk dominance seems to be a more important selection criterion in local interaction populations than in global interaction.[10]

In AVG2GROUP all populations very quickly reached a Y-equilibrium[11], 6 of them stayed there for the rest of the game. One population ended up in an X-equilibrium. In AVG2CIRCLE, 5 populations reached a Y-equilibrium in early rounds, four of them stayed there for the rest of the game. Two of the 3 populations that did not reach a Y-equilibrium reached an X-equilibrium towards the end of the game.

2.) Does local interaction structure matter?

For reference groups with four members, we compare the circle (AVG4-CIRCLE) and the lattice structure (AVG4LATTICE). In both games, subjects were given exactly the same instructions which contained information about the number of neighbors, but not about the specific type of interaction structure.

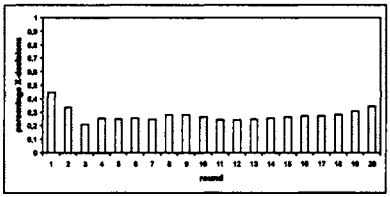


Figure 3: Percentage of X-decisions in AVG4CIRCLE ($n = 16, |N(i)| = 4$)

Figures 3 and 4 show the percentage of X-decisions over all 20 rounds

[9]The χ^2-test shows that in the first round the percentage of X-decisions in the open neighborhoods is not significantly different from the percentage of X-decisions in the closed neighborhoods (at a significance level of 5%).

[10]In Berninghaus, Ehrhart, and Keser (1998) similar experiments were conducted with different payoff functions. In these experiments, the per-period payoff of a player was defined as the *minimum payoff* a player could get from the opponents in his reference group. The experimental results under this particular design showed a significantly higher role of risk dominance in strategy selection than in AVG2CIRCLE.

[11]A Y- resp. X-equilibrium is a strategy configuration in which all players of the population choose Y resp. X.

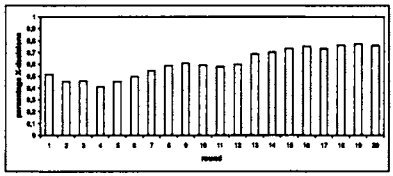


Figure 4: Percentage of X-decisions in AVG4LATTICE ($n = 16, |N(i)| = 4$)

in both treatments. In all rounds in game AVG4LATTICE, we observe a higher percentage of X-decisions than in game AVG4CIRCLE. In the first round, the difference is not statistically significant (χ^2-test). However, we observe a significant difference over all 20 rounds (two-sided U-test). We conclude that, everything else being equal, it matters whether players are distributed around a circle or on a lattice.

In game AVG4CIRCLE, four populations reached a Y-equilibrium, and 1 population reached an X-equilibrium at the very end of the game. In game AVG4LATTICE, only one population reached a Y-equilibrium, but 5 populations reached an X-equilibrium. Three of the 5 populations that reached an X-equilibrium stayed there until the end of the game. No population ever reached an equilibrium in which some players choose X and others choose Y.

Why was there a significant difference between the results of treatment AVG4CIRCLE and treatment AVG4LATTICE although the subjects were confronted with exactly the same experimental instructions? Our theoretical results in the previous section which are based on local best-reply behavior do not give an explanation of this phenomenon either. Instead of trying to find a theoretical ad-hoc model which may imply a satisfactory explanation of our experimental findings we study three different logit models as a pre-study for a theoretical model of behavior. These models were examined for the first time in Berninghaus, Ehrhart, and Keser (2002). A brief summary of these results is given below.

3.2 Logit Regressions

The first model is based on the simple assumption of local best reply to the observed distribution of one's neighbors' choices in the previous round. The second model additionally takes into consideration the player's own previous choice. In the third model, the average number of observed changes in the distribution of the neighbors' choices is incorporated. Although we can observe that the subjects' average response behavior usually changes

during the duration of the decision process, for the sake of simplicity, we consider only one decision rule for all periods. Let π denote the probability of choosing X, and $(1-\pi)$ the probability of choosing Y. According to the logit hypothesis the probability of choosing X is given by

$$\pi = F(x'\beta) = \frac{1}{1+e^{-x'\beta}},$$

where x denotes the vector of explanatory variables and β the vector of coefficients. The maximum likelihood estimates of all coefficients are presented in Table 2 below. For each game and each model the overall-F-test and the t-test for each estimator is calculated. Each test shows a significant result. Also, the calculated multiple correlation coefficients show high values, which increase from model 1 up to model 3. We infer from this that the data are well organized by the non-linear regression models and that the data are better described by a model with a higher number than by a model with a smaller number of parameters.

Model 1: $$\boxed{ln\left(\frac{\pi}{(1-\pi)}\right) = x'\beta = \beta_0 + \beta_1 x_1,}$$

where x_1 denotes the number of neighbors who selected X in the previous round.

The estimators of β_1 turned out to be significantly positive. That is, the probability for choosing X increases with the number of neighbors choosing X in the previous period. The estimated probabilities for choosing strategy X contingent on the distribution of the neighbors' choices are given later on.

Model 2: $$\boxed{ln\left(\frac{\pi}{(1-\pi)}\right) = x'\beta = \beta_0 + \beta_1 x_1 + \beta_2 x_2,}$$

where x_1 denotes the number of neighbors who selected X in the previous period, and x_2 is a variable equal to 1 if the player's own choice was X in the previous round and equal to 0 if it was Y.

The estimators of β_1 and β_2 were significantly positive. As in model 1, we conclude that the probability for choosing X increases with the number of neighbors choosing X in the previous period as well as if the player's choice was X in the previous period.

Model 3: $$\boxed{ln\left(\frac{\pi}{(1-\pi)}\right) = x'\beta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3,}$$

where x_1 again denotes the number of neighbors who chose X in the previous round, x_2 is a variable with value 1 if the player's own choice was X in the previous round and with value 0 if it was Y, and x_3 denotes the average frequency of observed changes in the distribution of the neighbors' choices over all previous rounds.

Model 3 was estimated only for treatments AVG4CIRCLE and AVG4-LATTICE. Subjects in these treatments switch strategies very often. The estimators of β_1, β_2, and β_3 turned out to be significantly positive. From the significant positive estimate of β_3 in both games we conclude that subjects tend to choose the risk-dominant X rather than the more risky but payoff-superior alternative Y if they have observed a lot of changes in their neighbors' choices in the previous periods.

By this observation we may explain the higher percentage of X-choices on the lattice than around the circle. In treatment AVG4LATTICE we observe more individual decision changes than in treatment AVG4CIRCLE. The frequencies of decision changes were 10.9 percent in AVG4CIRCLE and 14.1 percent in AVG4LATTICE.[12] Also, the frequencies of changes in the distribution of the neighbors' decisions that an individual player observes during the course of play differ significantly (difference test): 28.2 percent in AVG4CIRCLE and 38.3 Percent in AVG4LATTICE. This difference together with the observation, that the subjects tend to react on changes in their neighbors' choices by choosing X (see model 3), may give us an interesting explanation for the higher frequency of X-choices on the lattice than around the circle.

This result, however, raises another question: Although subjects had the same instruction sheets in the two treatments, and according to Table 2 used (approximately) the same decision rule, why do they behave differently? Why do subjects change their decisions more often when they are arranged on a lattice rather than on a circle? Obviously, this difference in behavior has to be traced back to the difference in interaction structures.

As a first approach, simulations were run (see Berninghaus, Ehrhart, and Keser, 2002) with the probabilistic history-dependent decision rules derived from model 3. For both games we use the same set of parameters derived from the estimated parameters of model 3 shown in Table 2. The initial probability to choose X was supposed to be 50 percent, which was approximately equal to the observed frequency of X-choices in both treatments. With each interaction structure one million 20-period-simulations were calculated. The results of the simulations are shown in Table 1. With the results of these simulations one can reproduce the ob-

[12]The difference is significant (difference test).

served differences in experiments. In AVG4CIRCLE the average frequency of decision changes is 12.1 percent and in game AVG4LATTICE 18.0 percent. Although these values are a little bit higher than the observed values 10.9 and 14.1 percent, the simulations reproduced higher values for the game AVG4LATTICE than for the game AVG4CIRCLE. The same conclusion holds for the frequencies of the observed changes in the distribution of the neighbors' decisions. The simulations reproduced higher values for AVG4LATTICE than for AVG4CIRCLE: 38.3 and 28.2 percent in the experiment and 41.1 and 25.1 percent in the simulations. Because we use the same individual decision rule, these differences must be caused by the different local interaction structures.

Table 1: Simulations: Conditions and results

CONDITIONS	
Treatments	AVG4CIRCLE and AVG4LATTICE
Decision rule	Model 3 with $\beta_0 = -4.0, \beta_1 = 1.5, \beta_2 = 2.0, \beta_3 = 0.7$
Initial decision	Prob. for X in the first round = 50%
Number of simulations	1 mill. 20-period simulations with each treatment

RESULTS				
Treatments	AVG4CIRCLE		AVG4LATTICE	
	Simulation	Experiment	Simulation	Experiment
Frequency of decision changes	12.1	10.9	18.0	14.1
Frequency of changes in the distribution of neighbors' decisions	25.1	28.2	41.1	38.3
Percentage of X-decisions	50.7	34.1	64.8	61.1

The simulations exactly reproduced what we observed in the experiments: The average number of X-decisions in a local interaction design on a lattice was higher than in circular interactions. From this we conclude, that the observed differences in strategy choice between AVG4CIRCLE and AVG4LATTICE are generated exclusively by the particular interaction structure.

R: Multiple correlation coefficient
DF: Degrees of freedom
F-Value: Value of the F-statistic of the overall-F-test[13]
β_k : Estimator of coefficient β_k
t-value: Value of the t-statistic

Summarizing, experimental evidence drastically demonstrates that local

[13]Goodness of fit test.

Table 2: Logit regressions for model 3

Treatment	R	DF	β_0	β_1	β_2	β_3
		F-Value	(t-value)	(t-value)	(t-value)	(t-value)
AVG4CIRCLE	0.847	2428	-4.42	1.62	1.98	0.78
		2053.2	(-21.2)	(20.5)	(18.0)	(3.5)
AVG4LATTICE	0.796	2428	-3.43	1.11	2.17	0.65
		1397.7	(-19.9)	(20.6)	(20.8)	(3.3)

interaction structures imposed on a population have a strong influence on equilibrium selection in coordination games. Again risk-dominance is an important criterion for strategy selection in coordination games. Imposing different local interaction structures on a population may specify these results further.

Although the experimental results of this section have shed some new light on the coordination problem in populations we are far from being able to give satisfactory explanations for these results. We explained the tendency of players in AVG4LATTICE to prefer the risk-dominant equilibrium by the observed behavior that players in this treatment switched strategies more often in each period than in treatment AVG4CIRCLE. But why do players switch strategies more often in a circular interaction structure than in a lattice interaction structure? Much more theoretical and experimental research needs to be done before we can give a satisfactory answer to such questions.

4 Conclusions

We regard our paper as a first step in explaining the evolution of conventions in a local interaction framework. Naturally, our model still has to be generalized into various directions. Concerning the theoretical part of our paper, one could first extend the model of matching players from simple 2-person to more general n-person basic games. Even asymmetric basic games could be considered. Second, we saw that a general model of dynamic strategy adaptation is still missing. Our results still depend on the assumption of an odd or even population size or on particular assumptions of the numerical proportions of the players' payoffs. Third, one could consider more general local interaction structures. Most researchers in this field used the circular or lattice interaction structures. The paper by Ostmann and Saboya (1999) indicates an interesting way of how to escape from these simplified models. Finally, the myopic (local) best-reply assumption

on individual behavior could be substituted by other behavioral rules, as for example, imitation or a more sophisticated type of behavior which has been empirically supported by our experiments.[14]

Concerning the experimental part of our paper we think of the following generalizations. First, one should get rid of the simple 2×2 coordination games. Sometimes subjects in our experiments seemed to be bored playing such simple games for many rounds. Basic coordination games with asymmetric Nash-equilibria may be a first step. Maybe one should even extend the concept of a convention to a more general framework. For example, choosing a particular bidding rule in an auction may constitute a convention in a population. Experiments with such basic games may be difficult to design. Maybe one could take account of the experience that has already been gained in auction experiments. Second, the assumption of fixed interaction structures in experiments should be relaxed. Subjects in each period should have to choose not only a particular strategy in a basic coordination game but also a reference group in which they want to play for the next periods.[15] As a preliminary study to this topic one could consider experiments on pure "network formation". In this type of experiments subjects can open or sever connections with other members of the population. By opening connections, subjects collect profits which may be reduced by connection costs players have to pay for opening and/or maintaining a connection. In the experiments, one is interested which type of networks will result when subjects are allowed to choose their neighbors. Recent experimental results show (see for example, Plott and Callander, 2005, Falk and Kosfeld, 2004, Berninghaus, Ehrhart, Ott and Vogt, 2004) that strict Nash-networks are focal points for network formation. Based on the experimental design which has proven to be appropriate in pure network formation experiments we plan to extend these experiments to incorporate strategy selection in coordination games when matched pair-wise with their partners chosen in the network formation game.

Finally, one should question whether the experimental design described in section 3 is appropriate to capture the idea of evolution. Either one should repeat the experiments for many more periods (than 20 rounds) or one should apply the method of "(almost) continuous time experimentation" which was developed in Berninghaus, Ehrhart and Keser (1999) or in Berninghaus and Ehrhart (2003). In an (almost) continuous time design subjects seem to experiment with strategies and switch strategies much more often than in discrete time experimentation. In some way this seems

[14] Some more detailed ideas on this type may be found in Berninghaus and Schwalbe (1996b).

[15] Experimental work on this topic has been done by Corbae and Duffie (2004).

to mimick the evolutionary adaptation of strategy choices in a population of players. Moreover, in the (almost) continuous time experiments subjects are allowed to take an action at any time during a given time interval, which essentially implies that subjects decide sequentially. When deciding about the neighbors **and** strategy choice in a 2-person coordination game it may be easier for a subject to solve this decision problem in an (almost) continuous time design.

References

[1] Amir M, Berninghaus S (1996) Another approach to learning and mutation in games. Games and Economic Behavior 14:19–43

[2] Berninghaus S (2003) The evolution of conventions-some theoretical and experimental aspects. Jahrbuch für Neue politische Ökonomie 21. In: Holler MJ et al. (eds.) The Political Economy of Institutional Evolution, Mohr Siebeck, Tübingen, pp. 101–132

[3] Berninghaus SK, Ehrhart KM (2001) Information and efficiency in coordination games: Recent experimental results. In: Debreu G, Neuefeind W, Trockel W (eds.) Economics Essays. Springer, Heidelberg, New York, pp. 19–39

[4] Berninghaus SK, Ehrhart KM (2003) The power of ESS: An experimental study. Journal of Evolutionary Economics 13:161–181

[5] Berninghaus SK, Ehrhart KM, Keser C (2002) Conventions and local interaction structures. Games and Economic Behavior 29:177–205

[6] Berninghaus SK, Ehrhart KM, Keser C (1999) Continuous-time strategy selection in linear population games. Experimental Economics 2:41–57

[7] Berninghaus SK, Ehrhart KM, Keser C (1998) Coordination and local interaction: Experimental evidence. Economics Letters 58:269–275

[8] Berninghaus SK, Ehrhart KM (1998) Time horizon and equilibrium selection in tacit coordination games: Experimental results. Journal of Economic Behavior and Organization 37:231–248

[9] Berninghaus S, Ehrhart KM, Ott M, Vogt B (2004) Searching for "stars" – recent experimental results on network formation. SFB 504 Working Paper 04-34, University of Mannheim

[10] Berninghaus SK, Schwalbe U (1996a) Conventions, local interaction, and automata networks. Journal of Evolutionary Economics 6:297–312

[11] Berninghaus SK, Schwalbe U (1996b) Evolution, interaction, and Nash-equilibria. Journal of Economic Behavior and Organization 29:57–85

[12] Callander S, Plott CR (2005) Principles of network development and evolution: an experimental study. To appear in: Journal of Public Economics

[13] Cooper RW, De Jong DV, Forsythe R, Ross TW (1992a) Forward induction in coordination games. Economics Letters 40:167–172

[14] Cooper RW, De Jong DV, Forsythe R, Ross TW (1992b) Communication in coordination games. Quarterly Journal of Economics 107:739–771

[15] Corbae D, Duffy J (2004) Experiments with network formation. Working Paper

[16] Dieckmann T (1999) The evolution of conventions with mobile players. Journal of Economic Behavior and Organization 38:93–111

[17] Ellison G (1993) Learning, local interaction, and coordination. Econometrica 61:1047–1071

[18] Ely J (1995) Local conventions. Mimeo, University of California, Berkeley

[19] Falk A, Kosfeld M (2003) Its all about connections: Evidence on network formation. IEW Working Paper 146, Universität Zürich

[20] Fogelman F, Goles E, Weisbuch G (1983) Transient length in sequential iterations of threshold functions. Disc. Appl. Math. 6:95–98

[21] Goles E (1985) Dynamical behavior of neural networks. SIAM J. Alg. Disc. Meth. 6:749–754

[22] Harsanyi JC, Selten R (1988) A General Theory of Equilibrium Selection in Games. MIT Press, Cambridge MA, London

[23] Hopper RH (1982) Left-right: Why driving rules differ. Transportation Quarterly 36:541–548

[24] Kandori M, Mailath GJ, Rob R (1993) Learning, mutation, and long run equilibria in games. Econometrica 61:29–56

[25] Lewis DK (1969) Convention - A Philosophical Study. Harvard University Press, Cambridge MA

[26] Ostmann A, Saboya M (1999) Symmetric homogeneous local interaction. Institute of Mathematical Economics, Working paper No. 314

[27] Robert F (1986) Discrete Iterations: A Metric Study. Springer-Verlag, New York

[28] Van Huyck J, Battalio RC, Beil RO (1990) Tacit coordination games, strategic uncertainty and coordination failure. American Economic Review 80:234–248

[29] Van Huyck J, Battalio RC, Beil RO (1993) Asset markets as an equilibrium selection mechanism: Coordination failure, game form auctions, and tacit communication. Games and Economic Behavior 5:485–504

[30] Young HP (1993) The evolution of conventions. Econometrica 61:57–84

Empirical Examination of Operational Loss Distributions

Svetlozar T. Rachev

Institut für Statistik und Mathematische Wirtschaftstheorie
Universität Karlsruhe

and

Department of Statistics and Applied Probability
UC Santa Barbara

Anna Chernobai

Department of Statistics and Applied Probability
UC Santa Barbara

Christian Menn

School of Operations Research and Industrial Engineering
Cornell University

1 Introduction

Until very recently, it has been believed that banks are exposed to two main types of risks: credit risk (the counterparty failure risk) and market risk (the risk of loss due to changes in market indicators, such as interest rates and exchange rates), in the order of importance. The remaining financial risks have been put in the category of *other* risks, operational risk being one of them. Recent developments in the financial industry have shown that the importance of operational risk has been largely under-estimated. Newly defined capital requirements set by the Basel Committee for Banking Supervision in 2004, require financial institutions to estimate the capital charge to cover their operational losses [6].

This paper is organized as follows. In Section 2 we give the definition of operational risk and describe the effect of the recent developments in the global financial industry on banks' exposures to operational risk. The following section, Section 3, will briefly outline the recent requirements set by the Basel Committee regarding the capital charge for operational

risk. After that we proceed to Section 4 that presents several alternative models that can be used for operational risk modeling. In Section 5 the class of heavy-tailed αStable distributions and their extensions are defined and reviewed. Section 6 consists of the empirical analysis with real operational risk data. Finally, Section 7 summarizes the findings and discusses directions for future work.

2 Definition of Operational Risk in Finance

Operational risk has been recently defined as 'the risk of loss resulting from inadequate or failed internal processes, people and systems or from external events' [4]. Examples include losses resulting from deliberate or accidental accounting errors, equipment failures, credit card fraud, tax non-compliance, unauthorized trading activities, business disruptions due to natural disasters and vandalism. Operational risk affects the soundness and efficiency of all banking activities.

Until recently, the importance of operational risk has been highly underestimated by the banking industry. The losses due to operational risk has been largely viewed as unsubstantial in magnitude, with a minor impact on the banking decision-making and capital allocation. However, increased investors' appetites have led to significant changes in the global financial industry during the last couple of decades - globalization and deregulation, accelerated technological innovation and revolutionary advances in the information network, and increase in the scope of financial services and products. These have caused significant changes in banks' risk profiles, making banks more vulnerable to various sources of risk. These changes have also brought the operational risk to the center of attention of financial regulators and practitioners.

A number of large-scale (exceeding $1 billion in value) operational losses, involving high-profile financial institutions, have shaken the global financial industry in the past two decades: BCCI (1991), Orange Country (1994), Barings Bank (1995), Daiwa Bank (1995), NatWest (1997), Allied Irish Banks (2002), the Enron scandal (2004), among others.

3 Capital Requirements for Operational Risk

The Basel Committee for Banking Supervision (BCBS) has brought into focus the importance of operational risk in 1998 [2], and since 2001 bank regulators have been working on developing capital-based counter-measures to protect the global banking industry against the risk of operational losses

- that have demonstrated to possess a substantial, and at times vital, danger to banks. It has been agreed to include operational risk into the scope of financial risks for which the regulatory capital charge should be set [3]. Currently in progress is the process of developing models for the quantitative assessment of operational risk, to be used for measuring the capital charge.

The Basel Capital Accord (Basel II) has been finalized in June 2004 [6]. It explains the guidelines for financial institutions regarding the capital requirements for credit, market and operational risks (Pillar I), the framework for the supervisory capital assessment scheme (Pillar II), and the market discipline principles (Pillar III). Under the first Pillar, several approaches have been proposed for the estimation of the regulatory capital charge. Bank is allowed to adopt one of the approaches, dependent upon fulfillment of a number of quantitative and qualitative requirements. The Basic Indicator Approach (that takes the capital charge to be a fixed fraction of the bank's gross income) and the Standardized Approach (under which the capital charge is a sum of fixed proportions of the gross incomes across all business lines) are the 'top-down' approaches, since the capital charge is determined 'from above' by the regulators; the Advanced Measurement Approaches (that involve the exact form of the loss data distributions) are the 'bottom-up' approaches, since the capital charge is determined 'from below', being driven by individual bank's internal loss data history and practices.

3.1 Loss Distribution Approach

The Loss Distribution Approach (LDA) is one of the proposed Advanced Measurement Approaches. It makes use of the exact operational loss frequency and severity distributions. A necessary requirement for banks to adopt this approach is an extensive internal database.

In the LDA, all bank's activities are classified into a matrix of 'business lines/event type' combinations. Then, for each combination, using the internal loss data the bank estimates two distributions: (1) the loss frequency and (2) severity. Based on these two estimated distributions, the bank computes the probability distribution function of the cumulative operational loss. The operational capital charge is computed as the simple sum of the one-year Value-at-Risk (VaR) measure (with confidence level such as 99.9%) for each 'business line/ event type' pair. The capital charge

for a general case (8 business lines and 7 event types) can be expressed as:[1]

$$\mathrm{K_{LDA}} = \sum_{j=1}^{8} \sum_{k=1}^{7} \mathrm{VaR}_{jk}. \tag{3.1}$$

where $\mathrm{K_{LDA}}$ is the one-year capital charge under the LDA, and VaR is the Value-at-Risk risk measure,[2] for a one-year holding period and high confidence level (such as 99.9%), based on the aggregated loss distribution, for each 'business line/event type' jk combination.

4 Aggregated Stochastic Models for Operational Risk

Following the guidelines of the Basel Committee, the aggregated operational loss process can be modeled by a *random sum* model, in which the summands are composed of random amounts, and the number of such summands is also a random process. The compound loss process is hence assumed to follow a stochastic process $\{S_t\}_{t\geq 0}$ expressed by the following equation:

$$S_t = \sum_{k=0}^{N_t} X_k, \quad X_k \overset{\text{iid}}{\sim} F_\gamma, \tag{4.1}$$

in which the loss magnitudes (severities) are described by the random independently and identically distributed (iid) sequence $\{X_k\}$ assumed to follow the distribution function (cdf) F_γ that belong to a parametric family of continuous probability distributions, and the density f_γ, and the counting process N_t is assumed to follow a discrete counting process. To avoid the possibility of negative losses, it is natural to restrict the support of the severity distribution to the positive half-line $\mathbb{R}_{>0}$. Representation (4.1) generally assumes (and we also use this assumption) independence between the frequency and severity distributions. The cdf of the compound

[1] Such representation *perfect correlation* between different 'business lines/ event type' combinations. Ignoring possible dependence structures within the banking business lines' and event type profiles may result in overestimation of the capital charge under the LDA approach. The latest Basel II proposal suggested to take into account possible dependencies in the model [6]. Relevant models would involve using techniques such as copulas (see for example numerous works by McNeil and Embrechts on the discussion of copulas), but this is outside the scope of this paper.

[2] $\mathrm{VaR}_{\Delta t,\ 1-\alpha}$ is the risk measure that determines the highest amount of loss that one can expect to lose over a pre-determined time interval (or holding period) Δt at a pre-specified confidence level $1-\alpha$. Detailed analysis of VaR models can be found in [11].

process is given by:

$$P(S_t \leq s) = \begin{cases} \sum_{n=1}^{\infty} P(N_t = n)\ F_\gamma^{n*}(s) & s > 0 \\ P(N_t = 0) & s = 0, \end{cases} \tag{4.2}$$

where F_γ^{n*} denotes the n-fold convolution with itself.

We summarize the basic properties of such compound process by the expressions for the mean and variance:[3]

$$\begin{aligned} &\text{Mean:} \quad \mathbb{E}S_t = \mathbb{E}N_t \cdot \mathbb{E}X, \\ &\text{Variance:} \quad \mathbb{V}S_t = \mathbb{E}N_t \cdot \mathbb{V}X + \mathbb{V}N_t \cdot (\mathbb{E}X)^2. \end{aligned} \tag{4.3}$$

The upper (right) tail behavior has a simple expression for the special case when X belongs to the class of sub-exponential distributions, $X \sim \mathcal{S}$, such as Lognormal, Pareto or the heavy-tailed Weibull. Then the upper tail of the compound process can asymptotically be approximated by (see e.g. [10]):

$$\text{Right tail:} \quad P(S_t > s) \propto \mathbb{E}N_t \cdot P(X > s), \quad s \to \infty. \tag{4.4}$$

4.1 Compound Homogeneous Poisson Process

A stochastic process of the form (4.1) and N_t following a Poisson process with a fixed intensity lambda (λ) is called a *compound homogeneous Poisson process.* It assumes a fixed intensity of the number of loss events in a unit of time. Incorporating the probability mass function of a Poisson distribution into the basic model of Equation (4.2), the cdf of the compound process becomes:

$$P(S_t \leq s) = \begin{cases} \sum_{n=1}^{\infty} \frac{(\lambda t)^n e^{-\lambda t}}{n!}\ F_\gamma^{n*}(s) & s > 0 \\ e^{-\lambda t} & s = 0. \end{cases} \tag{4.5}$$

The basic properties of a compound Poisson process can be summarized using Equations (4.3) and (4.4) as follows:

$$\begin{aligned} &\text{Mean:} \quad \mathbb{E}S_t = \lambda t \cdot \mathbb{E}X, \\ &\text{Variance:} \quad \mathbb{V}S_t = \lambda t \cdot \mathbb{V}X + \lambda t \cdot (\mathbb{E}X)^2, \\ &\text{Right tail:} \quad P(S_t > s) \propto \lambda t \cdot P(X > s), \quad s \to \infty, \quad X \sim \mathcal{S}. \end{aligned} \tag{4.6}$$

[3] Of course, this requires the existence of the first and second raw moments of the loss severity distribution.

The mean of the Poisson distribution is the parameter lambda, which is estimated via Maximum Likelihood as the simple arithmetic mean number of events in a unit of time. A number of tests exist to test the Poisson assumption. A common formal test is the Chi-square test. If the model is rejected, then one should consider a more complex alternative model. Next, we briefly review some of them.

4.2 Compound Cox Process

The compound homogeneous Poisson process, discussed earlier, is based on the counting process that is characterized by a fixed intensity lambda. We now relax this assumption. In real life sometimes there are reasons to believe that flow of loss events is often chaotic in nature, and occurrence at each fixed time interval is inhomogeneous and not easily predictable. The compound Cox process, also known in literature as the *doubly stochastic compound Poisson process*, involves a non-constant (or non-homogeneous) form of the intensity factor of the Poisson component of the model. The associated Poisson process is said to be controlled by the random measure $\Lambda(t) = \int_0^t \lambda(s)ds$. A number of scenarios can be considered.

Example 1. The intensity is a random variable, that follows a distribution function, discrete or continuous: $\lambda(t) \sim \mathcal{L}$. For example, λ may take two values λ_1 and λ_2 with probabilities α and $1-\alpha$, respectively. Another example is a Poisson process with intensity λ that follows a two-parameter Gamma distribution. Such counting model is known as a Negative-Binomial model. Such counting models are often called *mixed Poisson models*. The basic properties of compound mixed Poisson processes are dependent on the distribution of the underlying intensity process. Let μ_λ and σ_λ^2 denote the expectation and variance of λ. Then

$$\begin{aligned} &\text{Mean:} && \mathbb{E}S_t = \mu_\lambda t \cdot \mathbb{E}X, \\ &\text{Variance:} && \mathbb{V}S_t = \mu_\lambda t \cdot \mathbb{V}X + (\mu_\lambda t + t^2\sigma_\lambda^2) \cdot (\mathbb{E}X)^2, \\ &\text{Right tail:} && P(S_t > s) \propto \mu_\lambda t \cdot P(X > s), \quad s \to \infty, \quad X \sim \mathcal{S}. \end{aligned} \tag{4.7}$$

Example 2. The intensity is of form $\lambda(t)$ and is dependent on time. The associated cumulative intensity is of form $\Lambda(t)$, a positive non-decreasing process. Here, one example would be a deterministic process that fits the number of losses per unit of time, examined over a prolonged time interval. Another scenario would incorporate a random component into the model.

Here, Brownian Motion and other stochastic models can be of use. Given a particular value of the intensity, the conditional compound Cox process coincides with the compound homogeneous Poisson Process and preserves the properties.

4.3 Renewal Process

Another approach to aggregate losses occurring at random times would be to consider looking at the inter-arrival times, instead of the number of arrivals, in a fixed time interval. Such models are called the *renewal models.* A Poisson counting process implies that the inter-arrival times between the loss events are distributed as an Exponential random variable with mean $1/\lambda$. This assumption on the distribution of the inter-arrival times can be relaxed, and a wider class of distributions can be fitted to the loss inter-arrival times data.

An excellent reference on random sum models and applications to financial data is [1].

4.4 Aggregated Model for Left-Truncated Loss Data

In the operational loss modeling, one should be careful to possible rules that banks follow in recording their internal loss data into the databases. For example, a reality that is often neglected in practical models, is that banks record the losses beginning from a minimum collection threshold of \$5,000-\$10,000. In the external databases the threshold is even higher - \$1,000,000. Hence, the recorded (and observed) loss data is *left-truncated*, and the associated frequency is below the complete-data frequency. This has direct implication on the model specification. Correctly specifying the model for the truncated data, we arrive at the following expressions of the loss severity and frequency probability density/mass functions (pdf/pmf) (assuming, for simplicity, a simple homogeneous Poisson counting process):

$$\begin{aligned} &\text{Severity pdf:} \quad f_\gamma(x \mid x > u) = \begin{cases} \frac{f_\gamma(x)}{\bar{F}_\gamma(u)} & \text{if } x > u \\ 0 & \text{if } x \le u, \end{cases} \\ &\text{Frequency pmf:} \quad P(N_t = n) = \frac{(\tilde{\lambda}t)^n e^{-\tilde{\lambda}t}}{n!}, \end{aligned} \tag{4.8}$$

where u is the threshold level, $\bar{F}_\gamma(u) = 1 - F_\gamma(u)$, λ is the complete-data frequency parameter, $\tilde{\lambda} = \lambda \cdot \bar{F}_\gamma(u)$ is the truncated data frequency parameter, and N_t is the counting process for the number of losses exceeding u.

In application to operational risk, the truncated compound Poisson model has been introduced and studied in [8]. Further studies include [7], [12].

5 Pareto αStable Distributions

A wide class of distributions that appear highly suitable for modeling operational losses is the class of αStable (or Pareto Stable) distributions. Although no closed-form density form (in the general case) poses difficulties in the estimations, αStable distributions possess a number of attractive features that make them relevant in applications to a variety of financial models. An excellent reference on αStable is [14]. A profound discussion of applications to the financial data can be found in [13]. We now review the definition and basic properties.

5.1 Definition of an αStable Random Variable

A random variable X is said to follow an αStable distribution – we use the notation $X \sim S_\alpha(\sigma, \beta, \mu)$ – if for any $n \geq 2$, there exist $C_n > 0$ and $D_n \in \mathbb{R}$ such that

$$\sum_{k=1}^{n} X_k \stackrel{d}{=} C_n X + D_n, \tag{5.1}$$

where X_k, $k = 1, 2, ..., n$ are iid copies of X. The stability property is governed by the constant $C_n = n^{1/\alpha}$, $0 < \alpha \leq 2$. The stability property is a useful and convenient property, and dictates that the distributional form of the variable is preserved under affine transformations. $\alpha = 2$ corresponds to the Gaussian case. $0 < \alpha < 2$ refers to the non-Gaussian case. When we refer to a $S_\alpha(\sigma, \beta, \mu)$ distribution, we mean the latter case. References on the $S_\alpha(\sigma, \beta, \mu)$ distributions and properties include [14], [13].

5.2 Key characteristics of an αStable Random Variable

For the $S_\alpha(\sigma, \beta, \mu)$ random variables, the closed form density exists only for Gaussian ($\alpha = 2$), Cauchy ($\alpha = 1, \beta = 0$) and Lévy ($\alpha = 1/2, \beta = \pm 1$) densities. For the general case, the distribution is expressed by its characteristic function that takes the form:

$$\mathbb{E}e^{iuX} = \begin{cases} \exp\left(-|\sigma u|^\alpha (1 - i\beta(\operatorname{sign} u) \tan\frac{\pi\alpha}{2}) + i\mu u\right), & \alpha \neq 1 \\ \exp\left(-\sigma|u|(1 + i\beta\frac{2}{\pi}(\operatorname{sign} u) \ln|u|) + i\mu u\right), & \alpha = 1 \end{cases} \tag{5.2}$$

The four parameters[4] defining the $S_\alpha(\sigma, \beta, \mu)$ distribution are:

α, the index of stability: $\alpha \in (0,\ 2)$;

β, the skewness parameter: $\beta \in [-1,\ 1]$;

σ, the scale parameter: $\sigma \in \mathbb{R}_+$;

μ, the location parameter: $\mu \in \mathbb{R}$.

Because of the four parameters, the distribution is highly flexible and suitable for fitting to the data which is non-symmetric (skewed) and possesses a high peak (kurtosis) and heavy tails. The heaviness of tails is driven by the *power tail decay* property (see next).

We briefly present the basic properties of the $S_\alpha(\sigma, \beta, \mu)$ distribution. Let $X \sim S_\alpha(\sigma, \beta, \mu)$ with $\alpha < 2$, then

$$\begin{aligned} &\text{Mean:} && \mathbb{E}X = \begin{cases} \mu & \text{if } \alpha > 1 \\ \infty & \text{otherwise,} \end{cases} \\ &\text{Variance:} && \mathbb{V}X = \infty \quad \text{(no second moment)}, \\ &\text{Tail:} && P(|X| > x) \propto \text{const} \cdot x^{-\alpha},\ x \to \infty \text{ (power tail decay).} \end{aligned} \tag{5.3}$$

5.3 Useful Transformations of Pareto Stable Random Variables

For $\alpha > 1$ or $|\beta| < 1$, the support of $S_\alpha(\sigma, \beta, \mu)$ distribution equals the whole real line, and is useful for modeling data that can take negative and positive values. It would be unwise to directly apply this distribution to the operational loss data, because it takes only positive values. We suggest to use the following three transformations of the random variable to which the Stable law can be applied.

5.3.1 Symmetric αStable Random Variable

A random variable X is said to follow the symmetric αStable distribution, i.e. $X \sim S\alpha S$, if the $S_\alpha(\sigma, \beta, \mu)$ distribution is symmetric and centered around zero. Then there are only two parameters that need to be estimated, α and σ, and the remaining two are $\beta, \mu = 0$.

[4]The parametrization of αStable laws is not unique. The presented one has been propagated by Samorodnitsky and Taqqu [14]. An overview of the different approaches can be found in [15].

To apply $S\alpha S$ distribution to the operational loss severity data, one can do a simple transformation to the original data set: $Y = [-X;\ X]$. The resulting random variable Y is then symmetric around zero:

$$\begin{aligned} &f_Y(y) = g(y), \quad g \in S_\alpha(\sigma, 0, 0) \\ &x = |y|,\ \alpha \in (0,2),\ \sigma > 0. \end{aligned} \tag{5.4}$$

5.3.2 log-αStable Random Variable

It is often convenient to work with the natural *log* transformation of the original data. A typical example is the Lognormal distribution: if X follows a Lognormal distribution, then $\log X$ follows a Normal distribution with the same location and scale parameters μ and σ. The same procedure can be applied here. A random variable X is said to follow a log-αStable distribution, i.e. $X \sim \log S_\alpha(\sigma, \beta, \mu)$, if

$$\begin{aligned} &f_X(x) = \tfrac{g(\ln x)}{x}, \quad g \in S_\alpha(\sigma, \beta, \mu) \\ &x > 0,\ \alpha \in (0,2),\ \beta \in [-1,1],\ \sigma, \mu > 0. \end{aligned} \tag{5.5}$$

Fitting $\log S_\alpha(\sigma, \beta, \mu)$ distribution to data is appropriate when there is reason to believe that the data is very heavy-tailed, and the regular $S_\alpha(\sigma, \beta, \mu)$ distribution may not capture the heaviness in the tails.

5.3.3 Truncated αStable Random Variable

Another scenario would involve a restriction on the density, rather than a transformation of the original data set. The support of the $S_\alpha(\sigma, \beta, \mu)$ distribution can be restricted to the positive half-line, and the estimation part would involve fitting a truncated Stable distribution of the form:

$$\begin{aligned} &f_X(x) = \tfrac{g(x)}{1 - G(0)} \times \mathbb{I}_{x>0} \\ &\text{where} \quad \mathbb{I}_{x>0} := \begin{cases} 1 & \text{if} \quad x > 0 \\ 0 & \text{if} \quad x \leq 0, \end{cases} \end{aligned} \tag{5.6}$$

where $g(x) \in S_\alpha(\sigma, \beta, \mu)$, and $G(0)$ denotes the cdf of the $S_\alpha(\sigma, \beta, \mu)$ distribution at zero. Fitting the left-truncated Stable distribution to the data means fitting the right tail of the distribution.

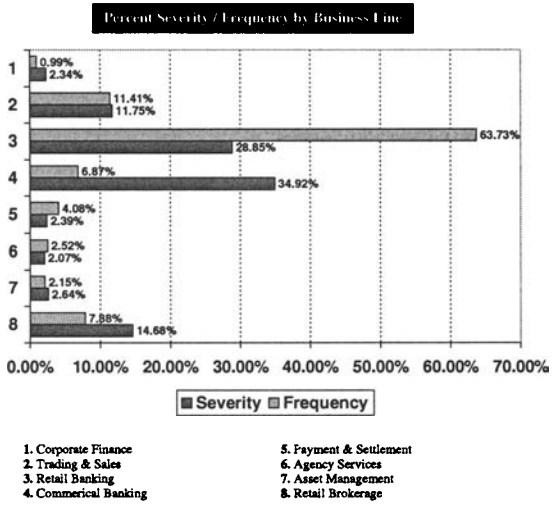
6 Empirical Examination of Operational Loss Data

In the previous sections we discussed the models that can be used to model operational losses. Choosing the right methodology is crucial for accurately estimating the operational risk regulatory capital charge. In addition, understanding the structure of the underlying model that drives the process of the occurrences and severity of the losses is vital for the sound risk management practices and control. In this section we examine the operational loss data and derive conclusions regarding the loss distributions that would be appropriate for modeling the data. We will in particular focus on the loss severity data.

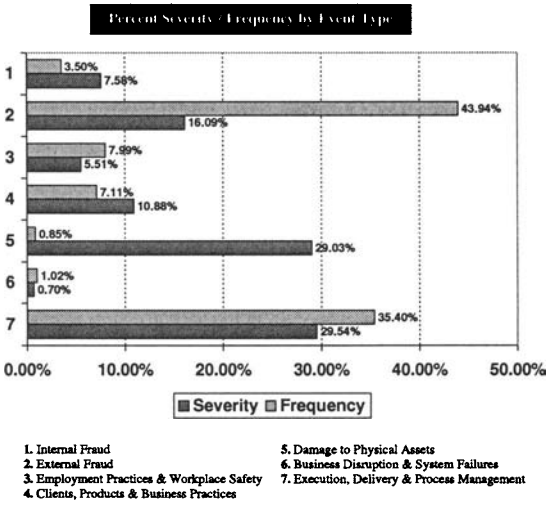
6.1 Results of Basel Committee Loss Data Collection Exercise

In 2002 the Risk Management Group of the BCBS carried out an Operational Loss Data Collection Excercise (OLDC) (also called the third Quantitative Impact Study (QIS3)) aimed at examining various aspects of banks' internal operational loss data [5]. Banks activities were broadly divided into eight business lines and seven loss event types (see Figure 1 for the full list of business lines and event types). Figure 1 demonstrates the severity of losses (i.e. loss amount) and the frequency of losses (i.e. number of losses) by business lines and event types, as a percentage of total. The results are based on the one year of loss data (collected in 2001) provided by 89 participant banks.

The results of the data collection exercise demonstrate a rough picture for the non-uniform nature of the distribution of loss amounts and frequency across various business lines and event types. The results also suggested that the losses are highly right-skewed and have a heavy right tail (i.e. the total loss amounts are highly driven by the high-magnitude 'tail events') [5]. For example, the Commercial Banking business line includes losses of a relatively low frequency (roughly 7% of total) but the second highest severity (roughly 35% of total). As for the losses classified by event type, the losses in the Damage to Physical Assets category (such as losses due to natural disasters) account for less than 1% of the total number of losses, but almost 30% of the aggregate amount. In particular, the 'retail banking/ external fraud' and 'commercial banking/ damage to physical assets' combinations account for over 10% of the total loss amount each, with the first pair accounting for 38% and the second for merely 0.1% of the total number of loss events [5].



(a) Business Line



(b) Event Types

Figure 1: Percentage Frequency and Severity of Operational Losses across Business Lines and Event Types

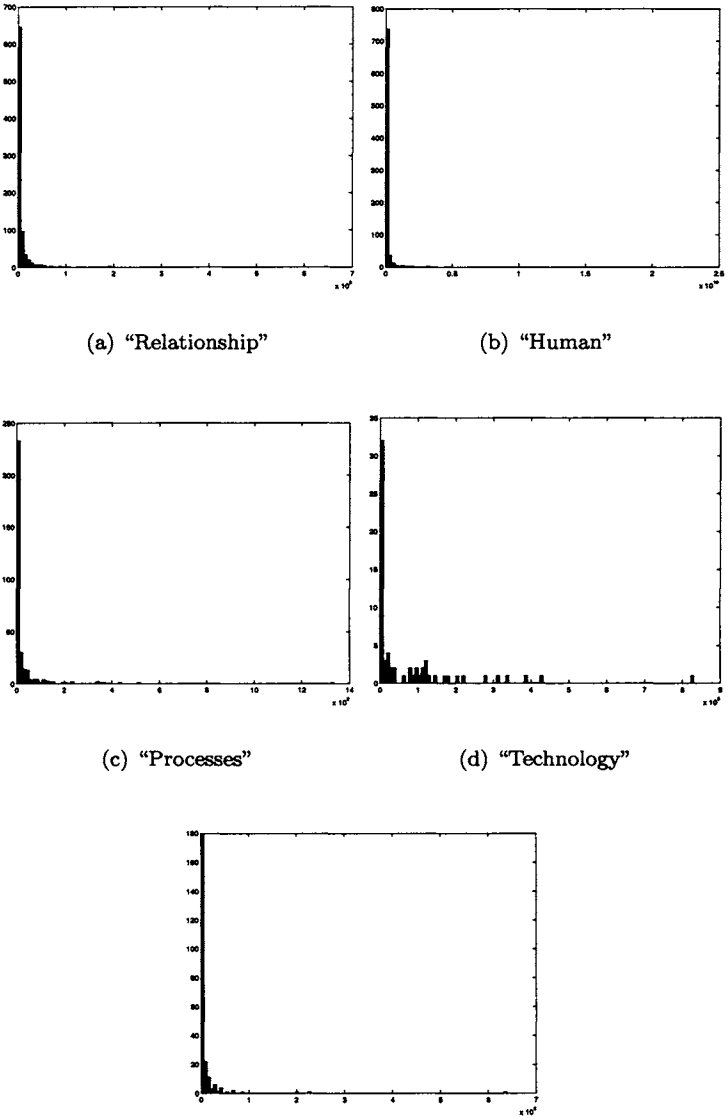
6.2 Analysis of 1980-2002 External Operational Loss Data

In this section we fit various distributions to the operational risk data, obtained from a major European operational loss data provider. The external database is comprised of operational loss events throughout the world. The original loss data cover losses in the period 1950-2002. As discussed earlier, the data in external databases are subject to minimum recording thresholds of \$1 million. A few recorded data points were below \$1 million in nominal value, so we excluded them from the dataset. Furthermore, we excluded the observations before 1980 because of relatively few data points available (which is most likely due to poor data recording practices). The final dataset for the analysis covered losses in US dollars for the time period between 1980 and 2002. It consists of five types of losses: "Relationship" (such as events related to legal issues, negligence and sales-related fraud), "Human" (such as events related to employee errors, physical injury and internal fraud), "Processes" (such as events related to business errors, supervision, security and transactions), "Technology" (such as events related to technology and computer failure and telecommunications) and "External" (such as events related to natural and man-made disasters and external fraud). The loss amounts have been adjusted for inflation using the Consumer Price Index from the U.S. Department of Labor. The numbers of data points of each of the "Relationship", "Human", "Processes", "Technology", and "External" types are $n = 849, 813, 325, 67$, and 233, respectively. Figure 2 presents the histograms for the five loss types of data. The histograms (the horizontal axis covers the entire range of the data) indicate the leptokurtic nature of the data: a very high peak is observed close to zero, and an extended right tail indicates the right-skewness and high dispersion of the data values.

6.2.1 Operational Loss Frequency Process

Figure 3 portrays the annually aggregated number of losses for the "External" type losses, shown by the dotted-line. It suggests that the accumulation is somewhat similar to a continuous cdf-like process, supporting the use of a non-homogeneous Poisson process. We consider two following functions,[5] each with four parameters:

[5]Of course, asymptotically (as time increases) such functions would produce a constant cumulative intensity. However, for this particular sample and this particular time frame, this form of the intensity function appears to provide a good fit.



(a) "Relationship"

(b) "Human"

(c) "Processes"

(d) "Technology"

(e) "External"

Figure 2: Histograms for operational loss data from external sources.

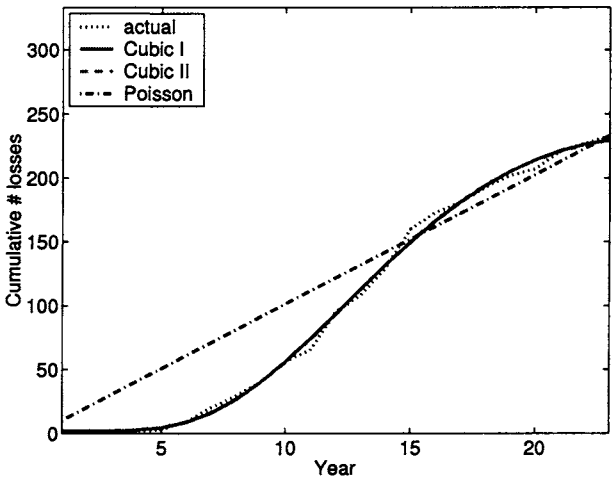


Figure 3: Annual accumulated number of "External" operational losses, with fitted non-homogeneous and simple Poisson models.

- **Type I**: Lognormal cdf-like process of form:

$$\Lambda(t) = a + \frac{b}{\sqrt{2\pi}c} \exp\left\{-\frac{(\log t - d)^2}{2c^2}\right\};$$

- **Type II**: Logweibull cdf-like process of form

$$\Lambda(t) = a - b \exp\left\{-c \log^d t\right\}.$$

We obtain the four parameters a, b, c, d by minimizing the Mean Square Error. Table 1 demonstrates the estimated parameters and the Mean Square Error (MSE) and the Mean Absolute Error (MAE) for the cumulative intensities and a simple homogeneous Poisson process with a constant intensity factor. Figure 3 shows the three fits plotted together with the actual aggregated number of events. The two non-linear fits appear to be superior to the standard Poisson, for all 5 loss datasets. We thus reject the conjecture that the counting process is simple Poisson.

Table 1: Fitted frequency functions to the "External" type losses.

process					MSE	MAE
Type I	a	b	c	d		
	2.02	305.91	0.53	3.21	16.02	2.708
Type II	a	b	c	d		
	237.88	236.30	0.00026	8.27	14.56	2.713
Poisson				λ		
				10.13	947.32	24.67

6.2.2 Operational Loss Distributions

The following loss distributions were considered for this study.

- Exponential $\mathcal{E}xp(\beta)$ $f_X(x) = \beta e^{-\beta x}$
  $x \geq 0,\ \beta > 0$
- Lognormal $\mathcal{LN}(\mu, \sigma)$ $f_X(x) = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left\{-\frac{(\log x - \mu)^2}{2\sigma^2}\right\}$
  $x \geq 0,\ \mu, \sigma > 0$
- Weibull $\mathcal{W}eib(\beta, \tau)$ $f_X(x) = \tau\beta x^{\tau-1} \exp\{-\beta x^\tau\}$
  $x \geq 0,\ \beta, \tau > 0$
- Logweibull $\log \mathcal{W}eib(\beta, \tau)$ $f_X(x) = \frac{1}{x}\tau\beta(\log x)^{\tau-1} \exp\{-\beta(\log x)^\tau\}$
  $x \geq 0,\ \beta, \tau > 0$
- Log-αStable $\log S_\alpha(\sigma, \beta, \mu)$ $f_X(x) = \frac{g(\ln x)}{x}, g \in S_\alpha(\sigma, \beta, \mu)$
  no closed-form density
  $x > 0,\ \alpha \in (0, 2),\ \beta \in [-1, 1],\ \sigma, \mu > 0$
- Symmetric αStable $S\alpha S(\sigma)$ $f_Y(y) = g(y), g \in S_\alpha(\sigma, 0, 0)$,
  no closed-form density
  $x = |y|,\ \alpha \in (0, 2),\ \sigma > 0$

All except the $S\alpha S$ distributions are defined on $\mathbb{R}_+$, making them applicable for the operational loss data. For the $S\alpha S$ distribution, we symmetrized the data by multiplying the losses by -1 and then adding them to the original dataset.

We fitted conditional loss distribution to the data (see Equation (4.8)) with the minimum threshold of $u = 1,000,000$, using the method of Maximum Likelihood. Parameter estimates are presented in Table 2.

Table 2: Estimated parameters for the loss data separated by event type.

$\widehat{\gamma}_{\text{MLE}}$	"Rel.-ship"	"Human"	"Proc."	"Tech."	"Ext."
$\mathcal{E}xp(\beta)$					
β	$11.25{\cdot}10^{-9}$	$7.27{\cdot}10^{-9}$	$3.51{\cdot}10^{-9}$	$13.08{\cdot}10^{-9}$	$9.77{\cdot}10^{-9}$
$\mathcal{LN}(\mu,\sigma)$					
μ	16.1911	15.4627	17.1600	15.1880	15.7125
σ	2.0654	2.5642	2.3249	2.7867	2.3639
$\mathcal{W}eib(\beta,\tau)$					
β	0.0032	0.0240	0.0021	0.0103	0.0108
τ	0.3538	0.2526	0.3515	0.2938	0.2933
$\log \mathcal{W}eib(\beta,\tau)$					
β	$0.27{\cdot}10^{-8}$	$30.73{\cdot}10^{-8}$	$0.11{\cdot}10^{-8}$	$11.06{\cdot}10^{-8}$	$2.82{\cdot}10^{-8}$
τ	7.0197	7.0197	7.1614	5.7555	6.2307
$\log S_\alpha(\sigma,\beta,\mu)$					
α	1.9340	1.4042	2.0000	2.0000	1.3313
β	-1	-1	0.8195	0.8040	-1
σ	1.5198	2.8957	1.6476	1.9894	2.7031
μ	15.9616	10.5108	17.1535	15.1351	10.1928
$S\alpha S(\sigma)$					
α	0.6592	0.6061	0.5748	0.1827	0.5905
σ	$1.0{\cdot}10^{7}$	$0.71{\cdot}10^{7}$	$1.99{\cdot}10^{7}$	$0.17{\cdot}10^{7}$	$0.71{\cdot}10^{7}$

6.2.3 Validation Tests for Loss Models

A variety of tests can be considered to examine the goodness of fit of the distributions to the data. Typical tests include performing the Likelihood-Ratio test, examination of Quantile-Quantile plots, forecasting, and various

tests based on the comparison of the fitted distribution and the empirical distribution (the so-called EDF-based tests). In this paper, we focus on the last testing procedure, because it allows us to compare separately the fits of the distributions around the center and around the tails of the data.

First, we compare the magnitudes of several EDF test statistics between different models. A lower test statistic value indicates a better fit (in the sense that the value of the norm, which is based on the distance between the fitted and empirical cdf's, is smaller). Second, we compare the p-values based on the EDF tests. p-values indicate the proportion of times in which the samples drawn from the same fitted distributions have a higher statistic value. In other words, a higher p-value suggests a better fit. The following test statistics are considered: Kolmogorov-Smirnov (D), Kuiper (V), quadratic Anderson-Darling (A^2), quadratic *"upper tail"* Anderson-Darling (A^2_{up}) and Cramér-von Mises (W^2), computed as

$$\begin{aligned} D &= \max\left\{D^+, D^-\right\}, \\ V &= D^+ + D^-, \\ A^2 &= n\int_{-\infty}^{\infty} \frac{(F_n(x)-\widehat{F}(x))^2}{\widehat{F}(x)(1-\widehat{F}(x))} d\widehat{F}(x), \\ A^2_{up} &= n\int_{-\infty}^{\infty} \frac{(F_n(x)-\widehat{F}(x))^2}{(1-\widehat{F}(x))^2} d\widehat{F}(x), \\ W^2 &= n\int_{-\infty}^{\infty} (F_n(x)-\widehat{F}(x))^2 d\widehat{F}(x), \end{aligned}$$

where $D^+ = \sqrt{n}\sup_x\{F_n(x)-\widehat{F}(x)\}$ and $D^- = \sqrt{n}\sup_x\{\widehat{F}(x)-F_n(x)\}$. The A^2_{up} statistic was introduced and studied in [9], and designed to put most of the weight on the upper tail. $F_n(x)$ is the empirical cdf, and $\widehat{F}(x)$ is defined as

$$\widehat{F}(x) = \begin{cases} \frac{\widehat{F}_\gamma(x)-\widehat{F}_\gamma(u)}{1-\widehat{F}_\gamma(u)} & x > u \\ 0 & x \le u. \end{cases}$$

Table 3[6] demonstrates that on the basis of the statistic values we would tend to conclude that Logweibull, Weibull or Lognormal densities describe best the dispersion of the operational loss data: the statistics are the lowest for these models in most cases. However, if we wish to test the null that a given dataset belongs to a family of distributions (such as Lognormal or

[6]The fit of the exponential distribution is totally unsatisfactory and the results have been omitted for saving space.

Table 3: Goodness-of-fit test statistics for the loss data.

	"Rel.-ship"	"Human"	"Proc."	"Tech."	"Ext."
		D			
$\mathcal{LN}$	0.8056	0.8758	0.6854	1.1453	0.6504
$\mathcal{W}eib$	0.5553	0.8065	0.6110	1.0922	0.4752
$\log \mathcal{W}eib$	0.5284	0.9030	0.5398	1.1099	0.6893
$\log S_\alpha(\sigma, \beta, \mu)$	1.5929	9.5186	0.6931	1.1540	7.3275
$S\alpha S$	1.1634	1.1628	1.3949	2.0672	0.7222
		V			
$\mathcal{LN}$	1.3341	1.5265	1.1262	1.7896	1.2144
$\mathcal{W}eib$	1.0821	1.5439	1.0620	1.9004	0.9498
$\log \mathcal{W}eib$	1.0061	1.5771	0.9966	1.9244	1.1020
$\log S_\alpha(\sigma, \beta, \mu)$	1.6930	9.5619	1.1490	1.7793	7.4089
$S\alpha S$	2.0695	2.1537	1.9537	2.8003	1.4305
		A^2			
$\mathcal{LN}$	0.7554	0.7505	0.4624	1.3778	0.5816
$\mathcal{W}eib$	0.7073	0.7908	0.2069	1.4536	0.3470
$\log \mathcal{W}eib$	0.4682	0.7560	0.1721	1.5355	0.4711
$\log S_\alpha(\sigma, \beta, \mu)$	3.8067	304.61	0.4759	1.3646	194.74
$S\alpha S$	4.4723	11.9320	6.5235	19.6225	1.7804
		A^2_{up}			
$\mathcal{LN}$	4.6122	4.5160	4.0556	6.4213	2.5993
$\mathcal{W}eib$	13.8191	8.6610	2.2340	4.8723	5.3662
$\log \mathcal{W}eib$	5.2316	4.5125	1.4221	5.2992	4.1429
$\log S_\alpha(\sigma, \beta, \mu)$	10.1990	4198.9	4.0910	6.4919	3132.6
$S\alpha S$	$2.6 \cdot 10^{14}$	$3.3 \cdot 10^{11}$	$6.8 \cdot 10^{14}$	$7.2 \cdot 10^{10}$	$1.2 \cdot 10^{10}$
		W^2			
$\mathcal{LN}$	0.1012	0.0804	0.0603	0.2087	0.0745
$\mathcal{W}eib$	0.0716	0.0823	0.0338	0.2281	0.0337
$\log \mathcal{W}eib$	0.0479	0.0915	0.0241	0.2379	0.0563
$\log S_\alpha(\sigma, \beta, \mu)$	0.7076	44.5156	0.0660	0.2072	24.3662
$S\alpha S$	0.3630	0.2535	0.3748	1.4411	0.1348

Table 4: p-values associated with the goodness-of-fit test statistics for the loss data. p-values were obtained from 1,000 simulated samples. Figures in bold show p-values whenever their values were the first or second highest.

	"Rel.-ship"	"Human"	"Proc."	"Tech."	"Ext."
		D			
$\mathcal{LN}$	0.082	0.032	0.297	<0.005	0.326
$\mathcal{W}eib$	0.625	0.103	0.455	<0.005	0.852
$\log \mathcal{W}eib$	0.699	0.074	0.656	<0.005	0.296
$\log S_\alpha(\sigma, \beta, \mu)$	0.295	**0.319**	0.244	<0.005	0.396
$S\alpha S$	0.034	**0.352**	0.085	**0.085**	**0.586**
		V			
$\mathcal{LN}$	0.138	0.039	0.345	0.005	0.266
$\mathcal{W}eib$	0.514	0.051	0.532	<0.005	0.726
$\log \mathcal{W}eib$	0.628	0.050	0.637	<0.005	0.476
$\log S_\alpha(\sigma, \beta, \mu)$	0.295	**0.324**	0.342	**0.007**	0.458
$S\alpha S$	<0.005	0.026	0.067	**0.067**	0.339
		A^2			
$\mathcal{LN}$	0.043	0.408	0.223	<0.005	0.120
$\mathcal{W}eib$	0.072	0.112	0.875	<0.005	0.519
$\log \mathcal{W}eib$	0.289	0.392	0.945	<0.005	0.338
$\log S_\alpha(\sigma, \beta, \mu)$	**0.290**	0.215	0.202	<0.005	0.284
$S\alpha S$	**0.992**	**0.436**	**0.964**	**>0.995**	**0.841**
		A^2_{up}			
$\mathcal{LN}$	0.401	0.408	0.367	0.067	0.589
$\mathcal{W}eib$	0.081	0.112	0.758	0.087	0.164
$\log \mathcal{W}eib$	0.282	0.392	0.977	0.114	0.283
$\log S_\alpha(\sigma, \beta, \mu)$	**0.288**	0.215	0.361	0.060	0.128
$S\alpha S$	<0.005	**0.436**	0.193	**>0.995**	**0.841**
		W^2			
$\mathcal{LN}$	0.086	0.166	0.294	<0.005	0.210
$\mathcal{W}eib$	0.249	0.188	0.755	<0.005	0.781
$\log \mathcal{W}eib$	0.514	0.217	0.918	<0.005	0.458
$\log S_\alpha(\sigma, \beta, \mu)$	**0.292**	**0.315**	0.258	<0.005	0.366
$S\alpha S$	<0.005	0.027	0.102	**0.964**	0.265

Stable), then the test is not parameter-free, and we need to estimate the p-values for each hypothetical scenario. These results are demonstrated in Table 4. Now the situation is quite different from the one in Table 3. The numbers in bold indicate the cases in which $\log S_\alpha(\sigma, \beta, \mu)$ or $S\alpha S$ fit resulted in first or second highest p-values across the same group (i.e. for the same type of EDF test for a range of distributions, with a particular dataset). As is clear from the table, in the majority of cases (17 out of 25) either $\log S_\alpha(\sigma, \beta, \mu)$ or $S\alpha S$, or even both, resulted in the highest p-values. This supports the conjecture that the overall distribution of operational losses[7] are heavy-tailed. Fitting $\log S_\alpha(\sigma, \beta, \mu)$ or $S\alpha S$ distributions to the data appears a valid solution.

7 Summary

The objective of this paper was to examine the models underlying in the operational risk process. The conjecture that operational losses follow a compound Cox process was investigated for the external operational loss data of five loss types covering a 23 year period. The results of the empirical analysis provide evidence of heavy tailedness of the data in the right tail. Moreover, fitting $\log S_\alpha(\sigma, \beta, \mu)$ distribution to the loss severity data or symmetric $S_\alpha(\sigma, \beta, \mu)$ distribution to the symmetrized data resulted in high p-values in a number of goodness of fit tests, suggesting a good fit. In particular, the two distributions are shown to fit the data very well in the upper tail, which remains the central concern in the framework of operational risk modeling and regulation.

Furthermore, the paper suggested a number of models for the frequency of losses. A simple Poisson process with a fixed intensity factor appears too restrictive and unrealistic. A non-homogeneous Poisson process with a time-varying intensity function was fitted to the loss data and showed a superior fit to the homogeneous Poisson process.

Directions for future research include developing robust models for the operational risk modeling. For example, with $S_\alpha(\sigma, \beta, \mu)$ distributions, for the case when the shape parameter α is below or equal to unity, the first moment (and hence the mean) and the second moment (hence the variance) do not exist, making such distribution difficult to use for practical purposes. Possible solutions would include working with trimmed data, truncated data, or 'Winsorized' data, or splitting the dataset into two parts - the low-

[7] Another approach would be to split each data set into two parts: the main body of the data and the right tail. Some empirical evidence suggests that the two parts of the data follow different laws. Extreme Value Theory is an approach that can be used for such analysis.

and medium-size losses and the tail losses - and analyzing the properties of each separately.

8 Acknowledgements

We are thankful to S. Stoyanov of the FinAnalytica Inc. for computational assistance. We are also grateful to S. Trück for providing us with the data on which the empirical study was based.

C. Menn is grateful for research support provided by the German Academic Exchange Service (DAAD).

S.T. Rachev gratefully acknowledges research support by grants from Division of Mathematical, Life and Physical Sciences, College of Letters and Science, University of California, Santa Barbara, the German Research Foundation (DFG) and the German Academic Exchange Service (DAAD).

References

[1] Bening VE, Korolev VY (2002) Generalized Poisson Models and their Applications in Insurance and Finance. VSP International Science Publishers, Utrecht, Boston

[2] BIS (1998) Operational risk management. http://www.bis.org

[3] BIS (2001) Consultative document: operational risk. http://www.bis.org

[4] BIS (2001) Working paper on the regulatory treatment of operational risk. http://www.bis.org

[5] BIS (2003) The 2002 loss data collection exercise for operational risk: summary of the data collected. http://www.bis.org

[6] BIS (2004) International convergence of capital measurement and capital standards. http://www.bis.org

[7] Chernobai A, Menn C, Rachev ST, Trück S (2005) Estimation of operational value-at-risk in the presence of minimum collection thresholds. Department of Statistics and Applied Probability, University of California Santa Barbara

[8] Chernobai A, Menn C, Trück S, Rachev S (2005) A note on the estimation of the frequency and severity distribution of operational losses. Mathematical Scientist 30(2)

[9] Chernobai A, Rachev S, Fabozzi F (2005) Composite goodness-of-fit tests for left-truncated loss samples. Department of Statistics and Applied Probability, University of California Santa Barbara

[10] Embrechts P, Klüppelberg C, Mikosch T (1997) Modeling Extremal Events for Insurance and Finance. Springer-Verlag, Berlin

[11] Jorion P (2000) Value-at-Risk: The New Benchmark for Managing Financial Risk, Second edition. McGraw-Hill, New York

[12] Moscadelli M, Chernobai A, Rachev ST (2005) Treatment of missing data in the field of operational risk: Effects on parameter estimates, EL, UL and CVaR measures. Operational Risk, June 2005

[13] Rachev ST, Mittnik S (2000) Stable Paretian Models in Finance. John Wiley & Sons, New York

[14] Samorodnitsky G, Taqqu MS (1994) Stable Non-Gaussian Random Processes. Stochastic Models with Infinite Variance. Chapman & Hall, London

[15] Zolotarev VM (1986) One-dimensional stable distributions. Translations of Mathematical Monographs, vol. 65. American Mathematical Society, Providence

Fuzzy-Nutzwertanalyse und Fuzzy-AHP

Heinrich Rommelfanger
Institut für Statistik und Mathematik
Johann Wolfgang Goethe-Universität

1 Einleitung

Die Nutzwertanalyse und der Analytic Hierarchical Process (AHP) sind bekannte, auch in der Praxis verwendete Verfahren zur Lösung von Mehrzielentscheidungen. Beide berücksichtigen die beschränkte Informationsverarbeitungskapazität bzw. die eingeschränkte Rationalität eines Entscheiders und entsprechen so dem Wunsch der Praktiker nach realistischeren und anwendbaren Entscheidungsunterstützungsmethoden. Wie empirisch nachgewiesen wurde, haben Menschen große Schwierigkeiten, Alternativen widerspruchsfrei anzuordnen, wenn mehr als zwei Ziele beachtet werden müssen, vgl. May [6, S. 9-13]. Entsprechend werden in der Nutzwertanalyse und dem AHP bei Vorliegen sehr vieler Ziele diese durch ein hierarchisch aufgebautes Zielsystem strukturiert, in dem schrittweise nur wenige, zumeist zwei oder drei Teilziele zu einem höheren Ziel aggregiert werden, vgl. Abbildung 1.

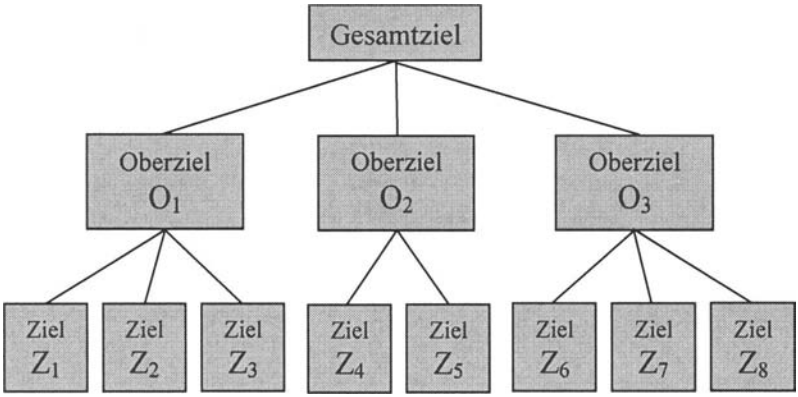


Abbildung 1: Hierarchisch aufgebautes Zielsystem

Die Aggregation der Werte erfolgt in beiden Verfahren mittels gewichteter Addition der partiellen Nutzenwerte, was eine starke Präferenzunabhängigkeit

der Ziele und vor allem kardinal skalierte Größen voraussetzt. Die Ermittlung von kardinal skalierten Nutzenwerten ist jedoch generell schwierig.

Die Bestimmung von Nutzenwerten bereitet insbesondere dann große Probleme, wenn die Bewertung der Zielkriterien ordinal skaliert oder sogar nur in linguistischer Form gegeben ist.

Zur Gewinnung der Nutzenwerte für die einzelnen Ziele auf der Basisebene der Zielhierarchie wird von Zangemeister [17] vorgeschlagen, eine Bewertungsmatrix aufzustellen, in der alle möglichen Zielerreichungsgrade in das Intervall [0 , 10] abgebildet werden. Eine derart präzise Einstufung soll durch darüber gelegte Intervallklassen, die verbal durch Benotungen von „sehr schlecht" bis „sehr gut" erläutert sind, unterstützt werden. Fraglich bleibt, ob ein Entscheider tatsächlich in der Lage ist, auf diese Weise jeder Teilzielausprägung einen wahrheitsgetreuen eindeutigen Nutzenwert zuzuordnen. Realistischer erscheint die in Abschnitt 5 diskutierte Annahme, dass ein Mensch nur Fuzzy-Nutzenwerte über [0 , 10] angegeben kann, vgl. [8, S.90ff].

Nach Zangemeister [17] kann die Wahl der Zielgewichte vom Entscheidungsträger frei vorgenommen werden. Oft wird aber empfohlen, zur Bestimmung der Gewichte paarweise die Austauschraten zwischen den Zielen zu ermitteln, d.h. man stellt fest, um wieviel sich der Nutzenwert bezüglich eines Ziels k erhöhen muss, wenn der Nutzenwert des Ziels r um den absoluten Wert Δ reduziert wird. Die sich so ergebenden Austauschraten $a_{kr} = g_r / g_r$ sind eindeutig bestimmt, wenn die Summe der Gewichte auf 1 normiert wird.

"Widerspruchsfreie Präferenzen" liegen bei einer linearen additiven Entscheidungsregel dann vor, wenn die ermittelten Austauschraten der ***Konsistenzbedingung*** $a_{kr} \cdot a_{rs} = a_{ks}$ genügen. Sind alle Austauschraten a_{rs} positiv, so folgt aus der Konsistenzbedingung und $a_{kk} = 1$ die Formel $a_{rk} = 1/a_{kr}$.

Um diese reziproken Paarvergleichsmatrizen aufstellen zu können, müssen sich die Austauschraten auf einer Verhältnisskala messen lassen, was in der Realität kaum gegeben ist, da Einzelwertfunktionen bestenfalls auf Intervall-Skalenniveau vorliegen. Bislang ist daher ungeklärt, unter welchen Voraussetzungen die reziproken Matrizen und die daraus abgeleiteten Gewichtsvektoren als sinnvoll konstruiert anzusehen sind.

Gemäß der Definition der Austauschraten weist eine konsistente Paarvergleichsmatrix **A** eine spezielle Form auf, bei der alle Spaltenvektoren Vielfache von einander sind und jede Spalte somit einen äquivalenten Gewichtsvektor darstellt. Durch Normierung der Summe der Gewichte auf 1 erhält man dann den normierten Gewichtsvektor, vgl. Abbildung 2.

Eine rechnerische Alternative zur Bestimmung desselben Gewichtsektors präsentiert Thomas L. Saaty [12] in seinem Analytic Hierarchy Process. Saaty nutzt die Tatsache aus, dass beim Vorliegen einer konsistenten Paarvergleichsmatrix **A** der Gewichtsvektor **g** dem Eigenvektor von **A** zum größten Eigenwert

von **A** entspricht. Dieser größte Eigenwert ist in diesem Fall gleich der Ordnung der konsistenten Paarvergleichsmatrix und alle übrigen Eigenwerte sind dann gleich 0.

$$\mathbf{A} = \begin{pmatrix} 1 & 8 & 2 & \frac{16}{5} \\ \frac{1}{8} & 1 & \frac{1}{4} & \frac{2}{5} \\ \frac{1}{2} & 4 & 1 & \frac{8}{5} \\ \frac{5}{16} & \frac{5}{2} & \frac{5}{8} & 1 \end{pmatrix} \qquad \mathbf{g} = \begin{pmatrix} \frac{16}{31} \\ \frac{2}{31} \\ \frac{8}{31} \\ \frac{5}{31} \end{pmatrix}$$

Abbildung 2: Paarvergleichsmatrix und normierter Gewichtsvektor

Bei der realen Anwendung kommt es häufig vor, dass sich keine konsistente Paarvergleichsmatrix ergibt. Die Ursache für "widersprüchliche Präferenzen" könnte darin liegen, dass die additive Entscheidungsregel inadäquat ist. Wahrscheinlicher ist aber, dass die lineare Entscheidungsregel brauchbar ist, die Widersprüche jedoch in der beschränkten Informationsverarbeitungskapazität oder der anderweitig eingeschränkten Rationalität des Entscheidungsträgers begründet sind. Entsprechend letzterer Argumentation tritt Saaty [12] dafür ein, bei kleineren Verstößen gegen die Konsistenzbedingung weiterhin die gewichtete Addition zu verwenden und, solange ein von ihm formulierter Konsistenz-Index von 0,1 nicht überschritten wird, als Gewichtsvektor weiterhin den Eigenvektor zum größten Eigenwert der Paarvergleichsmatrix **A** zu verwenden.

Ferner schlägt Saaty vor, Paarvergleichsmatrizen auch zur Bestimmung der Nutzenbewertungen zu verwenden. Die Zielhierarchie erhält dadurch eine weitere Ebene, da jedes Teilziel auf der bisherigen Basisebene in die Alternativen verzweigt wird, vgl. Abbildung 3. Diese Vorgehensweise ist allerdings nur dann sinnvoll, wenn wenige Alternativen vorliegen.

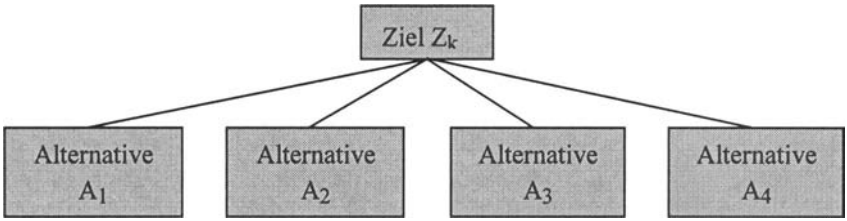


Abbildung 3: Zusätzliche Hierarchiestufe mit den Alternativen

Zentrales Element derartiger Mehrzielentscheidungen stellt damit die reziproke Paarvergleichsmatrix dar. In realen Entscheidungssituationen ist es jedoch

für Entscheider sehr schwierig, die Austauschraten bzw. für Alternativen die partiellen Nutzenwerte festzulegen. Saaty schlägt daher vor, eine 9-Punkte-Skala zu verwenden, die er wie folgt interpretiert, vgl. Saaty [11, S. 73, 74]:

Tabelle 1: Punkteskala von Saaty für die Paarvergleiche

1	gleiche Bedeutung	Beide verglichenen Elemente haben die gleiche Bedeutung für das nächst höhere Element (Ziel).
3	etwas größere Bedeutung	Erfahrung und Einschätzung sprechen für eine etwas größere Bedeutung.
5	erheblich größere Bedeutung	Erfahrung und Einschätzung sprechen für eine erheblich größere Bedeutung eines Elements im Vergleich zu einem anderen.
7	sehr viel größere Bedeutung	Die sehr viel größere Bedeutung eines Elements hat sich in der Vergangenheit klar gezeigt.
9	absolut dominierend	Es handelt sich um den größtmöglichen Bedeutungsunterschied zwischen Elementen.
2,4, 6,8	Zwischenwerte	Zwischen zwei benachbarten Urteilen muss eine Übereinkunft, ein Kompromiss, getroffen werden.

Saatys 9-Punkte-Skala ist rational nicht gerechtfertigt und angreifbar. Zum einen können andere Skalen zu anderen Rangordnungen der Alternativen führen, was in vergleichenden Untersuchungen verschiedener MADM-Verfahren gezeigt wurde. Z.B. wurden in [16] AHP-Ansätze mit zwei unterschiedlichen Skalen - Saaty's Originalskala und eine geometrische Skala - gegenübergestellt mit dem Ergebnis, dass die Rangordnungen jeweils unterschiedlich stark in Abhängigkeit der Anzahl von Alternativen, Kriterien und Verteilungen differieren. Viel entscheidender ist jedoch, dass die von Saaty verwendete 9-Punkte-Skala keine kardinal skalierten Größen liefert, so dass aus messtheoretischer Sicht die Bestimmung der reziproken Werte oder eine additive Verknüpfung mit Gewichten nicht akzeptabel ist. In Tung/Tang [15] werden daher Verschiebungen der Skalierung vorgeschlagen. Durch geeignete Transformationen kann - wie in [4] gezeigt - dem geforderten Skalenniveau zwar besser entsprochen werden, das erörterte Dilemma, dass die Paarvergleiche ordinal skaliert vorliegen, für die Bestimmung der reziproken Werte jedoch Verhältnisskalenniveau bzw. für die Berechnung der Gewichtsverteilung kardinal skalierte Daten zwingend sind, wird allerdings nicht überzeugend behoben.

Zum anderen ist die Definition der Präferenz in den Paarvergleichen äußerst schwammig, und der von Saaty [10] hierzu benutzte Begriff „Fuzziness" hat wenig zu tun mit der Fuzzy Set-Theorie. Es wird z.B. nicht erklärt, was er dar-

unter versteht, dass eine Alternative "etwas größere", "erheblich größere", sehr viel größere" Bedeutung" als die Vergleichsalternative hat.

Da außerdem Erfahrungen zeigen, dass Anwender Austauschraten eher im dem Sinne verstehen, dass ein Ziel x-mal wichtiger als ein anderes ist, scheint es für die praktische Umsetzung sinnvoller, dieser eingängigen Auffassung zu folgen und Saatys Interpretation zu verlassen. Darüber hinaus sollten nur "weiche" Paarvergleiche, wie "das Ziel 1 ist mir ca. 3-mal so wichtig wie Ziel 2", nicht künstlich verschärft werden, sondern mit der gegebenen Unschärfe mathematisch modelliert werden. Eine adäquate und realistischere Modellierung lässt sich durch die Verwendung von Fuzzy-Größen erreichen, wie dies schon Buckley [1] vorschlägt, vgl. auch [2], [5] und [7]. Im nachfolgenden Abschnitt 2 wird eine Variante dieses Ansatzes vorgestellt, bei der es ausreicht, dass ein Entscheider die Austauschraten nur näherungsweise in Form von Fuzzy-Intervallen des ε-λ-Typs angibt.

2 Bestimmung des Gewichtsvektors aus einer Paarvergleichsmatrix mit Fuzzy-Ausgleichsraten

In realen Problemen ist ein Entscheider oft nicht in der Lage, beim Paarvergleich alle Ausgleichsraten exakt anzugeben und daraus eine konsistente Paarvergleichsmatrix aufzustellen. Zumindest bei einigen der Paarvergleiche besitzt er häufig nur eine ungefähre Vorstellung, um wieviel er das eine Ziel wichtiger erachtet als das andere. Solche nur größenordnungsmäßig bekannten Größen lassen sich durch Fuzzy-Mengen mathematisch beschreiben. Dabei reicht es zumeist aus, die bei praktischen Anwendungen bewährte Form von Fuzzy-Intervallen des ε-λ-Typs zu verwenden, welche zusätzlich den Vorteil besitzen, dass arithmetische Rechnungen leicht ausführbar sind. Fuzzy-Intervalle $\tilde{a}_{ij} = (\underline{a}_{ij}^{\varepsilon}; \underline{a}_{ij}^{\lambda}; \underline{a}_{ij}; \bar{a}_{ij}; \bar{a}_{ij}^{\lambda}; \bar{a}_{ij}^{\varepsilon})^{\varepsilon,\lambda}$ des ε-λ-Typs haben stückweise lineare Zugehörigkeitsfunktionen und lassen sich durch 6 Werte hinreichend genau beschreiben, vgl. Abbildung 4 und Rommelfanger [9]. Als Spezialfälle umfassen sie auch trapezförmige Fuzzy-Intervalle, Fuzzy-Zahlen und auch reelle Zahlen. Bei geringem Informationsstand kann man auf das λ-Niveau verzichten.

Betrachten wir zur Illustration die Auswahl einer Studentenwohnung, bei der die 3 Ziele "Größe der Wohnung (in qm)", "Entfernung zur Universität (in Gehminuten)" und "Lärmbelästigung" berücksichtigt werden. Während in [13, S. 169ff] die in Tabelle 2 angegebene Paarvergleichsmatrix verwendet wird, wollen wir hier die in Tabelle 3 beschriebene Paarvergleichsmatrix mit Fuzzy-Intervallen verwenden. Einige der Paarvergleiche sind dabei bewusst recht unscharf gewählt.

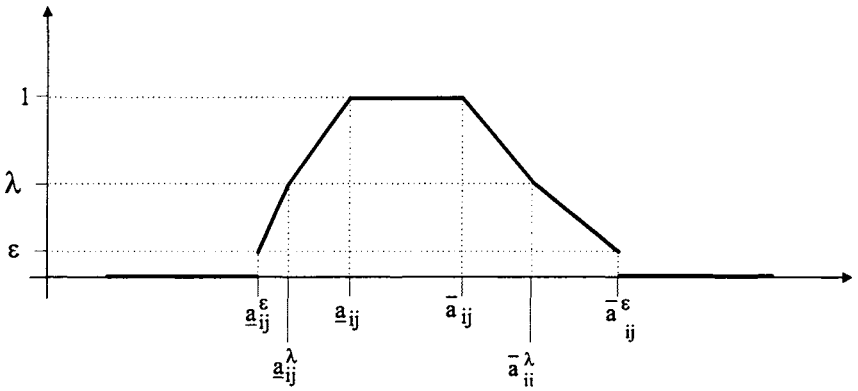


Abbildung 4: $\tilde{a}_{ij} = (\underline{a}_{ij}^{\varepsilon}; \underline{a}_{ij}^{\lambda}; \underline{a}_{ij}; \bar{a}_{ij}; \bar{a}_{ij}^{\lambda}; \bar{a}_{ij}^{\varepsilon})^{\varepsilon,\lambda}$

Tabelle 2: Paarvergleichsmatrix mit reellen Zahlen

	Größe	Entfernung	Lärm
Größe	1	9	3
Entfernung	$\frac{1}{9}$	1	$\frac{1}{4}$
Lärm	$\frac{1}{3}$	4	1

Tabelle 3: Paarvergleichsmatrix mit Fuzzy-Intervallen

	Größe	Entfernung	Lärm
Größe	(1;1;1;1;1;1)	(7;8;8,5 ;9,5;10;11)	(2;2,5;3;3;3,5;4)
Entfer-nung	$(\frac{1}{11};\frac{1}{10};\frac{2}{19};\frac{2}{17};\frac{1}{8};\frac{1}{7})$	(1;1;1 ;1;1;1)	$(\frac{1}{6};\frac{1}{5};\frac{2}{9};\frac{2}{7};\frac{1}{3};\frac{2}{5})$
Lärm	$(\frac{1}{4};\frac{3}{7};\frac{1}{3};\frac{1}{3};\frac{2}{5};\frac{1}{2})$	(2,5;3;3,5 ;4,5;5;6)	(1;1;1;1;1;1)
Σ	$(\frac{59}{44};\frac{107}{70};\frac{82}{57};\frac{74}{51};\frac{61}{40};\frac{23}{14})$	(10,5;12;13 ;15;16;18)	$(\frac{19}{6};\frac{16}{5};\frac{38}{9};\frac{30}{7};\frac{29}{6};\frac{27}{5})$

Würde man die Paarvergleichsmatrix mit Fuzzy-Intervallen interpretieren als eine Zusammenfassung von sechs Paarvergleichsmatrizen, bei denen jeweils nur die Werte von $\underline{a}_{ij}^{\varepsilon}$, $\underline{a}_{ij}^{\lambda}$, $\underline{a}_{ij}$, $\bar{a}_{ij}$, $\bar{a}_{ij}^{\lambda}$ oder $\bar{a}_{ij}^{\varepsilon}$ benutzt werden, so ließe sich für

jede einzelne dieser Paarvergleichsmatrizen der Eigenvektor zum größten Eigenwert berechnen. Es ist aber nicht zu erwarten, dass die so berechneten und dann normierten Eigenvektoren so geordnet sind, dass sie zu einem „Fuzzy-Eigenvektor" zusammengefasst werden können. Vielmehr stellt sich die kaum beantwortbare Frage, welcher dieser Eigenvektoren als Gewichtsvektor genommen werden sollte.

Wenig überzeugend ist unserer Ansicht nach der Vorschlag von Cheng und Mon [3], die vorliegenden Fuzzy-Ausgleichsraten zunächst zu defuzzifizieren und dann als Gewichtsvektor den normierten Eigenvektor zum größten Eigenwert der sich so ergebenden "scharfen" Paarvergleichsmatrix zu verwenden. Zur Defuzzifizierung der dort verwandten triangularen Fuzzy-Zahlen $\tilde{a}_{ij} = (\underline{a}_{ij}; a_{ij}; \overline{a}_{ij})$ schlagen Cheng und Mon die Verwendung des Wertes $\hat{a}_{ij} = (1-\lambda)\cdot \underline{a}_{ij}^{\alpha} + \lambda \cdot \overline{a}_{ij}^{\alpha}$ vor, wobei $\underline{a}_{ij}^{\alpha}$ und $\overline{a}_{ij}^{\alpha}$ die Endpunkte der α-Niveaumenge $[\underline{a}_{ij}^{\alpha}, \overline{a}_{ij}^{\alpha}]$ sind und λ ein Optimismusparamter ist, der neben dem Zugehörigkeitsniveau $\alpha \in]0,1]$ vom Entscheider festzulegen ist.

In dieser Arbeit soll daher ein anderer Weg zur Bestimmung eines Gewichtsvektors vorgeschlagen werden, der dem Konzept der Fuzzy-Mengen- Theorie eher gerecht wird und als Komponenten Fuzzy-Intervalle des ε-λ-Typs aufweist. Die gewählte Vorgehensweise basiert auf der Tatsache, dass in einer konsistenten Paarvergleichsmatrix alle Spaltenvektoren Vielfache voneinander sind und normiert den Gewichtsvektor bzw. den Eigenvektor zum größten Eigenwert ergeben. Ist die Konsistenz-Bedingung nicht erfüllt, ist es daher folgerichtig, den Gewichtsvektor aus den normierten Spaltenvektoren zu mitteln.

Als Operator bietet sich unserer Ansicht nach vor allem das Arithmetische Mittel an, da die so ermittelten Gewichte zur gewichteten Addition der Teilnutzen Verwendung finden. Buckley [1] verwendet das Geometrische Mittel zur Berechnung der Gewichte ohne dies weiter zu diskutieren - „It is not our intention to get involved in the debate over which procedure is best". Buckley weist lediglich darauf hin, dass im Falle einer konsistenten Paarvergleichsmatrix (mit reellen Zahlen) das Geometrische Mittel zum gleichen Gewichtsvektor führt wie Saatys Eigenvektor-Methode. Diese Aussage gilt aber auch für das Arithmetische Mittel. Für beide Mitteloperatoren ist es aber auf jeden Fall notwendig, zunächst die Spaltenvektoren zu normieren und dann erst zu mitteln. Die umgekehrte Reihenfolge, wie sie z.B. Buckley verwendet, führt zwar bei konsistenten Matrizen zum gleichen Gewichtsvektor, bei nicht-konsistenten Matrizen ist dies aber nicht mehr gegeben, wie Berechnungen zur Paarvergleichsmatrix aus Tabelle 4 zeigen.

Tabelle 4: Normierte Gewichtsvektoren zur Abbildung 4

Verfahren zur Berechnung des Gewichtevektors im Fall reeller Zahlen	Größe	Entfernung	Lärm
Eigenvektormethode nach Saaty	0,681	0,069	0,250
Arithmetisches Mittel nach Normierung	0,6804	0,0691	0,2505
Geometrisches Mittel nach Normierung	0,6798	0,0686	0,2516
Arithmetisches Mittel vor Normierung	0,660	0,069	0,271
Geometrisches Mittel vor Normierung (Buckley)	0,6262	0,0807	0,2931

Übertragen wir nun diese Überlegungen auf Fuzzy-Intervalle des ε-λ-Typs, so sind, um die Spaltenvektoren einer Paarvergleichsmatrix zu normieren, zunächst die Spaltensummen $\tilde{\sigma}_1, \tilde{\sigma}_2, \ldots, \tilde{\sigma}_n$ zu berechnen:

$$\tilde{\sigma}_j = (\underline{\sigma}_j^{\varepsilon}; \underline{\sigma}_j^{\lambda}; \underline{\sigma}_j; \overline{\sigma}_j; \overline{\sigma}_j^{\lambda}; \overline{\sigma}_j^{\varepsilon})^{\varepsilon,\lambda} = \tilde{a}_{1j} \oplus \tilde{a}_{2j} \oplus \cdots \oplus \tilde{a}_{nj}$$

$$= \left(\sum_{i=1}^{n} \underline{a}_{ij}^{\varepsilon}; \sum_{i=1}^{n} \underline{a}_{ij}^{\lambda}; \sum_{i=1}^{n} \underline{a}_{ij}; \sum_{i=1}^{n} \overline{a}_{ij}; \sum_{i=1}^{n} \overline{a}_{ij}^{\lambda}; \sum_{i=1}^{n} \overline{a}_{ij}^{\varepsilon}\right)^{\varepsilon,\lambda}. \quad (1)$$

Durch die Normierung

$$\tilde{a}_{ij}^{norm} = \tilde{a}_{ij} \cdot \tilde{\sigma}_j = \left(\frac{\underline{a}_{ij}^{\varepsilon}}{\overline{\sigma}_j^{\varepsilon}}; \frac{\underline{a}_{ij}^{\lambda}}{\overline{\sigma}_j^{\lambda}}; \frac{\underline{a}_{ij}}{\overline{\sigma}_j}; \frac{\overline{a}_{ij}}{\underline{\sigma}_j}; \frac{\overline{a}_{ij}^{\lambda}}{\underline{\sigma}_j^{\lambda}}; \frac{\overline{a}_{ij}^{\varepsilon}}{\underline{\sigma}_j^{\varepsilon}}\right)^{\varepsilon,\lambda} \quad (2)$$

berechnet man die Gewichte

$$\tilde{g}_i = \frac{1}{n} \cdot (\tilde{a}_{i1}^{norm} \oplus \tilde{a}_{i2}^{norm} \oplus \cdots \oplus \tilde{a}_{in}^{norm}), \quad (3)$$

die den Gewichtsvektor $\tilde{g}' = (\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_n)$ bilden.

Normiert man nach obigem Verfahren die Spaltenvektoren der Paarvergleichsmatrix in Tabelle 3, so erhält man:

Tabelle 5: Normierte Paarvergleichsmatrix mit Fuzzy-Intervallen

	Größe	Entfernung	Lärm
Größe	$(\frac{14}{23};\frac{40}{61};\frac{51}{74};\frac{57}{82};\frac{70}{107};\frac{44}{59})$	$(\frac{7}{18};\frac{1}{2};\frac{17}{30};\frac{19}{26};\frac{10}{12};\frac{22}{21})$	$(\frac{10}{27};\frac{15}{29};\frac{21}{30};\frac{27}{38};\frac{35}{32};\frac{24}{19})$
Entfernung	$(\frac{14}{253};\frac{4}{61};\frac{51}{703};\frac{57}{697};\frac{70}{856};\frac{44}{413})$	$(\frac{1}{18};\frac{1}{16};\frac{1}{15};\frac{1}{13};\frac{1}{12};\frac{2}{21})$	$(\frac{5}{162};\frac{6}{145};\frac{7}{135};\frac{9}{133};\frac{5}{48};\frac{12}{95})$
Lärm	$(\frac{7}{46};\frac{120}{427};\frac{51}{222};\frac{19}{82};\frac{28}{107};\frac{22}{59})$	$(\frac{5}{36};\frac{3}{16};\frac{7}{30};\frac{9}{26};\frac{5}{12};\frac{12}{21})$	$(\frac{5}{27};\frac{6}{29};\frac{7}{30};\frac{9}{38};\frac{5}{16};\frac{6}{19})$

Daraus ergibt sich durch Bildung des arithmetischen Mittels der Fuzzy-Gewichtsvektor:

Tabelle 6: Fuzzy-Gewichtsvektor

Größe	$(0{,}456;0{,}558;0{,}652;0{,}712;0{,}860;1{,}019)^{\varepsilon,\lambda}$
Entfernung	$(0{,}047;0{,}056;0{,}064;0{,}075;0{,}090;0{,}109)^{\varepsilon,\lambda}$
Lärm	$(0{,}159;0{,}225;0{,}232;0{,}272;0{,}330;0{,}420)^{\varepsilon,\lambda}$

Offensichtlich sind in der Paarvergleichsmatrix in Tabelle 3 einige Ausgleichsraten recht fuzzy vorgegeben, woraufhin die normierten Paarvergleiche in Tabelle 5 und die daraus berechneten Fuzzy-Gewichte in Tabelle 6 noch fuzzier geraten. Bedingt wird dies durch die verwendete Definition der erweiterten Division, die auf dem Zadehschen Erweiterungsprinzip basiert. Es können sich damit Gewichtskonstellationen ergeben mit einer Summe der Einzelgewichte größer 1, was wenig überzeugend erscheint. Darüber hinaus führt diese "traditionelle" Art der Normierung dazu, dass der Charakter der Austauschraten verändert wird: Sowohl aus scharfen Größen als auch aus Fuzzy-Zahlen resultieren Fuzzy-Intervalle des ε-λ-Typs. In Abschnitt 4 werden wir einen neuen Vorschlag zur Normierung der Spaltenvektoren vorstellen, der unserer Ansicht sinnvoller ist, da er die vorgegebene Unschärfe beibehält und die vorgegebenen Werte vom Typ her übernimmt.

3 Alternativenauswahl mit Fuzzy-Gewichten und scharfen Teilnutzenwerten

In diesem Abschnitt wollen wir die Frage diskutieren, wie die optimale Alternative ausgewählt werden soll, wenn der Entscheider nur in der Lage ist, die Gewichte der einzelnen Ziele ungenau, hier in Gestalt von Fuzzy-Intervallen des ε-λ-Typs auszudrücken. Dagegen sei es ihm möglich, die Teilnutzenwerte exakt anzugeben, wie dies in der Nutzwertanalyse unterstellt wird.

Illustriert werden soll die Vorgehensweise weiterhin an dem Beispiel „Auswahl einer Studentenwohnung", das sich an Schneeweiß [13, S. 169ff] anlehnt.

Dem Studenten werden die 4 Wohnungen A, B, C und D angeboten, die in Tabelle 7 durch die Attribute "Größe (qm)", "Entfernung zur Uni (in Gehminuten)" und "Lärmbelästigung" beschrieben werden.

Tabelle 7: Attribute der Wohnungen

Wohnung	A	B	C	D
Größe (qm)	52	23	15	30
Entfernung (Min)	30	5	25	15
Lärmbelästigung	laut	sehr laut	leise	noch nicht zu laut

Im betrachteten Fall sei der Entscheider in der Lage, diese Attribute durch Teilnutzenwerte auf einer Skala [0 , 10] exakt zu beschreiben:

Tabelle 8: Teilnutzenwerte u_{ik} *für die Attribute der Wohnungen*

Wohnung	A	B	C	D
Größe (qm)	10	2	1	4
Entfernung (Min)	1	10	2	5
Lärmbelästigung	2	1	9	5

Die Gesamtnutzenbewertung für jede Wohnung k errechnet sich dann gemäß der Formel

$$\tilde{u}_k = \tilde{g}_1 \cdot u_{1k} \oplus \tilde{g}_2 \cdot u_{2k} \oplus \cdots \oplus \tilde{g}_n \cdot u_{nk} \quad . \tag{4}$$

Für das Beispiel erhält man dann:

Tabelle 9: Gesamtnutzen der einzelnen Wohnungen

Wohnung	Gesamtnutzen
A	$(4{,}925;6{,}086;7{,}048;7{,}739;9{,}350;11{,}139)^{\varepsilon,\lambda}$
B	$(1{,}541;1{,}901;2{,}176;2{,}446;2{,}950;3{,}548)^{\varepsilon,\lambda}$
C	$(1{,}981;2{,}695;2{,}868;3{,}310;4{,}010;5{,}017)^{\varepsilon,\lambda}$
D	$(2{,}854;3{,}637;4{,}088;4{,}583;5{,}540;6{,}721)^{\varepsilon,\lambda}$

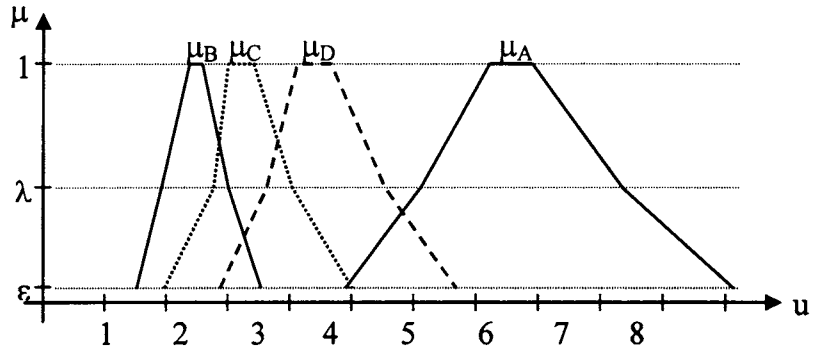


Abbildung 5: Gesamtnutzen der einzelnen Wohnungen

Ein geeignetes Kriterium zur Auswahl der optimalen Alternative ist die ε-Präferenz, die wie folgt definiert ist:

Eine Fuzzy-Menge $\widetilde{B}$ wird einer Fuzzy-Menge $\widetilde{C}$ auf dem Niveau $\varepsilon \in [0,1]$ vorgezogen und man schreibt $\widetilde{B} \succ_\varepsilon \widetilde{C}$, wenn ε die kleinste reelle Zahl ist, so dass

$$\text{Sup } B_\alpha \geq \text{Sup } C_\alpha \quad \text{und} \quad \text{Inf } B_\alpha \geq \text{Inf } C_\alpha \quad \text{für alle } \alpha \in [\varepsilon,1] \qquad (5)$$

und für wenigstens ein $\alpha \in [\varepsilon,1]$ eine dieser Ungleichungen im strengen Sinne erfüllt ist.

Dabei bezeichnen $B_\alpha = \{x \in X | \mu_B(x) \geq \alpha\}$ und $C_\alpha = \{x \in X | \mu_C(x) \geq \alpha\}$ die α-Niveaumengen von $\widetilde{B}$ bzw. $\widetilde{C}$.

Für Fuzzy-Intervalle $\widetilde{u}_k = (\underline{u}_k^\varepsilon ; \underline{u}_k^\lambda ; \underline{u}_k ; \overline{u}_k ; \overline{u}_k^\lambda ; \overline{u}_k^\varepsilon)^{\varepsilon,\lambda}$ des ε-λ-Typs vereinfachen sich die Bedingungsungleichungen der ε-Präferenz zu

$$\tilde{u}_k \succ_\varepsilon \tilde{u}_r \Leftrightarrow \underline{u}_k^\alpha > \underline{u}_r^\alpha \text{ und } \overline{u}_k^\alpha > \overline{u}_r^\alpha \text{ für } \alpha = \varepsilon, \lambda, 1. \quad (6)$$

Ein weitaus strengeres und zur Aufstellung einer Präferenzordnung meist weniger geeignetes Kriterium ist die ρ-Präferenz:

Eine Fuzzy-Menge $\tilde{B}$ wird einer Fuzzy-Menge $\tilde{C}$ auf dem Niveau $\rho \in [0,1]$ vorgezogen und man schreibt $\tilde{B} \succ_\rho \tilde{C}$, wenn ρ die kleinste reelle Zahl ist, so dass

$$\text{Inf } B_\alpha \geq \text{Sup } C_\alpha \quad \text{für alle } \alpha \in [\rho, 1]. \quad (7)$$

Die Konstruktion der hier verwendeten Fuzzy-Intervalle des ε-λ-Typs legt nahe, für ρ nur die Abstufungen ε, λ und 1 zu verwenden.

Vergleichen wir die Gesamtnutzen für die einzelnen Wohnungen, so ist nach dem ε-Präferenz-Kriterium die Wohnung A klar die beste; dann folgt D vor C. Die am schlechtesten bewertete Alternative ist die Wohnung B.

Selbst bei Verwendung der ρ-Präferenz ist die Alternative A auf dem Niveau λ besser als die anderen 3 Alternativen, verglichen mit der Alternative B sogar auf dem Niveau ε.

4 Ein neues Verfahren zur Normierung der Spalten der Paarvergleichsmatrix

Wie bereits aufgezeigt führt die Normierung der Spalten der Paarvergleichsmatrix mit Hilfe der erweiterten Division zu Werten, die relativ gesehen, noch fuzzier sind als die ursprünglichen. Darüber hinaus werden auch die vorliegenden scharfen Paarvergleiche bzw. die vorliegenden Fuzzy-Zahlen in Fuzzy-Intervalle überführt, was wenig sinnvoll ist. Auch stellt sich die Frage nach der Bedeutung der Normierung, wenn die Summen über die Werte auf dem ε-Niveau und auf dem λ-Niveau bedeutend kleiner oder größer als 1 sind.

Um dem Begriff "Normierung auf 1" genüge zu tun, überzeugt unserer Ansicht nach viel eher, wenn alle Parameter eines Fuzzy-Intervalls vom ε-λ-Typ durch die gleiche reelle Zahl dividiert werden. In Betracht kommt hier zur Normierung der j-ten Spalte vor allem das arithmetische Mittel $\sigma_j^* = \frac{1}{2}(\underline{\sigma}_j + \overline{\sigma}_j)$. Aber auch das rechnerisch aufwendigere arithmetische Mittel $\frac{1}{6}(\underline{\sigma}_j^\varepsilon + \underline{\sigma}_j^\lambda + \underline{\sigma}_j + \overline{\sigma}_j + \overline{\sigma}_j^\lambda + \overline{\sigma}_j^\varepsilon)$ ist eine denkbare Alternative.

Die Normierung erfolgt nun nach der Formel

$$\tilde{a}_{ij}^{*} = \frac{\tilde{a}_{ij}}{\sigma_j^{*}} = \left(\frac{\underline{a}_{ij}^{\varepsilon}}{\sigma_j^{*}}; \frac{\underline{a}_{ij}^{\lambda}}{\sigma_j^{*}}; \frac{\underline{a}_{ij}}{\sigma_j^{*}}; \frac{\bar{a}_{ij}}{\sigma_j^{*}}; \frac{\bar{a}_{ij}^{\lambda}}{\sigma_j^{*}}; \frac{\bar{a}_{ij}^{\varepsilon}}{\sigma_j^{*}}\right)^{\varepsilon,\lambda}. \tag{8}$$

Mit den so normierten Ausgleichsraten berechnet man dann die Gewichte

$$\tilde{g}_i^{*} = \frac{1}{n} \cdot (\tilde{a}_{i1}^{*} \oplus \tilde{a}_{i2}^{*} \oplus \cdots \oplus \tilde{a}_{in}^{*}), \tag{9}$$

die den Gewichtsvektor $\tilde{g}'^{*} = (\tilde{g}_1^{*}, \tilde{g}_2^{*}, \ldots, \tilde{g}_n^{*})$ bilden.

Normiert man mit diesem Verfahren die Spaltenvektoren der Paarvergleichsmatrix in Tab. 3 mit den Spaltensummen $(\sigma_1^{*}, \sigma_2^{*}, \sigma_3^{*}) = (1{,}445, 14, 4{,}254)$, so erhält man:

Tabelle 10: Mit σ_j^{*} normierte Paarvergleichsmatrix mit Fuzzy-Intervallen

	Größe	Entfernung	Lärm
Größe	(0,692;0,692;0,692; 0,692;0,692;0,692)	$(\frac{1}{2};\frac{4}{7};\frac{17}{28};\frac{19}{28};\frac{5}{7};\frac{11}{14})$	(0,470;0,568;0,705; 0,705;0,823;0,940)
Entfernung	(0,063;0,069;0,073; 0,081;0,087;0,099)	$(\frac{1}{14};\frac{1}{14};\frac{1}{14};\frac{1}{14};\frac{1}{14};\frac{1}{14})$	(0,039;0,047;0,052; 0,067;0,078;0,094)
Lärm	(0,173;0,231;0,231; 0,231;0,277;0,346)	$(\frac{5}{28};\frac{3}{14};\frac{7}{28};\frac{9}{28};\frac{5}{14};\frac{3}{7})$	(0,235;0,235;0,235; 0,235;0,235;0,235)

Daraus ergibt sich durch Bildung des arithmetischen Mittels der Fuzzy-Gewichtsvektor:

Tabelle 11: Fuzzy-Gewichtungsvektor $\tilde{g}^{*}$

Größe	$(0{,}554;0{,}610;0{,}668;0{,}692;0{,}743;0{,}806)^{\varepsilon,\lambda}$
Entfernung	$(0{,}058;0{,}062;0{,}065;0{,}073;0{,}079;0{,}088)^{\varepsilon,\lambda}$
Lärm	$(0{,}196;0{,}227;0{,}238;0{,}262;0{,}290;0{,}337)^{\varepsilon,\lambda}$

Ein Vergleich der Tabellen 5 und 9 bzw. 6 und 10 zeigt deutlich die Vorteile des neuen Normierungsverfahrens auf. Die in den Ausgleichsraten vorgegebene Unschärfe bleibt bei der Normierung erhalten und wird nicht vergrößert; außerdem wird die Gestalt der Fuzzy-Mengen bewahrt.

Eine weitere Konsequenz dieser Normierung ist, dass die Gesamtnutzenbewertungen, die nun nach der Formel

$$\tilde{u}_k = \tilde{g}_1^* \cdot u_{1k} \oplus \tilde{g}_2^* \cdot u_{2k} \oplus \cdots \oplus \tilde{g}_n^* \cdot u_{nk} \qquad (10)$$

berechnet werden, ebenfalls weniger fuzzy sind und damit die Präferenzordnung deutlicher wird, vgl. Abbildung 6. Für das Beispiel "Auswahl einer Studentenwohnung" erhält man nun die Gesamtnutzenwerte:

Tabelle 12: Gesamtnutzen $\tilde{u}_k$ der einzelnen Wohnungen

Wohnung	Gesamtnutzen
A	$(5{,}990;6{,}616;7{,}2218;7{,}517;8{,}089;8{,}822)^{\varepsilon,\lambda}$
B	$(1{,}884;2{,}068;2{,}224;2{,}376;2{,}566;2{,}829)^{\varepsilon,\lambda}$
C	$(2{,}434;2{,}777;2{,}940;3{,}196;3{,}511;4{,}015)^{\varepsilon,\lambda}$
D	$(3{,}486;3{,}885;4{,}187;4{,}443;4{,}817;5{,}349)^{\varepsilon,\lambda}$

Nach dem ε-Präferenz-Kriterium bleibt natürlich die Rangordnung der Wohnungen $A \succ_\varepsilon D \succ_\varepsilon C \succ_\varepsilon B$ erhalten. Die Rangfolge wird aber gefestigter, wie die ρ-Präferenz zeigt. Hiernach ist nun die Alternative A auf dem Niveau ε besser als die anderen 3 Alternativen. Auch die Alternative D ist noch auf dem Niveau ε besser als die Alternative B und auf dem Niveau λ besser als die Alternative C.

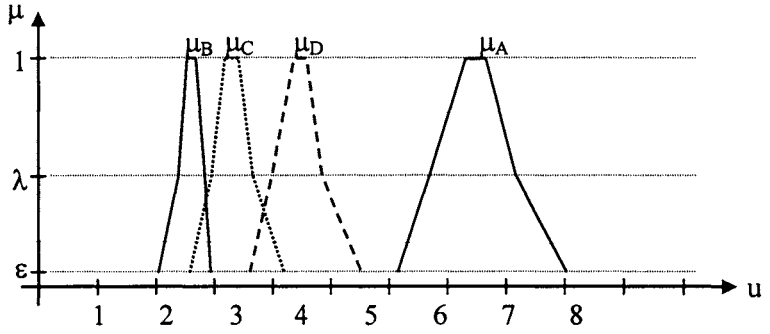


Abbildung 6: Gesamtnutzen $\tilde{u}_k$ der einzelnen Wohnungen

5 Alternativenauswahl mit Fuzzy-Gewichten und Fuzzy-Teilnutzenwerten

Wie bei der Nutzwertanalyse üblich wurde in Abschnitt 3 unterstellt, dass der Entscheider exakte Teilnutzenwerte für alle Alternativen angeben kann. Es ist aber zu bezweifeln, ob er dies in realen Entscheidungssituationen immer leisten kann, oder ob er zumindest einige der Teilnutzenwerte nur größenordnungsmäßig festzulegen vermag. Wir wollen zeigen, dass auch in diesen Fällen meist eine Rangordnung aufgestellt werden kann oder sich die Menge der in Betracht kommenden Alternativen zumindest erheblich reduzieren lässt.

Für den Fall, dass der Entscheider einige Teilnutzenwerte nur in Form von Fuzzy-Intervallen des ε-λ-Typs $\tilde{u}_{ik} = (\underline{u}_{ik}^{\varepsilon}; \underline{u}_{ik}^{\lambda}; \underline{u}_{ik}; \bar{u}_{ik}; \bar{u}_{ik}^{\lambda}; \bar{u}_{ik}^{\varepsilon})^{\varepsilon,\lambda}$ festlegen kann, berechnet sich der Gesamtnutzen einer Alternative k nach der Formel

$$\tilde{u}_k^* = \tilde{g}_1^* \otimes \tilde{u}_{1k} \oplus \tilde{g}_2^* \otimes \tilde{u}_{2k} \oplus \cdots \oplus \tilde{g}_n^* \otimes \tilde{u}_{nk} \,. \tag{11}$$

$$= (\underline{g}_i^{*\varepsilon} \cdot \underline{u}_{ik}^{\varepsilon}; \underline{g}_i^{*\lambda} \cdot \underline{u}_{ik}^{\lambda}; \underline{g}_i^{*} \cdot \underline{u}_{ik}; \bar{g}_i^{*} \cdot \bar{u}_{ik}; \bar{g}_i^{*\lambda} \cdot \bar{u}_{ik}^{\lambda}; \bar{g}_i^{*\varepsilon} \cdot \bar{u}_{ik}^{\varepsilon})^{\varepsilon,\lambda} \,.$$

Für das Beispiel "Auswahl einer Studentenwohnung" wollen wir nun annehmen, dass der Entscheidungsträger die Attribute der einzelnen Wohnungen durch Fuzzy-Teilnutzenwerte auf einer Skala [0 , 10] wie folgt beschreiben kann:

Tabelle 13: Fuzzy-Teilnutzenwerte $\tilde{u}_{ik}$ für die Attribute der Wohnungen

Wohnung	Größe	Entfernung	Lärmbelästigung
A	10	(0;0;0;1;2;3)	(0;1;2;3;4;5)
B	(0;1;2;2;3;4)	10	(0;0;0;1;2;3)
C	(0;0;0;1;2;3)	(0;1;2;3;4;5)	(7;8;9;9;9,5;10)
D	(4;4,5;5;6;7;8)	(3;4;5;5;6;7)	(4;4,5;5;6;6,5;7)

Die Gesamtnutzenbewertung für jede Wohnung k errechnet sich dann gemäß der Formel

$$\tilde{u}_k = \tilde{g}_1 \cdot u_{1k} \oplus \tilde{g}_2 \cdot u_{2k} \oplus \cdots \oplus \tilde{g}_n \cdot u_{nk} \,.$$

Für das Beispiel erhält man dann:

Tabelle 14: Gesamtnutzen $\tilde{u}_k^*$ der einzelnen Wohnungen

Wohnung	Gesamtnutzen
A	$(5{,}540;6{,}327;7{,}156;7{,}779;8{,}748;10{,}009)^{\varepsilon,\lambda}$
B	$(0{,}580;1{,}230;1{,}986;2{,}376;3{,}599;5{,}115)^{\varepsilon,\lambda}$
C	$(1{,}372;1{,}878;2{,}337;3{,}269;4{,}557;6{,}229)^{\varepsilon,\lambda}$
D	$(3{,}174;3{,}710;4{,}855;6{,}089;7{,}560;9{,}423)^{\varepsilon,\lambda}$

Die Abbildung 7 offenbart, dass selbst bei der Verwendung von unscharfen Paarvergleichen und unscharfen Teilnutzenbewertungen eine Rangfolge der Alternativen noch feststellbar ist. Nach dem ε-Präferenz-Kriterium gilt auch hier die Rangordnung $A \succ_\varepsilon D \succ_\varepsilon C \succ_\varepsilon B$. Wie eine Überprüfung mit der ρ-Präferenz zeigt, ist aber der erste Rangplatz von A gegenüber D nur noch auf dem Niveau $\rho \approx 0{,}8$ gesichert, gegenüber C auf einem Niveau von $\rho \approx 0{,}2$ und nur gegenüber B auf dem ε-Niveau.

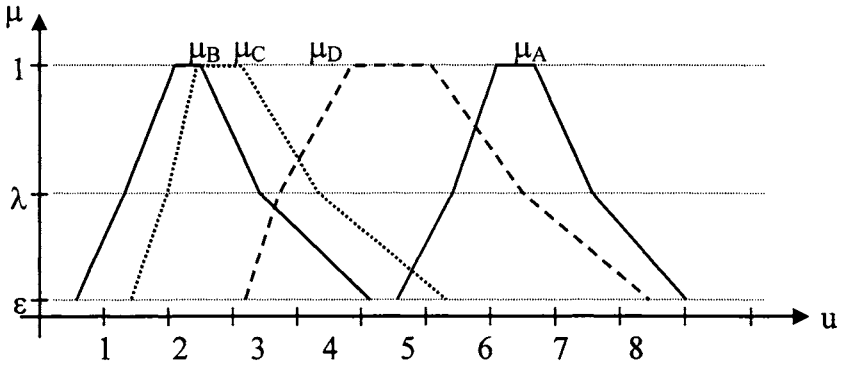


Abbildung 7: Gesamtnutzen $\tilde{u}_k^*$ der einzelnen Wohnungen

6 AHP-Ansatz zur Bewertung der Attribute

Zur Vervollständigung des Beispiels „Auswahl einer Studentenwohnung" sollen nun die Attribute Größe, Entfernung und Lärm über Paarvergleichsmatrizen bewertet werden.

Tabelle 15: Paarvergleichsmatrix für "Größe der Wohnung"

	A	B	C	D
A	(1;1;1;1;1;1)	(5;5,5;6; 6,5;7;7,5)	(9;9;9; 9;9;9)	(4;4,5;5;5;6;6)
B	$(\frac{2}{15};\frac{1}{7};\frac{2}{13};\frac{1}{6};\frac{2}{11};\frac{1}{5})$	(1;1;1; 1;1;1)	(2;2,5;3; 3;3,5;4)	$(\frac{2}{11};\frac{1}{5};\frac{2}{9};\frac{1}{4};\frac{2}{7};\frac{1}{3})$
C	$(\frac{1}{9};\frac{1}{9};\frac{1}{9};\frac{1}{9};\frac{1}{9};\frac{1}{9})$	$(\frac{1}{4};\frac{2}{7};\frac{1}{3};\frac{1}{3};\frac{2}{5};\frac{1}{2})$	(1;1;1; 1;1;1)	$(\frac{2}{13};\frac{1}{6};\frac{2}{11};\frac{1}{5};\frac{2}{9};\frac{1}{4})$
D	$(\frac{1}{6};\frac{1}{6};\frac{1}{5};\frac{1}{5};\frac{2}{9};\frac{1}{4})$	(3;3,5;4; 4,5;5;5,5)	(4;4,5;5; 5,5;6;6,5)	(1;1;1;1;1;1)

In der Tabelle 16 wird unterstellt, dass der Entscheider für den paarweisen Vergleich das λ-Niveau vernachlässigt und lediglich Fuzzy-Intervalle des Typs $(\underline{a}_{ij}^{\varepsilon};\ \underline{a}_{ij};\overline{a}_{ij};\ \overline{a}_{ij}^{\varepsilon})^{\varepsilon}$ festlegt. Nach der Berechnung des Gewichtsvektors werden dann zur weiteren Verarbeitung die Werte auf dem λ-Niveau durch lineare Approximation gewonnen. Dazu wird im Zahlenbeispiel $\lambda = 0{,}5$ und $\varepsilon = 0{,}1$ gesetzt. Die interpolierten Werte sind in der Tabelle 18 kursiv geschrieben.

Tabelle 16: Paarvergleichsmatrix für "Entfernung"

	A	B	C	D
A	(1;1; 1;1)	$(\frac{1}{9};\frac{1}{9};\frac{1}{9};\frac{1}{9})$	$(\frac{1}{4};\frac{1}{3};\frac{1}{3};\frac{1}{2})$	$(\frac{1}{7};\frac{1}{6};\frac{1}{5};\frac{1}{4})$
B	(9;9; 9;9)	(1;1; 1;1)	(6;7; 8;9)	(4;5; 5;6)
C	(2;3; 3;4)	$(\frac{1}{9};\frac{1}{8};\frac{1}{7};\frac{1}{6})$	(1;1; 1;1)	$(\frac{1}{5};\frac{1}{4};\frac{1}{4};\frac{1}{3})$
D	(4;5; 6;7)	$(\frac{1}{6};\frac{1}{5};\frac{1}{5};\frac{1}{4})$	(3;4; 4;5)	(1;1; 1;1)

Die Paarvergleichsmatrix für die Lärmbelästigung wird mit exakten Daten angesetzt, da Grundlagen für eine genauere Festlegung fehlen.

Tabelle 17: Paarvergleichsmatrix für "Lärmbelästigung"

	A	B	C	D
A	1	3	$\frac{1}{5}$	$\frac{1}{3}$
B	$\frac{1}{3}$	1	$\frac{1}{9}$	$\frac{1}{5}$
C	5	9	1	4
D	3	5	$\frac{1}{4}$	1

Nach dem neuen Verfahren ergeben sich dann die relativen Bewertungen in Bezug auf die Attribute "Größe", "Entfernung" und "Lärmbelästigung":

Tabelle 18: Relative Bewertungen der Attribute der einzelnen Wohnungen

$$\tilde{w}_{ik} = (\underline{w}_{ik}^{\varepsilon}; \underline{w}_{ik}^{\lambda}; \underline{w}_{ik}; \overline{w}_{ik}; \overline{w}_{ik}^{\lambda}; \overline{w}_{ik}^{\varepsilon})^{\varepsilon,\lambda}$$

Whg.	Größe	Entfernung	Lärm
A	(0,555; 0,585; 0,615; 0,625; 0,675; 0,685)	(0,043; *0,045*; 0,046; 0,047; *0,049*; 0,052)	0,1176
B	(0,078; 0,087; 0,098; 0,100; 0,111; 0,1235)	(0,567; *0,599*; 0,625; 0,645; *0,671*; 0,703)	0,0502
C	(0,044; 0,045; 0,0467; 0,0474; 0,0497; 0,0528)	(0,074; *0,083*; 0,091; 0,094; *0,104*; 0,115)	0,5952
D	(0,185; 0,203; 0,226; 0,243; 0,265; 0,287)	(0,180; *0,207*; 0,229; 0,243; *0,257*; 0,274)	0,2370

Die relative Bewertung der Lärmbelästigung hätte man natürlich auch mittels Eigenvektorenberechnung ermitteln können, der normierte Eigenvektor ist (0,110 ; 0,048 ; 0,611 ; 0,231).

Bei Vorgabe relat. Bewertungen $\tilde{w}_{ik} = (\underline{w}_{ik}^{\varepsilon}; \underline{w}_{ik}^{\lambda}; \underline{w}_{ik}; \overline{w}_{ik}; \overline{w}_{ik}^{\lambda}; \overline{w}_{ik}^{\varepsilon})^{\varepsilon,\lambda}$ berechnet sich der relative Gesamtnutzen einer Alternative k nach der Formel

$$\tilde{u}_k^* = \tilde{g}_1^* \otimes \tilde{w}_{1k} \oplus \tilde{g}_2^* \otimes \tilde{w}_{2k} \oplus \cdots \oplus \tilde{g}_n^* \otimes \tilde{w}_{nk} \qquad (12)$$

$$= (\underline{g}_i^{*\varepsilon} \cdot \underline{w}_{ik}^{\varepsilon}; \underline{g}_i^{*\lambda} \cdot \underline{w}_{ik}^{\lambda}; \underline{g}_i^{*} \cdot \underline{w}_{ik}; \overline{g}_i^{*} \cdot \overline{w}_{ik}; \overline{g}_i^{*\lambda} \cdot \overline{w}_{ik}^{\lambda}; \overline{g}_i^{*\varepsilon} \cdot \overline{w}_{ik}^{\varepsilon})^{\varepsilon,\lambda}.$$

Aus den Tabellen 11 und 18 errechnet man dann die relativen Gesamtnutzen:

Tabelle 19: Gesamtnutzen $\tilde{w}_k^*$ der einzelnen Wohnungen

Wohnung	Gesamtnutzen
A	$(0{,}333;0{,}386;0{,}442;0{,}467;0{,}540;0{,}596)^{\varepsilon,\lambda}$
B	$(0{,}086;0{,}102;0{,}118;0{,}129;0{,}150;0{,}178)^{\varepsilon,\lambda}$
C	$(0{,}145;0{,}158;0{,}179;0{,}196;0{,}218;0{,}2536)^{\varepsilon,\lambda}$
D	$(0{,}159;0{,}190;0{,}222;0{,}248;0{,}286;0{,}335)^{\varepsilon,\lambda}$

Die Abbildung 8 zeigt auch hier bei Verwendung des ε-Präferenz-Kriterium die Rangordnung $A \succ_\varepsilon D \succ_\varepsilon C \succ_\varepsilon B$. Darüber hinaus bestätigt die ρ-Präferenz, dass A die weitaus beste Alternative ist, denn diese ist gegenüber den Alternativen B und C auf dem Niveau $\rho = \varepsilon$ gesichert und im Vergleich zu Alternative D ist ρ nur geringfügig größer als ε.

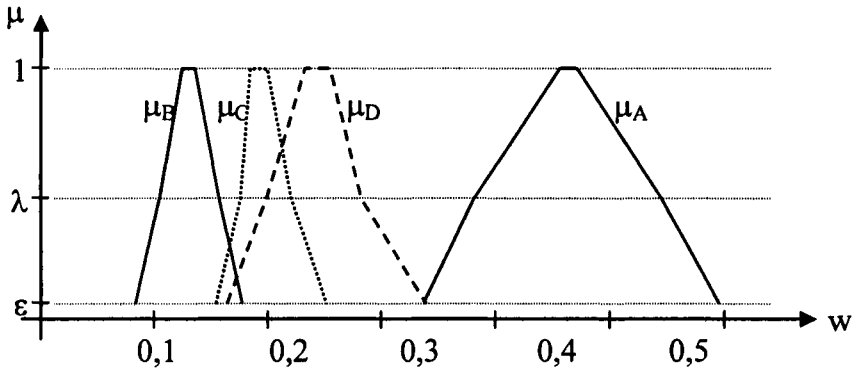


Abbildung 8: Gesamtnutzen $\tilde{u}_k^*$ der einzelnen Wohnungen

7 Schlussbemerkungen

Die vorstehenden Ausführungen verdeutlichen überzeugend, dass keine Notwendigkeit besteht, nur größenordnungsmäßig bekannte Daten künstlich zu eindeutigen Zahlen zu verdichten. Durch die Verwendung von Fuzzy-Nutzwertkonzept oder Fuzzy-AHP lässt sich die Gefahr stark verringern, mit einem Modell zu arbeiten, dessen Lösung nicht unbedingt auch eine gute Lösung des Re-

alproblems darstellt. Da Fuzzy-Intervalle auch die Spezialfälle Fuzzy-Zahlen und determinstische Zahlen umfassen, eröffnen die Fuzzy-Ansätze die Chance, den individuellen Kenntnisstand des Entscheiders adäquat einfließen zu lassen und glaubwürdige Lösungen zu erarbeiten.

Mit dem Nutzwertkonzept oder der AHP-Methode lassen sich aussagekräftige Rangordnungen der Alternativen auch dann aufstellen, wenn Ausgleichsraten und/oder Teilnutzenbewertungen nur in Form von Fuzzy-Intervallen beschrieben werden können. Selbstverständlich müssen die Fuzzy-Verfahren nicht immer so klare Ergebnisse wie im Beispiel liefern, da eine künstlich, auf dem Niveau reeller Zahlen erzeugte Trennschärfe nicht mehr gegeben ist. Die Anzahl der weiterhin in Betracht kommenden Alternativen lässt sich aber normalerweise deutlich reduzieren.

Die Fuzzy-Verfahren bieten auf diese Weise auch erstmalig einen Ausweg aus dem traditionellen Informationsbeschaffungsdilemma, denn mit der Ermittlung akzeptabler "mittlerer Werte", die für die klassischen Verfahren unabdingbar sind, ist ein sehr hoher Informationsaufwand verbunden, wenn die Gefahr der Fehlmodellierung möglichst gering gehalten werden soll. Wird dagegen das Realproblem durch ein Fuzzy-Modell beschrieben, gelingt es normalerweise mit dem vorliegenden Informationsstand im ersten Schritt zumindest einen großen Teil der Alternativen auszuschließen. Unter Abwägung von Kosten und Nutzen können dann weitere Informationen zielgerichtet beschafft werden, um genauere Aussagen über die Rangordnungen der verbliebenen Alternativen zu ermöglichen. Da im Gegensatz zu der umfangreichen ex ante-Informationsaufnahme bei klassischen Entscheidungsmodellen teure zusätzliche Informationen iterativ und zielgerichtet beschafft werden, führt diese Vorgehensweise zu einer deutlichen Verringerung der Informationskosten.

Selbstverständlich sind Fuzzy-Erweiterungen auch für große hierarchische Zielsysteme möglich. Man muss sogar erwarten, dass die Ersparnis an Informationskosten umso größer ist, je komplexer das Entscheidungsproblem ist.

Literatur

[1] Buckley JJ (1985) Fuzzy hierarchical analysis. Fuzzy Sets and Systems 17:233–247

[2] Chen ShJ, Hwang ChL (1992) Fuzzy Multiple Attribute Decision Making. Springer, Heidelberg

[3] Cheng ChH, Mon DL (1994) Evaluating weapon system by Analytical Hierarchy Process based on fuzzy scales. Fuzzy Sets and Systems 63:1–10

[4] Dodd FJ, Donegan HA, McMaster TBM (1995) Inverse inconsistency in analytic hierarchies. European Journal of Operational Research 80:86–93

[5] Laarhove PLM van, Pedrycz W (1983) A fuzzy extension of Saaty's priority theory. Fuzzy Sets and Systems 11:229 –241

[6] May KO (1954) Intransitivity, utility and the aggregation of preference patterns. Econometrica 22:1–19

[7] Ramik J (1999) Multicriteria decision making – AHP (in tschechischer Sprache). Silesian University Press, OPF SU, Karvina

[8] Rommelfanger H (1994) Fuzzy Decision Support-Systeme – Entscheiden bei Unschärfe, 2. Aufl. Springer, Heidelberg

[9] Rommelfanger H (1996) Linear programming and applications. European Journal of Operational Research 92:512–727

[10] Saaty ThL (1978) Exploring the interface between hierarchies, multiple objectives and fuzzy sets. Fuzzy Sets and Systems 1:57–68

[11] Saaty ThL (1996) The Analytic Hierarchy Process. RWS Publications, Pittsburgh

[12] Saaty ThL (1995) Decision Making for Leaders. RWS Publications, Pittsburgh

[13] Schneeweiß Ch (1991) Planung 1. Springer, Heidelberg

[14] Siler W, Buckley JJ (2004) Fuzzy Expert Systems. Wiley-Interscience

[15] Tung SL, Tang SL (1998) A comparison of the Saaty's AHP and modified AHP for right and left eigenvector inconsistency. European Journal of Operational Research 106:123–128

[16] Zanakis StH, Solomon A, Wishart N, Dublish S (1998) Multi-attribute decision making: A simulation comparison of select methods. European Journal of Operations Research 107:507–529

[17] Zangemeister Ch (1972) Nutzwertanalyse. In: Tumm GW (Hrsg.) Die neuen Methoden der Entscheidungsfindung. Verlag Moderne Industrie, München, S. 264–285

Piecewise Linear Bertrand Oligopoly

Joachim Rosenmüller

Institute of Mathematical Economics
University of Bielefeld

Abstract

We describe a modell of price competition between firms with piecewise linear cost functions. Thus, we consider "Bertrand oligopoly", an n-person noncooperative game in which players choose prices and the market, reflected by a decreasing demand function, reacts discontinuously as total demand concentrates on those firms that offer minimal prices. Firms do not have to be identical. But a notion of similarity between firms is necessary in order to prove the existence of a Nash (-Bertrand) equilibrium. Here we are only interested in an equilibrium involving all firms – the case of subgroups with "similar" members deserves an additional study.

1 Bertrand Oligopoly

Within this paper we discuss the existence of equilibria within a certain type of Bertrand Oligolpoly. [1] The main feature is the structure of the cost functions of the firm, these are supposed to be piecewise linear and convex. Such cost functions appear naturally in the context of network flow structures, where flows passing through capacity limited nodes and edges generate costs depending on the choice of edges as well. We think of such kind of flow as electricity or data material on an electronic net. Routing the flow optimally (cost minimizing) results in a linear programming problem, the solution of which yields a piecewise linear cost function. See [6] for a detailed model of this type.

The technique is not far away from standard procedures. However, apart from missing differentiability assumptions we also do not assume symmetric firms.

Most of the literature seems to rely on at least one of these assumptions. DASTIDAR [2] discusses the asymmetric case as well (assuming that costfunctions are twice differentiable), however the assumptions imposed on the model vary during the presentation. HOERNIG [4] constructs in addition to the continuum of pure equilibria existing a host of mixed ones. See also MASKIN [5] for mixed equilibria. Symmetry is also assumed in

HEHENKAMP–LEININGER [3], who discuss evolutionary Bertrand equilibria.

It would seem that none of the properties derived in the context of this literature suffers when differentiability is sacrificed and firms are just "similar".

As frequently, it is assumed that firms have a limited capacity of production. Yet they are supposed to meet market demand at the level required. The game in which firms may plan to sell less than required has different strategies and payoffs. Yet it seems that the type of equilibrium exhibited would constitute an equilibrium in the extended game as well. Within our present framework, we will not attend to this question.

The model is specified essentially by a set of piecewise linear costfunctions for the firms and a demand function of the market. We specify this data as follows.

For any nonnegative convex, monotone function D on the reals we denote by D' the derivative of a linear support function of D at t. This derivative is unique up to at most countably many points.

A decreasing function is *slowly decreasing* if it does not decrease faster than $1/t$, i.e., if $\frac{D(t)}{t} \geq -D'(t)$ holds true for all t in the domain of definition. Economically this reflects nonneagtive marginal expenditure.

Given positive real numbers d_0 and p_0, we call a function

$$D \; : \; [0, p_0] \to [0, d_0]$$

a ***demand function*** if it is continuous at 0, convex, and slowly decreasing. A demand function is hence continuous and differentiable with the exception of at most countably many points.

On the other hand, let for $K \in \mathbb{N}$

$$\boldsymbol{C}^{(0)} \; := \; (\boldsymbol{A}^{(0)}, \boldsymbol{B}^{(0)}) \in \mathbb{R}^{2K} \tag{1.1}$$

be such that $\boldsymbol{A}^{(0)} = (A_k^{(0)})_{k=0,\ldots K}$ and $\boldsymbol{B}^{(0)} = (B_k^{(0)})_{k=0,\ldots K}$ are real numbers *strictly increasing* in k and satisfy $A_0^{(0)} = 0, \; B_0^{(0)} = 0$. We put

$$\Delta_0^{(0)} \; := \; 0, \; \Delta_k^{(0)} \; := \; \frac{\Delta B_k^{(0)}}{\Delta A_k^{(0)}} \; := \; \frac{B_k^{(0)} - B_{k-1}^{(0)}}{A_k^{(0)} - A_{k-1}^{(0)}}. \tag{1.2}$$

We assume that $\Delta_k^{(0)}$ is as well *strictly increasing* in k and satisfies

$$\Delta_K = d_0, \; A_K \Delta_K - B_K \leq p_0. \tag{1.3}$$

Given these conditions, we identify the data (1.1)with the strictly increasing piecewise linear function $C^{(0)}$ given by

$$\begin{aligned} C^{(0)} &: \quad [0, d_0] \to [0, p_0], \\ C^{(0)}(t) &: \quad \max\left\{ A_k^{(0)} t - B_k^{(0)} \,\middle|\, k = 0, \dots K \right\} \quad (t \in [0, d_0]). \end{aligned} \tag{1.4}$$

As a consequence, the numbers $\Delta_k^{(0)}$ describe the arguments at which the function shows kinks: it is seen that

$$C^{(0)}(t) \;=\; A_k^{(0)} t - B_k^{(0)} \quad (t \in [\Delta_k^{(0)}, \Delta_{k+1}^{(0)}]) \tag{1.5}$$

holds true (cf. Figure 1). Thus E.g. (1.3) shows that

$$C^{(0)}(d_0) \leq p_0$$

is satisfied, thus the domain of definition is indeed $[0, d_0]$ and the range is contained in $[0, p_0]$.

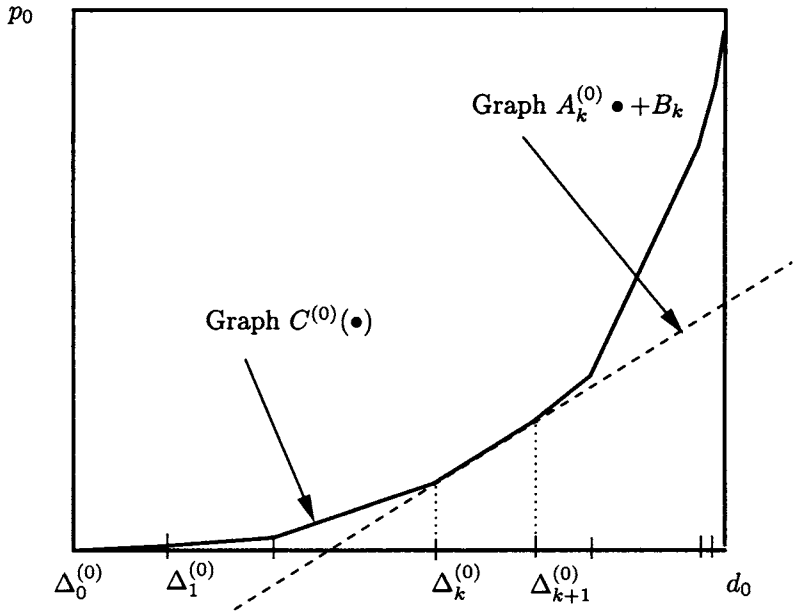


Figure 1: A Cost Function

Given p_0 and d_0, we call $C^{(0)}$ or $C^{(0)}$ a ***cost function*** if, in addition, we have

$$A_K^{(0)} \;=\; C^{(0)\prime}(d_0) > p_0, \quad p_0 d_0 < C^{(0)}(d_0) \;=\; A_K^{(0)} d_0 + B_K^{(0)}. \tag{1.6}$$

The first inequality shows that marginal cost at maximum production exceeds maximal prices. The latter reads also $p_0 d_0 - C^{(0)}(d_0) < C^{(0)}(0) = 0$, meaning that, at maximal prices, a firm's profit at the maximal possible demand is less than at zero production. The advantage of dealing with this simple setup is provided by the topology available for cost functions; this is given by the Euclidean metric on $\mathbb{R}^{2K}$.

Definition 1.1. *A **Piecewise Linear Bertrand Oligopoly (PLBO)** for a set of players $I := \{1, \dots n\}$ is a set of data*

$$\mathcal{O} := (p_0, d_0, D, (C^{(i)})_{i \in I}) \tag{1.7}$$

such that p_0, d_0 are the domain of definitions, D is a demand function and $C^{(i)}$ represents the cost function for player $i \in I$.

Given some price $p \in [0, p_0]$, we first consider the function

$$\begin{aligned} G^{(i)} &= G_p^{(i)} : [0, d_0] \to \mathbb{R} \\ G^{(i)}(t) &:= pt - C^{(i)}(t) \quad (t \in [0, d_0]). \end{aligned} \tag{1.8}$$

This function describes the profit of player $i \in I$ if the total demand of the market would accumulate at this player (the monopolistic profit function at fixed price).

However, the actual payoff within the N-person game resulting from price competition is defined via the modified demand function customary in Bertrand oligopoly as follows:

Definition 1.2. *1. For any (price) vector $\boldsymbol{p} \in \mathbb{R}^n$ let*

$$M_I(\boldsymbol{p}) := arg\ min_I\ \boldsymbol{p} = \left\{ i \in I \,\middle|\, p_i = \min_{k \in I} p_k \right\} \tag{1.9}$$

*denote the set of minimizing arguments or **minimizers** of p.*

2. Let $D : [0, p_0] \to \mathbb{R}$ be a demand function. Then, for $i \in I$ the function

$$\begin{aligned} \boldsymbol{D}^i &: [0, p_0]^I \to \mathbb{R} \\ \boldsymbol{D}^i(\boldsymbol{p}) &:= \begin{cases} 0 & i \notin M_I(\boldsymbol{p}) \\ \frac{D(p_i)}{|M_I(\boldsymbol{p})|} & i \in M_I(\boldsymbol{p}) \end{cases} \quad (\boldsymbol{p} \in [0, p_0]^I) \end{aligned} \tag{1.10}$$

*is the **Bertrand demand** function resulting from D.*

3. *Finally, let C^i or $C^{(i)}$ respectively be the costfunction of firm or player i. Then the (oligolpolistic Bertrand)* ***profit function*** *of player i is the function*

$$\begin{aligned} G^i &:= [0,p_0]^I \to \mathbb{R} \\ G^i(\boldsymbol{p}) &:= p_i D^i(\boldsymbol{p}) - C^{(i)}(D^i(\boldsymbol{p})) \end{aligned} \tag{1.11}$$

We note that $\Gamma = \Gamma^{\mathcal{O}} =: ([0,p_0]^I, (G^i)_{i\in I})$ is the n–person game based on the data of $\mathcal{O}$, the Nash equilibria of which we are concerned with. These Nash equilibria are referred to as ***Bertrand equilibria***.

For a beginning, we attempt to establish a Bertrand equilibrium in which all players participate. Then we are dealing with a price vector (strategy n–tupel) $\boldsymbol{p} = (p,\ldots,p)$, in which case some definitions simplify e.g. to

$$D^i(\boldsymbol{p}) = \frac{D(p)}{n}, \quad G^i(\boldsymbol{p}) = p\frac{D(p)}{n} - C^{(i)}\left(\frac{D(p)}{n}\right).$$

Now we wish to discuss some ***necessary*** conditions for equilibria and, if possible, establish a situation in which these conditions turn out to be ***sufficient*** as well.

Fix $\overline{\boldsymbol{p}} = (\overline{p},\ldots,\overline{p})$ in order to tentatively denote an equilibrium. Let $\overline{\xi} := D(\overline{p})$ so that

$$G^i(\overline{\boldsymbol{p}}) = \frac{\overline{p}\overline{\xi}}{n} - C^{(i)}(\frac{\overline{\xi}}{n}) = G^{(i)}\left(\frac{\overline{\xi}}{n}\right)$$

is player i's payoff in equilibrium. Here we refer to the function $G^{(i)} = G^{(i)}_{\overline{p}}$ defined via (1.8) for the fixed price $\overline{p}$.

First of all, suppose that player i wants to deviate in a way that all market demand is concentrated at his firm. That is, the player lowers his price to $\overline{p}-\varepsilon$. The equilibrium condition can be formulated to be

$$\begin{aligned} G^{(i)}\left(\frac{\overline{\xi}}{n}\right) &= G^i(\overline{\boldsymbol{p}}) \geq G^i(\overline{\boldsymbol{p}} - \varepsilon e^i) \\ &= (\overline{p}-\varepsilon)D(\overline{p}-\varepsilon) - C^{(i)}(D(\overline{p}-\varepsilon)) \quad (\varepsilon > 0) \end{aligned}$$

which implies for $\varepsilon \to 0$

$$G^{(i)}\left(\frac{\overline{\xi}}{n}\right) \geq \overline{p}\overline{\xi} - C^{(i)}(\overline{\xi}). \tag{1.12}$$

Thus,

$$\bar{p}\frac{\bar{\xi}}{n} - C^{(i)}(\frac{\bar{\xi}}{n}) \geq \bar{p}\bar{\xi} - C^{(i)}(\bar{\xi}). \tag{1.13}$$

is a necessery condition for a Bertrand equilibrium involving all players. The condition indicates that it should not be profitable for player i to draw all the market demand $\bar{\xi}$ *at equilibrium* on himself compared to share of $\frac{1}{n}\bar{\xi}$ he obtains when the equilibrium is sustained.

Essentially we would like to estabish a situation in which (1.13) is part of a *sufficient* condition as well. To this end we prove a standart lemma which is based on concavity of the cost functions and on slowly decreasing demand.

Lemma 1.3. *Let* $\mathcal{O}$ *be a* PLBO *(Definition 1.1) and let* G^i *be the resulting Bertrand profit function of player* i *(Definition 1.2). Then, for all* $\bar{p} \in [0, p_0]^I$ *and* $t > 0$ *such that* $\bar{p} - te^i \in [0, p_0]^I$ *is true, we have*

$$G^i(\bar{p} - te^i) \leq \bar{p}_i D(\bar{p}_i) - C^{(i)}\left(D(\bar{p}_i)\right). \tag{1.14}$$

Proof: Assume first of all that D is differentiable.

Note that $-C^{(i)} \circ D$ is a decreasing function. Also, D is convex and decreasing, hence we have for positive and suitable arguments ξ and η

$$D(\xi - \eta) \leq D(\xi) - \eta D'(\xi - \eta).$$

Applying this we find

$$\begin{aligned}
G^i(\bar{p} - te^i) &= (\bar{p}_i - t)D(\bar{p}_i - t) - C^{(i)}\Big(D(\bar{p}_i - t)\Big) \\
&\leq (\bar{p}_i - t)\Big[D(\bar{p}_i) - tD'(\bar{p}_i - t)\Big] \\
&\quad - C^{(i)}\Big(D(\bar{p}_i)\Big) \\
&= \bar{p}_i D(\bar{p}_i) - C^{(i)}\Big(D(\bar{p}_i)\Big) \\
&\quad - t\Big[D(\bar{p}_i) + \bar{p}_i D'(\bar{p}_i - t)\Big] \\
&\quad + \underbrace{t^2 D'(\bar{p}_i - t)}_{\leq 0} \\
&\leq \bar{p}_i D(\bar{p}_i) - C^{(i)}\Big(D(\bar{p}_i)\Big) - t\Big[D(\bar{p}_i) + \bar{p}_i D'(\bar{p}_i)\Big] \\
&\leq \bar{p}_i D(\bar{p}_i) - C^{(i)}\Big(D(\bar{p}_i)\Big);
\end{aligned}$$

the last inequality uses the requirement that D is slowly decreasing,

q.e.d.

We note that, in the particular case of $\bar{\boldsymbol{p}} = (\bar{p}, \ldots, \bar{p})$, equation (1.14) reads $\boldsymbol{G}^i(\bar{\boldsymbol{p}} - te^i) \leq \bar{p}_i D(\bar{p}_i) - C^{(i)}(D(\bar{p}_i)) =: \bar{p}\bar{\xi} - C^{(i)}(\bar{\xi}) = G^{(i)}(\bar{\xi})$. Therefore the lemma shows that equation (1.13) indeed implies that player i cannot profitably deviate by decreasing his price arbitrarily, i.e., (1.13) is sufficient in order to establish (part of) the equilibrium condition.

There is a second type of deviation of a player from equilibrium that we have to take into account. At this version, player i inserts a price exceeding the common equilibrium price $\bar{p}$. Naturally, it is much easier to see that this is not profitable. For, if $\hat{\boldsymbol{p}} = (\bar{p}, \ldots, \bar{p} + te^i, \ldots, \bar{p})$ for some $t > 0$ denotes the resulting strategy n-tuple, then the Bertrand demand accumulating at player i is $\boldsymbol{D}^i(\hat{\boldsymbol{p}}) = 0$, hence player i's payoff is

$$\boldsymbol{G}^i(\hat{\boldsymbol{p}}) = G^{(i)}(0) = -C^{(i)}(0) (= 0), \tag{1.15}$$

(if we are assuming zero fixed costs). Combining these ideas we obtain

Corollary 1.4. *Let $\mathcal{O}$ be a PLBO. Also, let $\bar{p} \in [0, p_0]$ and $\bar{\xi} := D(\bar{p})$. Suppose that, for all $i \in I$, the inequalities*

$$\bar{p}\frac{\bar{\xi}}{n} - C^{(i)}(\frac{\bar{\xi}}{n}) \geq \bar{p}\bar{\xi} - C^{(i)}(\bar{\xi}) \tag{1.16}$$

and

$$\bar{p}\frac{\bar{\xi}}{n} - C^{(i)}(\frac{\bar{\xi}}{n}) \geq -C^{(i)}(0) \tag{1.17}$$

are satisfied. Then $\bar{\boldsymbol{p}} := (\bar{p}, \ldots, \bar{p})$ is a Bertrand equilibrium in $\mathcal{O}$ involving all players.

The problem is that the quantity $\bar{\xi}$ does not depend on i, it has to be chosen simultaneously for all players. The simple idea to generate this quantity is described as follows.

If, for some $i \in \boldsymbol{I}$ and $\bar{p} \in [0, p_0]$ the inequality

$$\bar{p} \geq C^{(i)\prime} = A_0^{(i)} \tag{1.18}$$

is satisfied, then $G^{(i)}$ has either a second zero in $[0, p_0]$ or is nonnegative within all of the interval. Let

$$\xi_0 = \max\left\{x \in [0, d_0] \,\middle|\, \xi > 0, G^{(i)}(\xi) = 0\right\} \tag{1.19}$$

with the understanding that $\xi_0 = d_0$ whenever the *max* has to be extended over the empty set. Also, let

$$\hat{\xi} := \min\left\{x \in [0, d_0] \,\middle|\, G^{(i)}(\xi) \geq G^{(i)}(\eta) \quad \eta \in [0, d_0]\right\} \tag{1.20}$$

denote the first maximizer of the function $G^{(i)}$. Then we have

Corollary 1.5. *Let $\bar{p} \in [0, p_0]$ and write $\bar{\xi} := D(\bar{p})$. Suppose the corresponding function $G^{(i)} = G^{(i)}_{\bar{p}}$ with respect to its zeros and maximizers satisfies*

$$\widehat{\xi}^i \leq \frac{\bar{\xi}}{n} \leq \xi_0^i \tag{1.21}$$

for all $i \in I$. Then $\bar{p} = (\bar{p}, \ldots, \bar{p})$ is a Bertrand equilibrium.

Proof: As $\frac{\bar{\xi}}{n}$ is located to the right of the maximizer of $G^{(i)}$, we have (1.16). Furthermore, as $\frac{\bar{\xi}}{n}$ is located to the left of the zero of the same function, we have (1.17).

q.e.d.

An illustrating picture is obtained as follows (Figure 2).

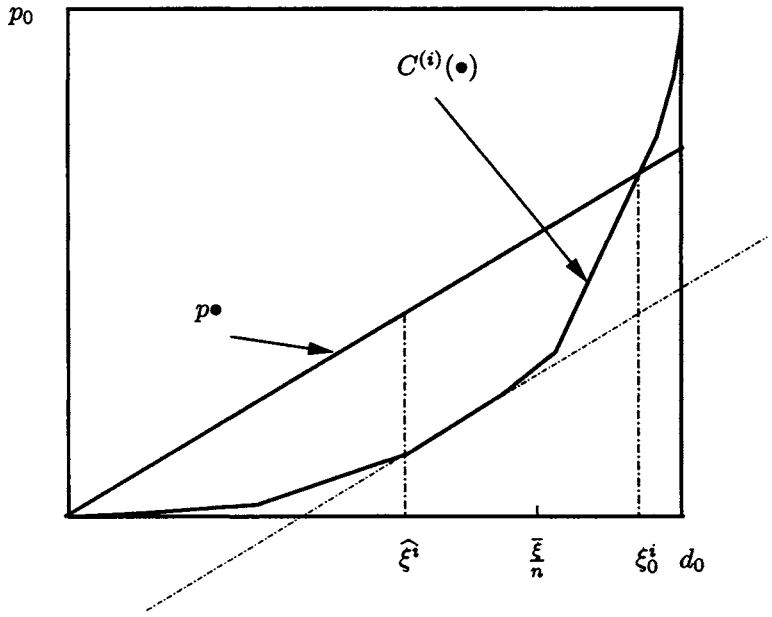


Figure 2: The Location of Equilibrium Demand

The linear function with slope p and a cost function $C^{(i)}$ are depicted simultaneously. The maximizer $\widehat{\xi}^i$ is obtained as the first point admitting of a tangent p at the cost function. The zero ξ_0^i is the last point at which the graphs of both functions intersect. If we can find $\bar{\xi}$ such that

$\frac{\overline{\xi}}{n}$ is *simultaneously* contained within the interval spanned by both points, then an equilibrium prevails. Slightly more formal: the function $G^{(i)}$ depends on p and so do the maximizer and the zero. Define interval–valued correspondences on $[0, p_0]$ by

$$\Xi^i(p) := [\widehat{\xi}^i(p), \xi_0^i(p)] \quad (p \in [0, p_0] \quad , i \in \boldsymbol{I}). \tag{1.22}$$

and

$$\boldsymbol{\Xi}(p) = \bigcap_{i \in \boldsymbol{I}} \Xi^i(p) \quad (p \in [0, p_0]). \tag{1.23}$$

Then we have

Theorem 1.6. *Suppose* $\overline{p} \in [0, p_0]$ *satisfies*

$$\frac{\overline{\xi}}{n} = \frac{D(\overline{p})}{n} \in \boldsymbol{\Xi}(\overline{p}). \tag{1.24}$$

Then $\overline{\boldsymbol{p}} = (\bar{p}, \ldots, \bar{p})$ *is a Bertrand equilibrium.*

Proof: Obviously this theorem is just a reformulation of the previous corollary. **q.e.d.**

Now in order to supply an existence theorem, some preparations are necessary in order to understand the behavior of the interval–valued correspondence $\boldsymbol{\Xi}$. This is the topic of the next section.

2 The DMP Correspondence

We start out discussing some properties of the correspondence Ξ^i that results from player i's monopolistic profit function $G^{(i)}$. For this purpose, the generic index $^{(i)}$ is tentatively omitted for the sake of more translucence.

So we consider a cost function $\boldsymbol{C}$ or C and the derived profit function G.

Let $p \geq A_0$. For any k with $A_k > p$, the function $A_k \bullet - B_k$ intersects the function $p\bullet$ at $\frac{B_k}{A_k - p}$; therefore, if the graph of C intersects the graph of $p\bullet$ (i.e., the straight line with slope p) beyond the origin, then this occcurs at the point η_0 given by

$$\begin{aligned} \eta_0 = \eta_0(p) &= \min\left\{ \frac{B_k}{A_k - p} \,\middle|\, A_k > p \right\} \\ &=: \frac{B_{k^0}}{A_{k^0} - p} \geq \frac{B_{k^0}}{A_{k^0} - A_{k^0 - 1}} \end{aligned} \tag{2.1}$$

If we agree on $k_0 := K$ for the empty set in (2.1), then the index k_0 is uniquely defined. In particular, if it so happens that $p = A_0$ is the case, then $k_0 = 1$ and $\eta_0 = \Delta_1$ follow at once. Now, η_0 defines the point at which profit is zero (apart from the origin), provided there is such point located within the admissible interval. Hence we put

$$\xi_0 = \xi_0(p) := \min\left\{d_0, \frac{B_{k^0}}{A_{k^0} - p}\right\} \quad (p \geq A_0). \tag{2.2}$$

This way we have defined the function

$$\xi_0 : [A_0, d_0] \to \mathbb{R} \tag{2.3}$$

which depends continuously on the data $\boldsymbol{C}$ and on p. The function ξ_0 is closely related to the ***average cost function*** which is given by

$$\begin{aligned} M_k := & \quad \frac{C(\Delta_k)}{\Delta_k} \\ = & \quad \frac{A_k\Delta_k - B_k}{\Delta_k} \\ = & \quad A_k - \frac{B_k}{\Delta_k} \quad (k = 1, \ldots, K). \end{aligned} \tag{2.4}$$

If p represents a slope between the average slope at Δ_k and Δ_{k+1}, then the graph of $p\bullet$ intersects the one of C just within the interval $[\Delta_k, \Delta_{k+1}]$. Hence we find that ξ_0 can be represented via

$$\xi_0(p) = \frac{B_k}{A_k - p} \quad (M_k \leq p \leq M_{k+1}, \quad k = 1, \ldots, K, \quad p \geq A_0). \tag{2.5}$$

Next, the (profit-) *maximizer correspondence* derived from $\boldsymbol{C}$ is described by

$$\widehat{\Xi}(p) := \begin{cases} \{\Delta_k\} & A_{k-1} < p < A_k \\ [\Delta_k, \Delta_{k+1}] & p = A_k \end{cases}, \quad (p \geq A_0) \tag{2.6}$$

Obviously, this correpondence is interval–valued and upper hemi continuous (*uhc*) in p. The construction shows, however, that it is also *uhc* in $\boldsymbol{C}$. Accordingly, the ***smallest profit maximizer*** is given by

$$\widehat{\xi}(p) := \begin{cases} \Delta_k & \text{if} \quad A_{k-1} < p \leq A_k, \\ 0 & \text{if} \quad p = A_0 \quad . \end{cases} \tag{2.7}$$

this quantity is either the singleton contained in $\widehat{\Xi}(p)$ or the minimum of the interval defining this correspondence. Therefore, we obtain the correspondence $\Xi : [0, p_0] \to \mathcal{P}(\mathbb{R})$ which is given by

$$\Xi(p) := \begin{cases} [\widehat{\xi}(p), \xi_0(p)] & p \in [A_0, p_0] \\ [\Delta_0, \Delta_1] = [0, \Delta_1] & p = A_0 \\ \emptyset & p \in [0, A_0) \end{cases} \tag{2.8}$$

which we call the ***dmp–correspondence*** derived from C. This is motivated as it describes the interval of non–positive or *decreasing marginal profit* for a player whose cost function is described by C. From our construction it follows easily that we have

Lemma 2.1. *Whenever $p \geq A_0$, then the dmp–correspondence Ξ is non-empty, interval–valued and uhc in p and C.*

Our next task is to estimate the length of the interval describing Ξ. We claim that this is a positive constant depending in an *uhc* way on our data To make this more precise, we claim

Lemma 2.2. *Given p_0, d_0, let C be a cost function. Then there is a lower bound $\beta > 0$ such that, whenever $p \geq A_1$, it follows that*

$$|\Xi(p)| = \xi_0(p) - \widehat{\xi}(p) \geq \beta$$

holds true.

Proof:

First of all assume that the graphs of C and $p\bullet$ intersect beyond the origin. Choose the index $\widehat{k}$ such that

$$\widehat{\xi} = \Delta_{\widehat{k}}$$

holds true. As $p \geq A_1$, we have $\widehat{k} \geq 1$ Also, let k^0 be given as in (2.2). Then clearly $\widehat{k} \leq k^0$ (both quantities depending on p). Note that $k^0 \geq 2$ follows from our assumption $p \geq A_1$. Now, if ("in the worst case"), it so happens that we have $k^0 = \widehat{k}$, then from (2.2) we deduce

$$\begin{aligned} \xi_0(p) - \widehat{\xi}(p) &= \frac{B_{k^0}}{A_{k^0} - p} - \frac{\Delta B_{\widehat{k}}}{\Delta A_{\widehat{k}}} \\ &\geq \frac{B_{k^0}}{A_{k^0} - A_{k^0-1}} - \frac{\Delta B_{\widehat{k}}}{\Delta A_{\widehat{k}}} \\ &= \frac{B_{k^0-1}}{\Delta A_{k^0}}, \end{aligned} \tag{2.9}$$

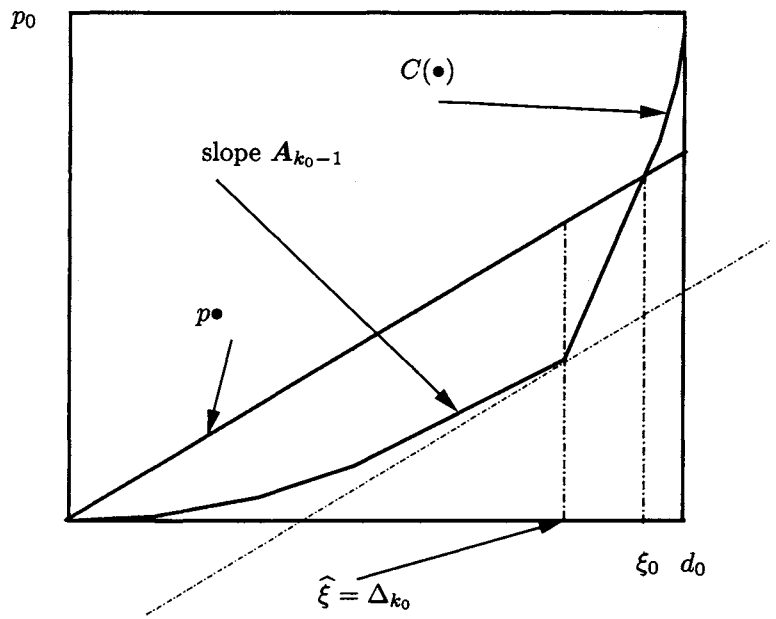


Figure 3: The Worst Case

(see Figure 3). If $k^0 > \widehat{k}$, then the estimate is even stronger. Therefore, (2.9) provides the desired lower bound.

If the graphs of C and $p\bullet$ do not intersect at some point apart from the origin, then

$$\xi_0(p) - \widehat{\xi}(p) = d_0 - \Delta_{K-1} = \Delta_K - \Delta_{K-1} \tag{2.10}$$

follows from the fact that we have $p \leq p_0 \leq A_K$ (cf. equation (1.6)).

q.e.d.

Remark 2.3. *In the above situation, if $p = A_0$ happens to be true, then we obtain*

$$\xi_0(p) - \widehat{\xi}(p) = \Delta_1 = \Delta_1 - \Delta_0 \ . \tag{2.11}$$

For $A_0 \leq p < A_1$ the estimate provided by (2.9) does not yield a lower bound. The width of $\Xi(p)$ is arbitrarily small when p approaches A_0. Yet, if we fix some $\varepsilon > 0$ and the argument p avoids the intervall $[A_0, A_0 + \varepsilon]$, then again the width of the width of $\Xi(p)$ is bounded away from 0 by some constant $\beta = \beta(\varepsilon, C)$. Thus the statement of the above theorem can be sharpened accordingly. This version may be preferable if the interval $[\Delta_1, \Delta_2]$ is relatively large.

On the other hand, as p increases, so does $\xi_0(p)$ and hence k^0. Imagine that the slopes A_k are about equally distributed so that the differences ΔA_k

are about equal. Then (2.9) suggests that, as B_{k^0-1} increases, the interval $\Xi(p)$ also increases with p. A more refined analysis shows that this depends on a relation between marginal and mean cost. The graph of $\Xi(\bullet)$ is an area bounded by the piecewise constant function $\widehat{\xi}(\bullet)$ from below and by the minimum of certain hyperbola from above (in view of (2.2)). We claim that, depending on the curvature of C, this graph "widens" with increasing p. Essentially, the idea is that marginal cost increases faster than mean cost. We clarify the precise meaning as follows.

Definition 2.4. *We shall say that a cost function C admits of* ***proper MM increments*** *if, for any $L \in \mathbb{N}$ there exists $K \in \mathbb{N}$ such that for all $k \geq l \geq K$ with $\Delta_k - \Delta_l \leq L$ it follows that $A_{l-1} \geq M_{k+1}$ is true.*

We have

Theorem 2.5. *If C amits of proper MM increments, then $|\xi_0(p) - \widehat{\xi}(p)|$ is arbitrary large for increasing p.*

Proof: Note that the statement of Definition 2.4 can be equivalently given in the following version:

> For any $L \in \mathbb{N}$ there exists $K \in \mathbb{N}$ such that for all $k \geq l \geq K$ with $A_{l-1} < M_{k+1}$ it follows that $\Delta_k - \Delta_l > L$ is true.

Now let $L \in \mathbb{N}$ be a ("large") integer and choose K accordingly. Pick $l \geq K + 1$ and consider a price $p \in (A_{l-1}, A_l]$. Then in view of (2.7) we find

$$\widehat{\xi}(p) = \Delta_l \ . \tag{2.12}$$

As ξ_0 increases in a strictly monotone way, we have

$$\xi_0(p) - \widehat{\xi}(p) > \xi_0(A_{l-1}) - \Delta_l. \tag{2.13}$$

Now choose k such that for $p = A_{l-1}$ one has

$$M_k \leq p = A_{l-1} < M_{k+1} \tag{2.14}$$

and hence

$$\xi_0(p) = \xi_0(A_{l-1}) = \frac{B_k}{A_k - p} \geq \frac{B_k}{A_k - M_k} = \Delta_k. \tag{2.15}$$

We know that $A_{k+1} \geq M_{k+1}$, hence $k + 1 > l - 1$ is necessarily true. Therefore we have $k \geq K$.

Combining (2.13) and (2.15) we obtain

$$\xi_0(p) - \widehat{\xi}(p) \geq \Delta_k - \Delta_l > L$$

which exceeds L in view of the above version of Definition 2.4,

q.e.d.

Example 2.6. Let c be a positive constant and let

$$A_k = 2ck, \quad B_k = ck(k+1) \quad (k = 0, \ldots, K) \tag{2.16}$$

such that

$$\Delta_0 = 0, \quad \Delta_k = k \quad , M_k = c(k-1) \quad (k = 1, \ldots, K) \tag{2.17}$$

is computed at once.

In view of (2.7) we find

$$\widehat{\xi}(p) = k \quad (2c(k-1) < p \leq 2ck)$$

i.e.,

$$\widehat{\xi}(p) = k \quad (\frac{p}{2c} \leq k < \frac{p}{2c} + 1).$$

Similarly, formula (2.5) shows

$$\xi_0(p) = \frac{k(k+1)}{2k - \frac{p}{c}} \quad (c(k-1) \leq p \leq ck)$$

i.e.,

$$\xi_0(p) = \frac{k(k+1)}{2k - \frac{p}{c}} \quad (\frac{p}{c} \leq k \leq \frac{p}{c} + 1).$$

If we write $t := \frac{p}{c}$ for the moment, then we obtain

$$\begin{aligned} \xi_0(t) &= \frac{k(k+1)}{2k-t} \quad (t \leq k \leq t+1) \\ \widehat{\xi}(t) &= k \qquad\qquad (\frac{t}{2} \leq k < \frac{t}{2} + 1). \end{aligned} \tag{2.18}$$

From this we derive an estimate

$$\begin{aligned} \xi_0(t) - \widehat{\xi}(t) &> \frac{t(t+1)}{2(t+1)-t} - \left(\frac{t}{2} + 1\right) \\ &= \frac{t^2 - 2t - 4}{2t+4}, \end{aligned} \tag{2.19}$$

which increases like $\frac{t}{2}$ for increasing t.

Now, if we rewrite (2.18) as

$$\begin{aligned} \xi_0(t) &= \frac{k(k+1)}{2k-t} \quad (k-1 \leq t \leq k, \quad k = 1, \ldots, K) \\ \widehat{\xi}(t) &= k \qquad\qquad (2k-2 < t \leq 2k), \end{aligned} \tag{2.20}$$

then we can provide a sketch of the correspondence Ξ as in Figure 4.

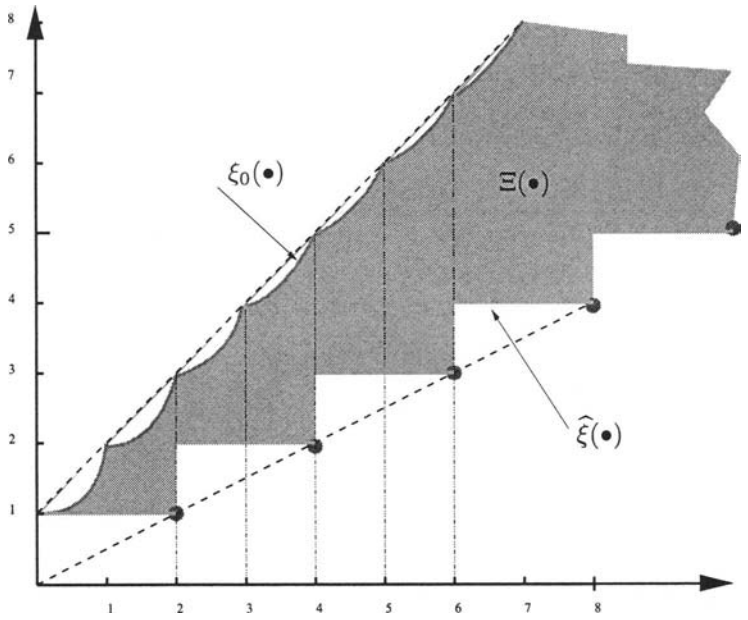


Figure 4: A DMP Correspondence

Note that this figure represents the correspondence with $t = \frac{p}{c}$ as the independent variable. If we want to represent it as a correspondence in p, then the above sketch has to be rescaled by a factor c, it is shrinking for $c > 1$ and expanding for $c < 1$.

Definition 2.7. *Let $\mathcal{O}$ be a* PLBO. *Then, for every player $i \in \boldsymbol{I}$, the correspondence $\Xi^{(i)}$ derived from $\boldsymbol{C}^{(i)}$ is the* ***dmp–correpondence*** *of player i. The correspondence*

$$\begin{aligned} \boldsymbol{\Xi} : [0, p_0] &\to \mathcal{P}(\mathbb{R}) \\ \boldsymbol{\Xi}(p) &:= \bigcap_{i \in I} \Xi^{(i)}(p) \end{aligned} \tag{2.21}$$

is called the ***dmp–correspondence*** *of $\mathcal{O}$.*

Corollary 2.8. *Let $\bar{\boldsymbol{C}}$ be a costfunction. Then there is a neighborhood of $\bar{\boldsymbol{C}}$ such that for any* PLBO *$\mathcal{O}$ with costfunctions $\boldsymbol{C}^{(i)}$ located within this neighborhood and for any $p \geq \bar{A}_1^{(i)}$ $(i \in \boldsymbol{I})$, the dmp–correspondence is nonempty, interval–valued, and* uhc *in p as well as in the data of $\mathcal{O}$.*

In other words, if the cost functions of the firms are similar, then the dmp–correspondence is nonempty provided the argument p is not too small. Similarly as in Remark 2.3, the dmp–correpondence "widens" in a sense with increasing p and this all the more with the curvature of the costfunction (which is similar for all of them) increases. Figure 4 provides the intuition: suppose that constants c_i $(i \in I)$ describe the various cost functions of players given as in Example 2.6. This amounts to a variation of the rescaling factors to be applied to Figure 4. As there is a minimum width of the correpondence to the right of $A_1 = 1$, a moderate rescaling will provide a nonempty intersection of *all* correpondences of the players.

Definition 2.9. *For any PLBO $\mathcal{O}$ the price $\bar{A}_1 := \max\{A_1^{(i)} \mid i \in I\}$ is called the **max–min marginal price** of $\mathcal{O}$. We shall say that $\mathcal{O}$ is a PLBO with **similar firms** if there is a cost function $\bar{C}$ and a neighborhood of this function such that the conclusions of Corollary 2.8 are satisfied. That is, for $p \geq \bar{A}_1$ the DMP correspondence satisfies*

$$\Xi(p) \neq \emptyset. \tag{2.22}$$

The above theorem is a *local* one. It calls for cost functions in a joint neighborhood. A *global* theorem can be constructed in the spirit of Theorem 2.5. We require that the relations between marginal costs and mean cost *globally* do not vary to much between the members of the oligopoly.

Definition 2.10. *We shall say that a PLBO $\mathcal{O}$ has **uniform MM increments**, if there is $K \in \mathbb{N}$ such that for $k, l \geq K$ and any pair of players $i, j \in I$ the following two conditions are satisfied:*

1. $A_k^i \geq M_k^j$,
2. *If $\Delta_k^i \leq \Delta_l^j$ holds true, then $A_{l-1}^i \geq M_{k+1}$ follows.*

Theorem 2.11. *Let $\mathcal{O}$ be an PLBO with uniform MM increments. Then there is $K \in \mathbb{N}$ such that for $p > A_K^i$ $(i \in I)$ we obtain $\Xi(p) \neq \emptyset$.*

Proof: We prove that, for any pair $i, j \in I$ and for sufficiently large p the relation $\xi_0^i > \widehat{\xi}^j$ is satisfied. If so, then we see that

$$\underline{\xi}_0 := \min_{i \in I} \xi_0^i \geq \max_{j \in I} \widehat{\xi}^j \tag{2.23}$$

is satisfied. It follows at once that $\underline{\xi}_0 \in \Xi(p)$ holds true. The proof follows exactly the path led in the proof of Theorem 2.5.

Now fix i and j. Let $p > \max_{\tau \in I} A_K^\tau$ and choose l such that $p \in (A_{l-1}^j, A_l^j]$ holds true. Then necessarily we obtain $l - 1 \geq K$. Also, we

know that $\widehat{\xi}^j(p) = \Delta_l^j$ holds true (from (2.7), cf. the corresponding step in Theorem 2.5). Next, choose k such that

$$M_k^i \leq A_{l-1}^j < M_{k+1}^i \tag{2.24}$$

is true. From condition *1.* above (which in this context is an assumption and in Theorem 2.5 was a result) we know that $l-1 < k+1$, hence $k \geq K$. Similarly as in (2.15) we obtain

$$\xi_0^i(A_{l-1}^j) \geq \Delta_k^i.$$

Finally, the (reverse formulation of) condition *2.* implies

$$\xi_0^i(p) - \widehat{\xi}^j(p) \geq \xi_0^i(A_{l-1}^j) - \Delta_l^j \geq \Delta_k^i - \Delta_l^j > 0, \tag{2.25}$$

q.e.d.

3 An Existence Theorem

We start out with some auxiliary theorems.

Theorem 3.1. *Let* $\Theta : [0, p_0] \to \mathcal{P}([0, d_0])$ *be an* uhc *and convex valued correpondence and let* $F : [0, p_0] \to [0, d_0]$ *be a continuous function. Assume that* $\Theta(p) \neq \emptyset$ $(p \geq \alpha)$ *holds true for some* $\alpha \in (0, p_0)$. *Now, if*

$$\Theta(\alpha) \cap [0, F(\alpha)] \neq \emptyset \tag{3.1}$$

and

$$\Theta(p_0) \cap [F(p_0), d_0] \neq \emptyset \tag{3.2}$$

holds true, then there exists $\bar{p} \in [0, p_0]$ *with* $F(\bar{p}) \in \Theta(\bar{p})$.

Proof: This is an obvious generalization of the intermediate value theorem. It can be proved by the same procedure or by a suitable application of the Kakutani fixed point theorem.

q.e.d.

Definition 3.2. *Let* $\mathcal{O}$ *be an PLOB. Let* $\alpha = \bar{A}_1$ *be the* max–min *marginal price and let* $\Xi = [\widehat{\xi}, \xi_0]$ *be the* dmp *correspondence. If*

$$\widehat{\xi}(\alpha) \leq \frac{D(\alpha)}{n} \tag{3.3}$$

and

$$\xi_0(p_0) \geq \frac{D(p_0)}{n} \tag{3.4}$$

holds true, then we shall say that demand and supply are ***intersecting****. If* α *is some other quantitiy, then we will use the definition accordingly.*

Corollary 3.3. *Let $\mathcal{O}$ be an PLOB with similar costfunctions (cf. Definition 2.9). Assume that demand and supply are intersecting (Definition 3.2). Then there exists $\bar{p} \in [\alpha, p_0]$ satisfying $\frac{D(\bar{p})}{n} \in \Xi(\bar{p})$.*

Proof: Put $\Theta := \Xi$ and $F := \frac{D}{n}$. Then apply Theorem 3.1.

q.e.d.

Thus, we require that at the *max–min* marginal price the total production (averaged out in a sense) is not sufficient to satisfy the demand and that, on the other hand, at the maximal price the demand is below of the possibilities of total production. If so, then there is a price at which per capita demand is located within the interval of decreasing profits.

Combining Corollary 3.3 and Theorem 1.6 we obtain

Theorem 3.4. *Let $\mathcal{O}$ be an PLOB with similar costfunctions (Definition 2.9). Assume that demand and supply are intersecting (Definition 3.2). Then there exists a Bertrand equilibrium. Within a certain neighborhood, the Bertrand equilibrium correspondence is* uhc *in the data of $\mathcal{O}$.*

The global versions are obtain in a quite similar fashion. However, with respect to Definition 3.2, the role of α has to be changed.

Corollary 3.5. *Let $\mathcal{O}$ be an PLOB with uniform MM increments. (cf. Definition 2.10). Let K be defined accordingly and let $\alpha = \max_{i \in I} A^i_K$. Assume that demand and supply are intersecting. Then there exists $\bar{p} \in [\alpha, p_0]$ satisfying $\frac{D(\bar{p})}{n} \in \Xi(\bar{p})$.*

Proof: In view of Theorem 2.11 we know that the dmp correspondence is nonempty for $p > \alpha$. Therefore we can again apply Theorem 3.1 and obtain the analogous result.

q.e.d.

Theorem 3.6. *Let $\mathcal{O}$ be an PLOB with uniform MM increments (Definition 2.10). Let α be as in Corollary 3.5 and assume that demand and supply are intersecting. Then there exists a Bertrand equilibrium. Within a certain neighborhood, the Bertrand equilibrium correspondence is* uhc *in the data of $\mathcal{O}$.*

Remark 3.7. *The framework of the model can be relaxed with respect to the uniform domain of definition required in Definition 1.1. It is sufficient to require that the costfunctions are mappings*

$$C^i : [0, \bar{d}^i] \to \mathbb{R} \quad (i \in I).$$

Thus, firms may have varying capacities. The role of p_0 can be played by any real number satisfying $p_0 \geq \max\{C^i(\bar{d}^i) | i \in I\}$. The property of similarity can

at once be formulated in this framework (and leads to capacity boundaries that are close to each other in a well defined sense). The intersecting property has to be slightly reformulated, e.g., w.r.t. $d_0 := \min\{\bar{d}^i | i \in \boldsymbol{I}\}$.

References

[1] Bertrand JLF (1883) Théorie mathématique de la richesse social par Léon Walras: Recherches sur les principes mathématique de la théorie des richesses par Augustin Cournot. Journal des Savants, pp. 499–508

[2] Dastidar KG (1995) On the existence of pure Bertrand equilibrium. Economic Theory 5:19–32

[3] Hehenkamp B,Leininger W (1999) A note on evolutionary stability of Bertrand equilibrium. Evolutionary Economics 9:367–371

[4] Hoernig SH (2001) Mixed Bertrand equilibria under decreasing returns to scale: An embarrassment of riches. Economics Letters 74:359–362

[5] Maskin ES (1986) The existence of equilibria with price setting firms. American Economic Review 76:382–386

[6] Rosenmüller J (2003) The electricity market. Working Paper 349, Institute of Mathematical Economics, University of Bielefeld, Bielefeld, Germany, March 13

List of Authors

Barz, Christiane, Institut für Wirtschaftstheorie und Operations Research, Universität Karlsruhe, 76128 Karlsruhe

Berninghaus, Siegfried, Institut für Wirtschaftstheorie und Operations Research, Universität Karlsruhe, 76128 Karlsruhe

Bräsel, Heidemarie, Institut für Algebra und Geometrie, Otto-von-Guericke-Universität, Postfach 4120, 39016 Magdeburg

Chernobai, Anna, Department of Statistics and Applied Probability, University of California, Santa Barbara, USA

Dinkelbach, Werner, Lehrstuhl für Betriebswirtschaftslehre, insbesondere Operations Research und Logistik, Universität des Saarlandes, 66041 Saarbrücken

Drexl, Andreas, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel, 24098 Kiel

Eichhorn, Wolfgang, Institut für Wirtschaftstheorie und Operations Research, Universität Karlsruhe, 76128 Karlsruhe

Entrup, Matthias Lütke, Department of Production Management, TU Berlin, 10585 Berlin

Frahm, Johannes, Vattenfall Europe Information Services GmbH, Überseering 12, 22297 Hamburg

Geiger, Martin Josef, Institut für Betriebswirtschaftslehre, Universität Hohenheim, 70593 Stuttgart

Grunow, Martin, Department of Production Management, TU Berlin, 10585 Berlin

Günther, Hans-Otto, Department of Production Management, TU Berlin, 10585 Berlin

Habenicht, Walter, Institut für Betriebswirtschaftslehre, Universität Hohenheim, 70593 Stuttgart

Hempel, Lorenz, Institut für Mathematik und Physik, Bauhaus-Universität Weimar, 99423 Weimar

Hinderer, Karl, Institut für Mathematische Stochastik, Universität Karlsruhe, 76128 Karlsruhe

Houtman, Joachim, Seminar für Logistik und Verkehr, Johann-Wolfgang-Goethe-Universität, 60054 Frankfurt am Main

Inderfurth, Karl, Lehrstuhl für Betriebswirtschaftslehre, insbesondere Produktion und Logistik, Otto-von-Guericke-Universität, 39016 Magdeburg

Isermann, Heinz, Seminar für Logistik und Verkehr, Johann-Wolfgang-Goethe-Universität, 60054 Frankfurt am Main

Kämmerer, Lutz, TXTe-solutions GmbH, Mansfelder Straße 48, 06108 Halle (Saale)

Kimms, Alf, Lehrstuhl für Allgemeine Betriebswirtschaftslehre, insbesondere Produktion und Logistik, TU Bergakademie Freiberg, 09599 Freiberg

Kolisch, Rainer, Lehrstuhl für Technische Dienstleistungen und Operations Management, TU München, 80333 München

Liou, Yeong-Cheng, Department of Information Management, Cheng Shiu University, Kaohsiung County 833, Taiwan

Menn, Christian, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, USA

Morlock, Martin, Department of Economics, University of Gießen, 35394 Gießen

Pallaschke, Diethard, Institut für Statistik und Mathematische Wirtschaftstheorie, Universität Karlsruhe, 76128 Karlsruhe

Rachev, Svetlozar T., Institut für Statistik und Mathematische Wirtschaftstheorie, Universität Karlsruhe, 76128 Karlsruhe

Rommelfanger, Heinrich, Institut für Statistik und Mathematik, Johann-Wolfgang-Goethe-Universität, 60054 Frankfurt am Main

Rosenmüller, Joachim, Institute of Mathematical Economics, Universität Bielefeld, 33501 Bielefeld

Salewski, Frank, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel, 24098 Kiel

Schaible, Siegfried, A. G. Anderson Graduate School of Management, University of California, Riverside, USA

Schmidt, Joachim, PTV Planung Transport Verkehr AG, Stumpfstraße 1, 76131 Karlsruhe

Schmiedel, Roland, Institut für Mathematik und Physik, Bauhaus-Universität Weimar, 99423 Weimar

Schwindt, Christoph, Institut für Wirtschaftswissenschaft, TU Clausthal, 38678 Clausthal-Zellerfeld

Stark, Christoph, Institut für Wirtschaftswissenschaft, TU Clausthal, 38678 Clausthal-Zellerfeld

Stieglitz, Michael, Institut für Mathematische Stochastik, Universität Karlsruhe, 76128 Karlsruhe

Sucky, Eric, Seminar für Logistik und Verkehr, Johann-Wolfgang-Goethe-Universität, 60054 Frankfurt am Main

Trautmann, Norbert, Assistenzprofessur für Quantitative Methoden der BWL, Universität Bern, 3012 Bern, Switzerland

van Beek, Paul, Operations Research and Logistics Group, Wageningen University, 6706 KN Wageningen, The Netherlands

Waldmann, Karl-Heinz, Institut für Wirtschaftstheorie und Operations Research, Universität Karlsruhe, 76128 Karlsruhe

Yao, Jen-Chih, Department of Applied Mathematics, National Sun Yat-sen University, Kaoshiung, Taiwan

Zhang, Shuo, Department of Production Management, TU Berlin, 10585 Berlin

Zimmermann, Jürgen, Institut für Wirtschaftswissenschaft, TU Clausthal, 38678 Clausthal-Zellerfeld